Akshay Bhala

Employee Attrition

HW01

Importing Libraries

```
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

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

```
library(dplyr)
library(arules)
```

```
## Loading required package: Matrix
```

```
## 
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
## 
##     expand, pack, unpack
```

```
## 
## Attaching package: 'arules'
```

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

```
## The following objects are masked from 'package:base':
## 
##     abbreviate, write
```

```
library(ggplot2)
library(arulesViz)
```

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'seriation':
##   method         from
##   reorder.hclust gclus
```

```
library(shiny)
library(rsconnect)
```

```
## Warning: package 'rsconnect' was built under R version 3.6.2
```

```
##
## Attaching package: 'rsconnect'
```

```
## The following object is masked from 'package:shiny':
##
##     serverInfo
```

Reading file using read_csv

```
myData <- read_csv("employee_attrition.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Attrition = col_character(),
##   BusinessTravel = col_character(),
##   Department = col_character(),
##   EducationField = col_character(),
##   Gender = col_character(),
##   JobRole = col_character(),
##   MaritalStatus = col_character(),
##   Over18 = col_character(),
##   OverTime = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
View(myData)
myData[myData==""] <- NA
```

Checking count of NA

```
sum(is.na(myData))
```

```
## [1] 11
```

Mode function

```
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
```

Removing Outliers function

```
outlierKD <- function(dt, var) {
     var_name <- eval(substitute(var),eval(dt))
     na1 <- sum(is.na(var_name))
     m1 <- mean(var_name, na.rm = T)
     par(mfrow=c(2, 2), oma=c(0,0,3,0))
     boxplot(var_name, main="With outliers")
     hist(var_name, main="With outliers", xlab=NA, ylab=NA)
     outlier <- boxplot.stats(var_name)$out
     mo <- mean(outlier)
     var_name <- ifelse(var_name %in% outlier, NA, var_name)
     boxplot(var_name, main="Without outliers")
     hist(var_name, main="Without outliers", xlab=NA, ylab=NA)
     title("Outlier Check of", outer=TRUE)
     na2 <- sum(is.na(var_name))
     cat("Outliers identified:", na2 - na1, "n")
     cat("Propotion (%) of outliers:", round((na2 - na1) / sum(!is.na(var_name))*100, 1), "n")
     cat("Mean of the outliers:", round(mo, 2), "n")
     m2 <- mean(var_name, na.rm = T)
     cat("Mean without removing outliers:", round(m1, 2), "n")
     cat("Mean if we remove outliers:", round(m2, 2), "n")
     dt[as.character(substitute(var))] <- invisible(var_name)
     assign(as.character(as.list(match.call())$dt), dt, envir = .GlobalEnv)
     cat("Outliers successfully removed", "n")
     return(invisible(dt))
     }
```

Coverting the columns to factors function

```
converttofactor <- function(vec)
{
  vec <- trimws(as.character(vec))
  vec <- as.factor(vec)
}
```

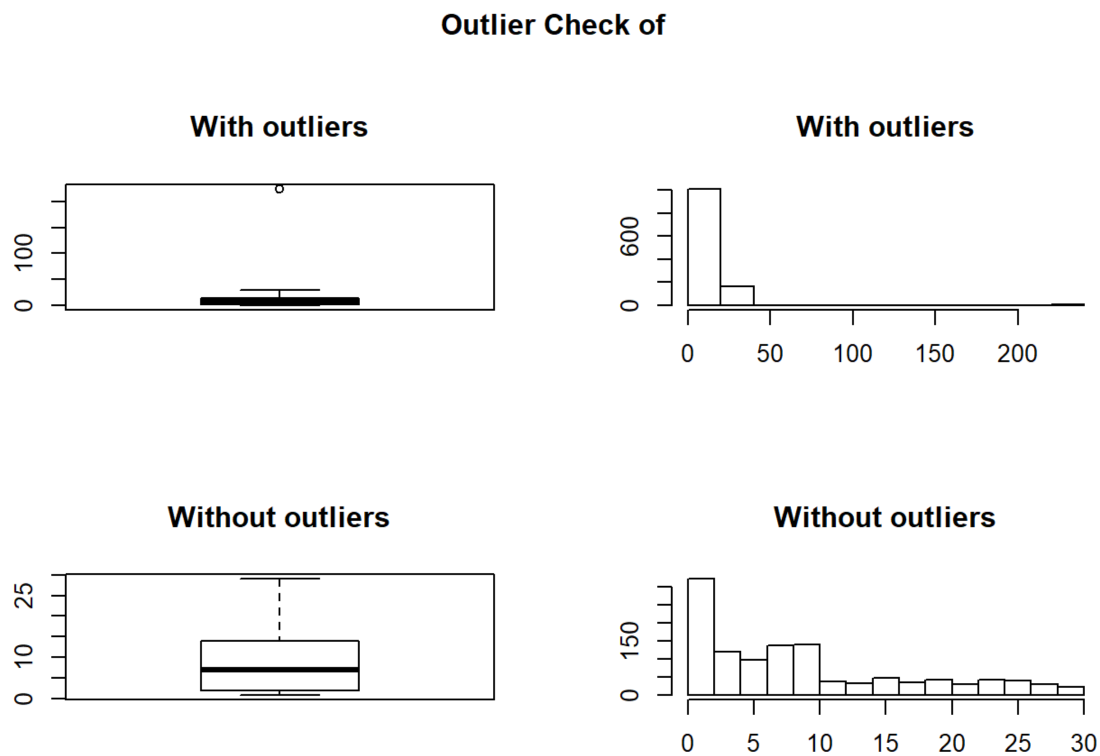
As we saw there are total 11 NA’s in the data set.Approach to remove NA’s is as follows: - checking outliers in columns. If outliers/Skewness present, remove them and then take the mean to replace Na’s and other blank columns - if no outliers just take the mean of numeric columns - for ordinal / categorical columns use mode to remove NA’s

Checking if outliers exist for numeric columns and outcasting those outliers and NA’s

```
summary(myData$DistanceFromHome)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   1.000   2.000   7.000   9.496  14.000 224.000       2
```

```
outlierKD(myData, DistanceFromHome)
```



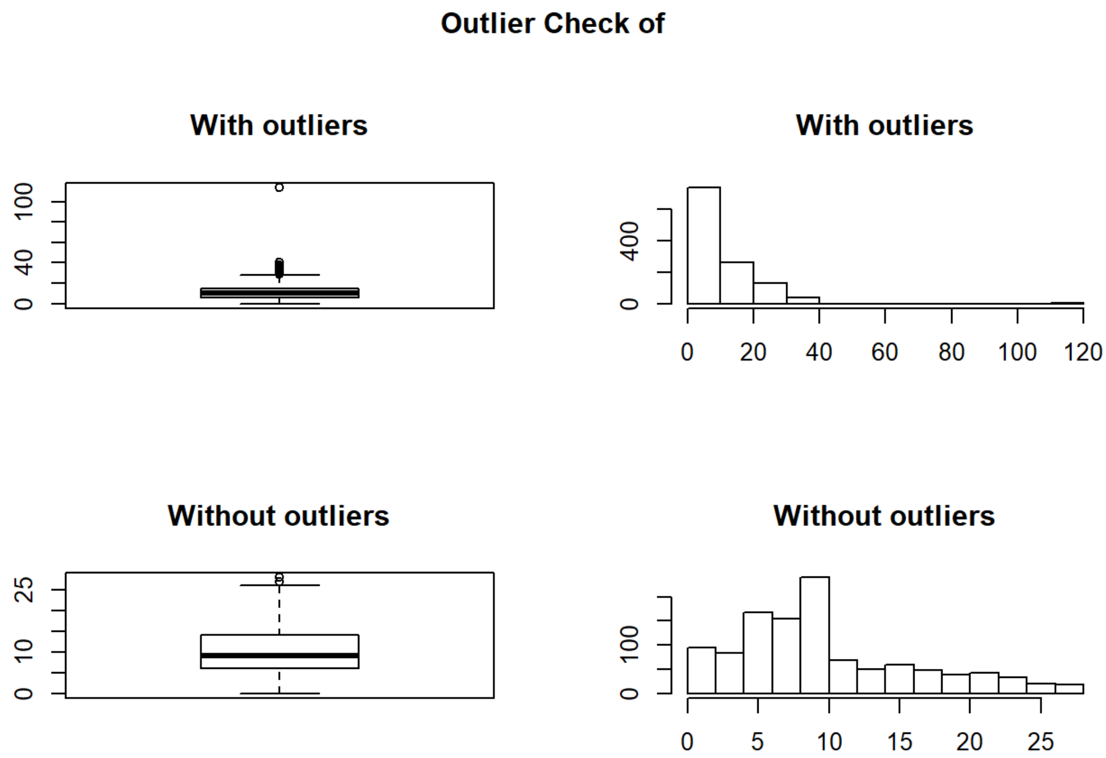

```
## Outliers identified: 1 nPropotion (%) of outliers: 0.1 nMean of the outliers: 224 nMean without removing outliers: 9.5 nMean if we remove outliers: 9.31 nOutliers successfully removed n
```

```
myData$DistanceFromHome[(is.na(myData$DistanceFromHome))]<- round(mean(myData$DistanceFromHome,na.rm=TRUE))

summary(myData$TotalWorkingYears)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0     6.0    10.0    11.4    15.0   114.0       2
```

```
outlierKD(myData, TotalWorkingYears)
```



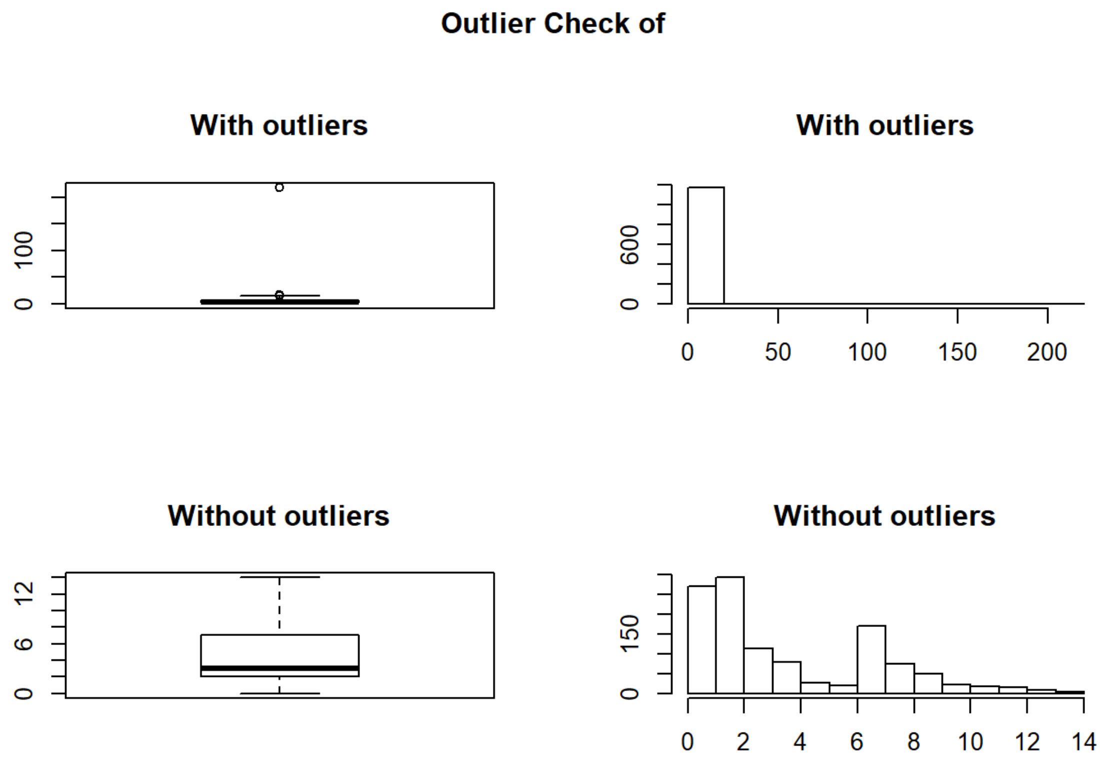

```
## Outliers identified: 54 nPropotion (%) of outliers: 4.8 nMean of the outliers: 34.07 nMean witho
ut removing outliers: 11.4 nMean if we remove outliers: 10.31 nOutliers successfully removed n
```

```
myData$TotalWorkingYears[(is.na(myData$TotalWorkingYears))]<- round(mean(myData$TotalWorkingYears,n
a.rm=TRUE))

summary(myData$YearsWithCurrManager)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   4.242   7.000 219.000
```

```
outlierKD(myData, YearsWithCurrManager)
```




```
## Outliers identified: 14 nPropotion (%) of outliers: 1.2 nMean of the outliers: 30.71 nMean witho
ut removing outliers: 4.24 nMean if we remove outliers: 3.92 nOutliers successfully removed n
```

```
myData$YearsWithCurrManager[(is.na(myData$YearsWithCurrManager))]<- round(mean(myData$YearsWithCurr
Manager,na.rm=TRUE))
```

Replacing Na’s using mode / mean for respective columns

```
summary(myData$JobLevel)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   1.000   2.000   2.069   3.000   5.000       1
```

```
myData$JobLevel[(is.na(myData$JobLevel))]<- getmode(myData$JobLevel)

summary(myData$PercentSalaryHike)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    11.0    12.0    14.0    15.3    18.0    25.0       1
```

```
myData$PercentSalaryHike[(is.na(myData$PercentSalaryHike))]<- getmode(myData$PercentSalaryHike)

summary(myData$PerformanceRating)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   3.000   3.000   3.000   3.163   3.000   4.000       1
```

```
myData$PerformanceRating[(is.na(myData$PerformanceRating))]<- getmode(myData$PerformanceRating)

summary(myData$RelationshipSatisfaction)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   1.000   2.000   3.000   2.718   4.000   4.000       1
```

```
myData$RelationshipSatisfaction[(is.na(myData$RelationshipSatisfaction))]<- getmode(myData$Relation
shipSatisfaction)

summary(myData$YearsSinceLastPromotion)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   0.000   1.000   2.125   2.000  15.000       1
```

```
myData$YearsSinceLastPromotion[(is.na(myData$YearsSinceLastPromotion))]<- round(mean(myData$YearsSi
nceLastPromotion,na.rm=TRUE))
```

Replacing Na’s using mode function for columns consisting characters

```
myData$Gender[(is.na(myData$Gender))]<- getmode(myData$Gender)

myData$OverTime[(is.na(myData$OverTime))]<- getmode(myData$OverTime)
```

Checking are there any Na’s left

```
sum(is.na(myData))
```

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

Converting character to factors

```
char_var <- sapply(myData, is.character)
myData[, char_var] <- lapply(myData[, char_var], as.factor)
str(myData)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 1176 obs. of  35 variables:
##  $ Age                     : num  30 52 42 55 35 51 42 23 38 27 ...
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 3 3 1 3 3
3 3 3 3 ...
##  $ DailyRate               : num  1358 1325 462 177 1029 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 3 2 2 3 2 2 2 3 ...
##  $ DistanceFromHome        : num  16 11 14 8 16 26 1 20 6 2 ...
##  $ Education               : num  1 4 2 1 3 4 2 1 2 1 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 4 4 2 3 2 2 5 3 ...
##  $ EmployeeCount           : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeNumber          : num  1479 813 936 1278 1529 ...
##  $ EnvironmentSatisfaction : num  4 4 3 4 4 1 4 1 4 3 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 2 1 1 2 1 1 1 2 1 2 ...
##  $ HourlyRate              : num  96 82 68 37 91 66 43 97 40 85 ...
##  $ JobInvolvement          : num  3 3 2 2 2 3 2 3 2 3 ...
##  $ JobLevel                : num  2 2 2 4 3 4 2 2 1 1 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 3 8 1 1 4 5 3
3 9 ...
##  $ JobSatisfaction         : num  3 3 3 2 2 3 4 3 3 1 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 2 2 3 1 3 2 2 3 2 1 ...
##  $ MonthlyIncome           : num  5301 3149 6244 13577 8606 ...
##  $ MonthlyRate             : num  2939 21821 7824 25592 21195 ...
##  $ NumCompaniesWorked      : num  8 8 7 1 1 2 9 0 1 0 ...
##  $ Over18                  : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OverTime                : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 1 1 1 ...
##  $ PercentSalaryHike       : num  15 20 17 15 19 14 13 14 11 11 ...
##  $ PerformanceRating       : num  3 4 3 3 3 3 3 3 3 3 ...
##  $ RelationshipSatisfaction: num  3 2 1 4 4 3 4 2 2 2 ...
##  $ StandardHours           : num  80 80 80 80 80 80 80 80 80 80 ...
##  $ StockOptionLevel        : num  2 1 0 1 0 1 1 0 1 1 ...
##  $ TotalWorkingYears       : num  4 9 10 10 11 10 8 5 5 5 ...
##  $ TrainingTimesLastYear   : num  2 3 6 3 3 2 4 2 3 3 ...
##  $ WorkLifeBalance         : num  2 3 3 3 1 2 3 3 3 3 ...
##  $ YearsAtCompany          : num  2 5 5 33 11 20 4 4 5 4 ...
##  $ YearsInCurrentRole      : num  1 2 4 9 8 6 3 3 4 3 ...
##  $ YearsSinceLastPromotion : num  2 1 0 15 3 4 0 1 0 0 ...
##  $ YearsWithCurrManager    : num  2 4 3 0 3 4 2 2 4 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Age = col_double(),
##   ..   Attrition = col_character(),
##   ..   BusinessTravel = col_character(),
##   ..   DailyRate = col_double(),
##   ..   Department = col_character(),
##   ..   DistanceFromHome = col_double(),
##   ..   Education = col_double(),
##   ..   EducationField = col_character(),
##   ..   EmployeeCount = col_double(),
##   ..   EmployeeNumber = col_double(),
##   ..   EnvironmentSatisfaction = col_double(),
##   ..   Gender = col_character(),
##   ..   HourlyRate = col_double(),
##   ..   JobInvolvement = col_double(),
##   ..   JobLevel = col_double(),
##   ..   JobRole = col_character(),
##   ..   JobSatisfaction = col_double(),
##   ..   MaritalStatus = col_character(),
##   ..   MonthlyIncome = col_double(),
```

```
##   ..   MonthlyRate = col_double(),
##   ..   NumCompaniesWorked = col_double(),
##   ..   Over18 = col_character(),
##   ..   OverTime = col_character(),
##   ..   PercentSalaryHike = col_double(),
##   ..   PerformanceRating = col_double(),
##   ..   RelationshipSatisfaction = col_double(),
##   ..   StandardHours = col_double(),
##   ..   StockOptionLevel = col_double(),
##   ..   TotalWorkingYears = col_double(),
##   ..   TrainingTimesLastYear = col_double(),
##   ..   WorkLifeBalance = col_double(),
##   ..   YearsAtCompany = col_double(),
##   ..   YearsInCurrentRole = col_double(),
##   ..   YearsSinceLastPromotion = col_double(),
##   ..   YearsWithCurrManager = col_double()
##   .. )
```

Exploratory data analysis

We are performing Exploratory Data analysis to check which variables are responsible for Employee Attrition

```
ETD <- myData %>% group_by(Attrition) %>%
  summarise(count=n(), DailyRate=round(mean(DailyRate), 1),
      DistanceFromHome=round(mean(DistanceFromHome),1),EnvironmentSatisfaction=round(mean(Environme
ntSatisfaction),1),HourlyRate=round(mean(HourlyRate),1),
      JobSatisfaction=round(mean(JobSatisfaction),1),MonthlyIncome=round(mean(MonthlyIncome),1),Num
CompaniesWorked=round(mean(NumCompaniesWorked),1),PercentSalaryHike=round(mean(PercentSalaryHike),1
),PerformanceRating=round(mean(PerformanceRating),1),RelationshipSatisfaction=round(mean(Relationsh
ipSatisfaction),1),TotalWorkingYears=round(mean(TotalWorkingYears),1),TrainingTimesLastYear=round(m
ean(TrainingTimesLastYear),1),WorkLifeBalance=round(mean(WorkLifeBalance),1),YearsAtCompany=round(m
ean(YearsAtCompany),1),YearsInCurrentRole=round(mean(YearsInCurrentRole),1),YearsSinceLastPromotion
=round(mean(YearsSinceLastPromotion),1),YearsWithCurrManager=round(mean(YearsWithCurrManager),1))
ETD <- as.data.frame(t(ETD))
ETD
```

```
##                              V1     V2
## Attrition                    No    Yes
## count                       991    185
## DailyRate                 811.9  738.7
## DistanceFromHome            9.0   11.2
## EnvironmentSatisfaction     2.8    2.4
## HourlyRate                 65.7   66.4
## JobSatisfaction             2.8    2.4
## MonthlyIncome            6845.3 4812.5
## NumCompaniesWorked          2.7    3.0
## PercentSalaryHike          15.3   15.3
## PerformanceRating           3.2    3.2
## RelationshipSatisfaction    2.7    2.7
## TotalWorkingYears          10.8    7.6
## TrainingTimesLastYear       2.8    2.6
## WorkLifeBalance             2.8    2.6
## YearsAtCompany              7.2    5.2
## YearsInCurrentRole          4.4    2.8
## YearsSinceLastPromotion     2.2    1.8
## YearsWithCurrManager        4.1    2.7
```

We have observed that Employee Attrition = yes has a count of 185 which can be due to low Daily Rate/ Hourly Rate /Monthly Income and more Distance from Home. We will further explore whether this is true using various visualizations and ARM.

Converting to Factors

```
myData$Education <- converttofactor(myData$Education)
myData$EnvironmentSatisfaction <- converttofactor(myData$EnvironmentSatisfaction)
myData$JobInvolvement <- converttofactor(myData$JobInvolvement)
myData$JobLevel <- converttofactor(myData$JobLevel)
myData$JobSatisfaction <- converttofactor(myData$JobSatisfaction)
myData$NumCompaniesWorked <- converttofactor(myData$NumCompaniesWorked)
myData$PerformanceRating <- converttofactor(myData$PerformanceRating)
myData$RelationshipSatisfaction <- converttofactor(myData$RelationshipSatisfaction)
myData$StockOptionLevel <- converttofactor(myData$StockOptionLevel)
myData$TrainingTimesLastYear <- converttofactor(myData$TrainingTimesLastYear)
myData$NumCompaniesWorked <- converttofactor(myData$NumCompaniesWorked)
myData$WorkLifeBalance<-converttofactor(myData$WorkLifeBalance)
myData$Attrition <- as.factor(myData$Attrition)
myData$BusinessTravel <- as.factor(myData$BusinessTravel)
myData$Department <- as.factor(myData$Department)
myData$EducationField <- as.factor(myData$EducationField)
myData$Gender <- as.factor(myData$Gender)
myData$JobRole <- as.factor(myData$JobRole)
myData$MaritalStatus <- as.factor(myData$MaritalStatus)
myData$OverTime <- as.factor(myData$OverTime)
```

Our data is ready

```
str(myData)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 1176 obs. of  35 variables:
##  $ Age                     : num  30 52 42 55 35 51 42 23 38 27 ...
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 2 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 3 3 1 3 3
3 3 3 3 ...
##  $ DailyRate               : num  1358 1325 462 177 1029 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 3 2 2 3 2 2 2 3 ...
##  $ DistanceFromHome        : num  16 11 14 8 16 26 1 20 6 2 ...
##  $ Education               : Factor w/ 5 levels "1","2","3","4",..: 1 4 2 1 3 4 2 1 2 1 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 4 4 2 3 2 2 5 3 ...
##  $ EmployeeCount           : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeNumber          : num  1479 813 936 1278 1529 ...
##  $ EnvironmentSatisfaction : Factor w/ 4 levels "1","2","3","4": 4 4 3 4 4 1 4 1 4 3 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 2 1 1 2 1 1 1 2 1 2 ...
##  $ HourlyRate              : num  96 82 68 37 91 66 43 97 40 85 ...
##  $ JobInvolvement          : Factor w/ 4 levels "1","2","3","4": 3 3 2 2 2 3 2 3 2 3 ...
##  $ JobLevel                : Factor w/ 5 levels "1","2","3","4",..: 2 2 2 4 3 4 2 2 1 1 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 3 8 1 1 4 5 3
3 9 ...
##  $ JobSatisfaction         : Factor w/ 4 levels "1","2","3","4": 3 3 3 2 2 3 4 3 3 1 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 2 2 3 1 3 2 2 3 2 1 ...
##  $ MonthlyIncome           : num  5301 3149 6244 13577 8606 ...
##  $ MonthlyRate             : num  2939 21821 7824 25592 21195 ...
##  $ NumCompaniesWorked      : Factor w/ 10 levels "0","1","2","3",..: 9 9 8 2 2 3 10 1 2 1 ...
##  $ Over18                  : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OverTime                : Factor w/ 2 levels "No","Yes": 1 1 1 2 1 1 2 1 1 1 ...
##  $ PercentSalaryHike       : num  15 20 17 15 19 14 13 14 11 11 ...
##  $ PerformanceRating       : Factor w/ 2 levels "3","4": 1 2 1 1 1 1 1 1 1 1 ...
##  $ RelationshipSatisfaction: Factor w/ 4 levels "1","2","3","4": 3 2 1 4 4 3 4 2 2 2 ...
##  $ StandardHours           : num  80 80 80 80 80 80 80 80 80 80 ...
##  $ StockOptionLevel        : Factor w/ 4 levels "0","1","2","3": 3 2 1 2 1 2 2 1 2 2 ...
##  $ TotalWorkingYears       : num  4 9 10 10 11 10 8 5 5 5 ...
##  $ TrainingTimesLastYear   : Factor w/ 7 levels "0","1","2","3",..: 3 4 7 4 4 3 5 3 4 4 ...
##  $ WorkLifeBalance         : Factor w/ 4 levels "1","2","3","4": 2 3 3 3 1 2 3 3 3 3 ...
##  $ YearsAtCompany          : num  2 5 5 33 11 20 4 4 5 4 ...
##  $ YearsInCurrentRole      : num  1 2 4 9 8 6 3 3 4 3 ...
##  $ YearsSinceLastPromotion : num  2 1 0 15 3 4 0 1 0 0 ...
##  $ YearsWithCurrManager    : num  2 4 3 0 3 4 2 2 4 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Age = col_double(),
##   ..   Attrition = col_character(),
##   ..   BusinessTravel = col_character(),
##   ..   DailyRate = col_double(),
##   ..   Department = col_character(),
##   ..   DistanceFromHome = col_double(),
##   ..   Education = col_double(),
##   ..   EducationField = col_character(),
##   ..   EmployeeCount = col_double(),
##   ..   EmployeeNumber = col_double(),
##   ..   EnvironmentSatisfaction = col_double(),
##   ..   Gender = col_character(),
##   ..   HourlyRate = col_double(),
##   ..   JobInvolvement = col_double(),
##   ..   JobLevel = col_double(),
##   ..   JobRole = col_character(),
##   ..   JobSatisfaction = col_double(),
##   ..   MaritalStatus = col_character(),
##   ..   MonthlyIncome = col_double(),
```

```
##   ..   MonthlyRate = col_double(),
##   ..   NumCompaniesWorked = col_double(),
##   ..   Over18 = col_character(),
##   ..   OverTime = col_character(),
##   ..   PercentSalaryHike = col_double(),
##   ..   PerformanceRating = col_double(),
##   ..   RelationshipSatisfaction = col_double(),
##   ..   StandardHours = col_double(),
##   ..   StockOptionLevel = col_double(),
##   ..   TotalWorkingYears = col_double(),
##   ..   TrainingTimesLastYear = col_double(),
##   ..   WorkLifeBalance = col_double(),
##   ..   YearsAtCompany = col_double(),
##   ..   YearsInCurrentRole = col_double(),
##   ..   YearsSinceLastPromotion = col_double(),
##   ..   YearsWithCurrManager = col_double()
##   .. )
```

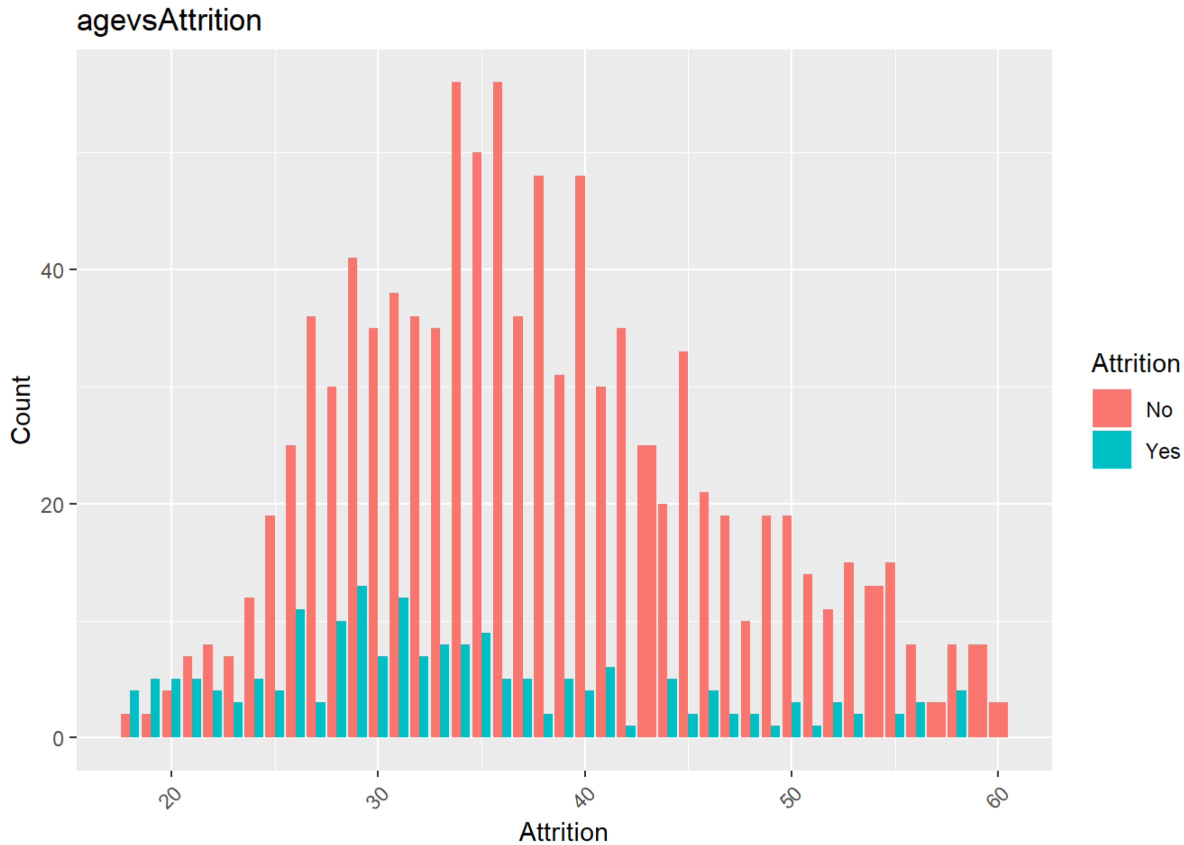
removing columns like EmployeeCount/EmployeeNumber/Over18/StandardHours

```
df1 <- myData
df1 <- df1[,c(-9,-10,-22,-27)]
```

Data Visualization

1

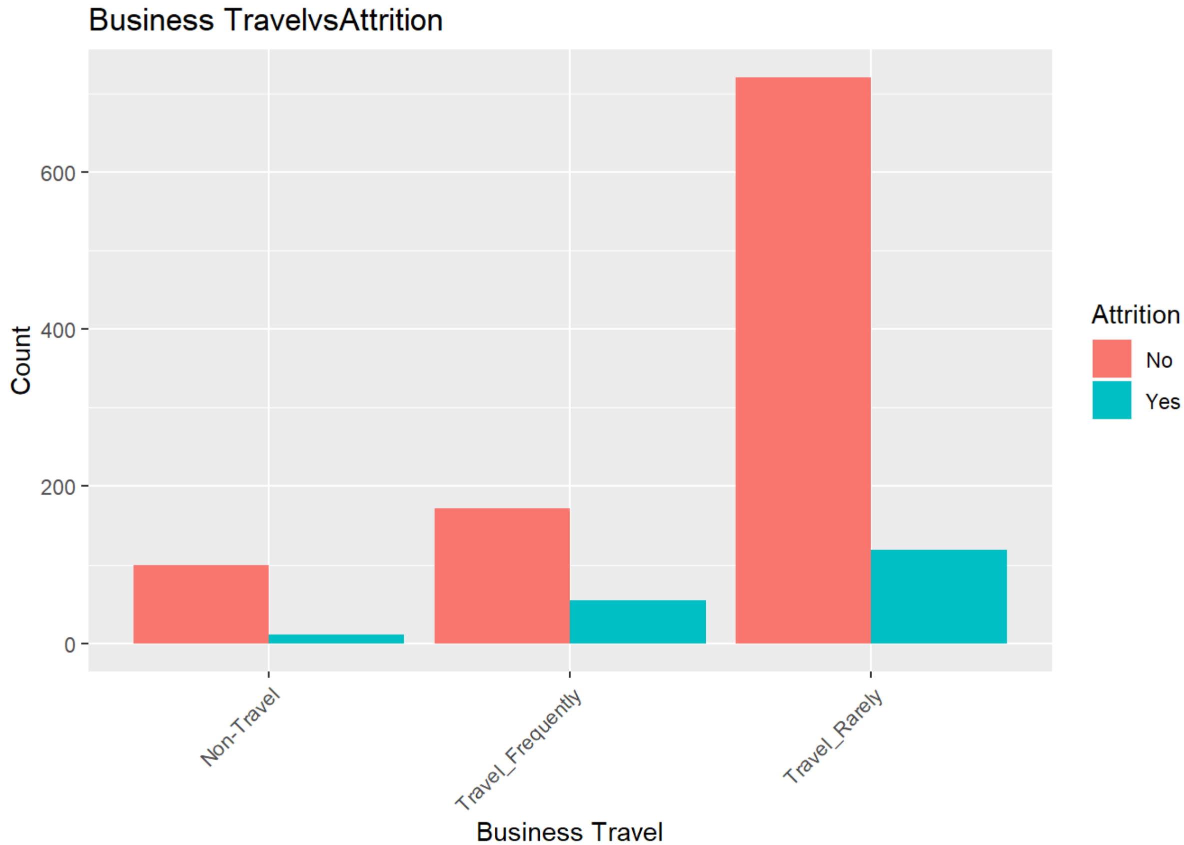
```
agevsAttrition <- ggplot(df1) +
  aes(x = df1$Age, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("agevsAttrition") +
  labs(x = "Attrition", y = "Count", fill = "Attrition")
agevsAttrition
```



Analysis: The above graph shows that between age 20 to 35 there are highest no. of employee attrition.

2

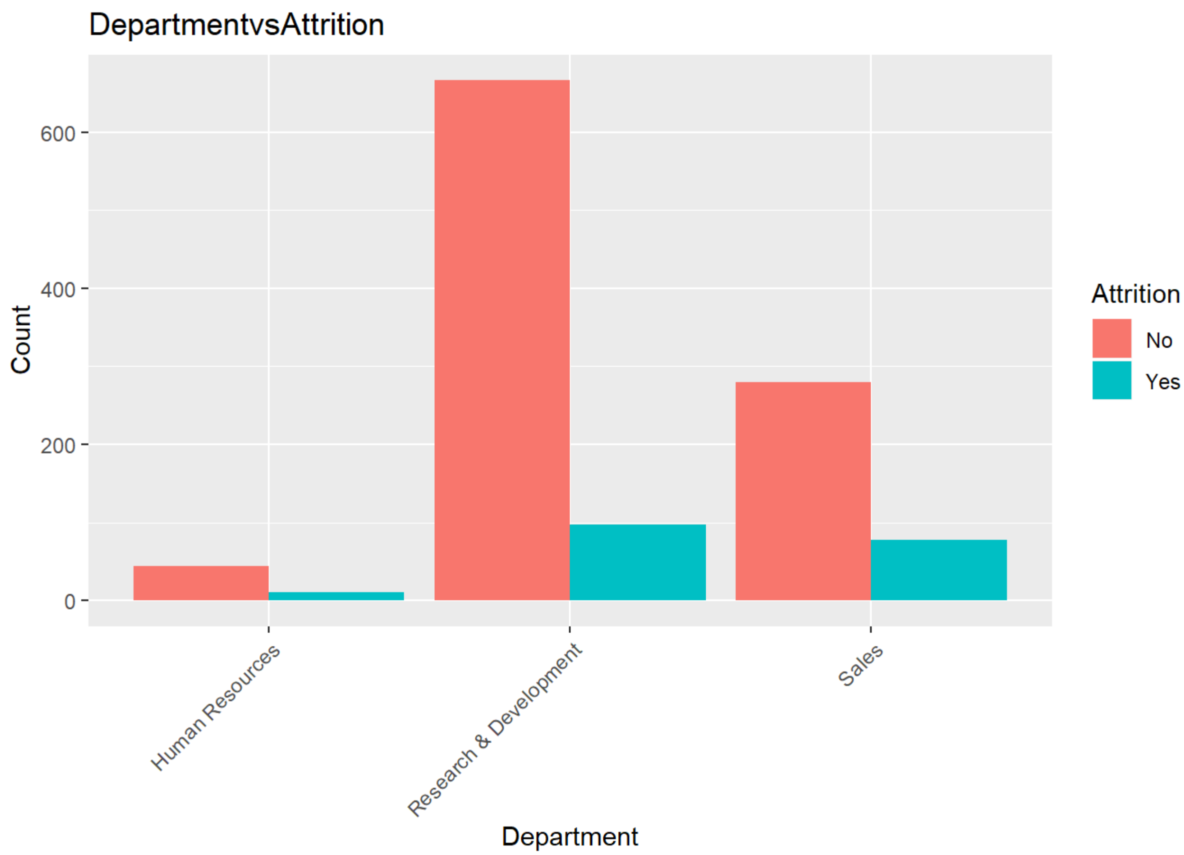
```
BTvsAttrition <- ggplot(df1) +
  aes(x = df1$BusinessTravel, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Business TravelvsAttrition") +
  labs(x = "Business Travel", y = "Count", fill = "Attrition")
BTvsAttrition
```



Analysis: The above graph shows that Employees who travel rarely tends more towards employee attrition.

3

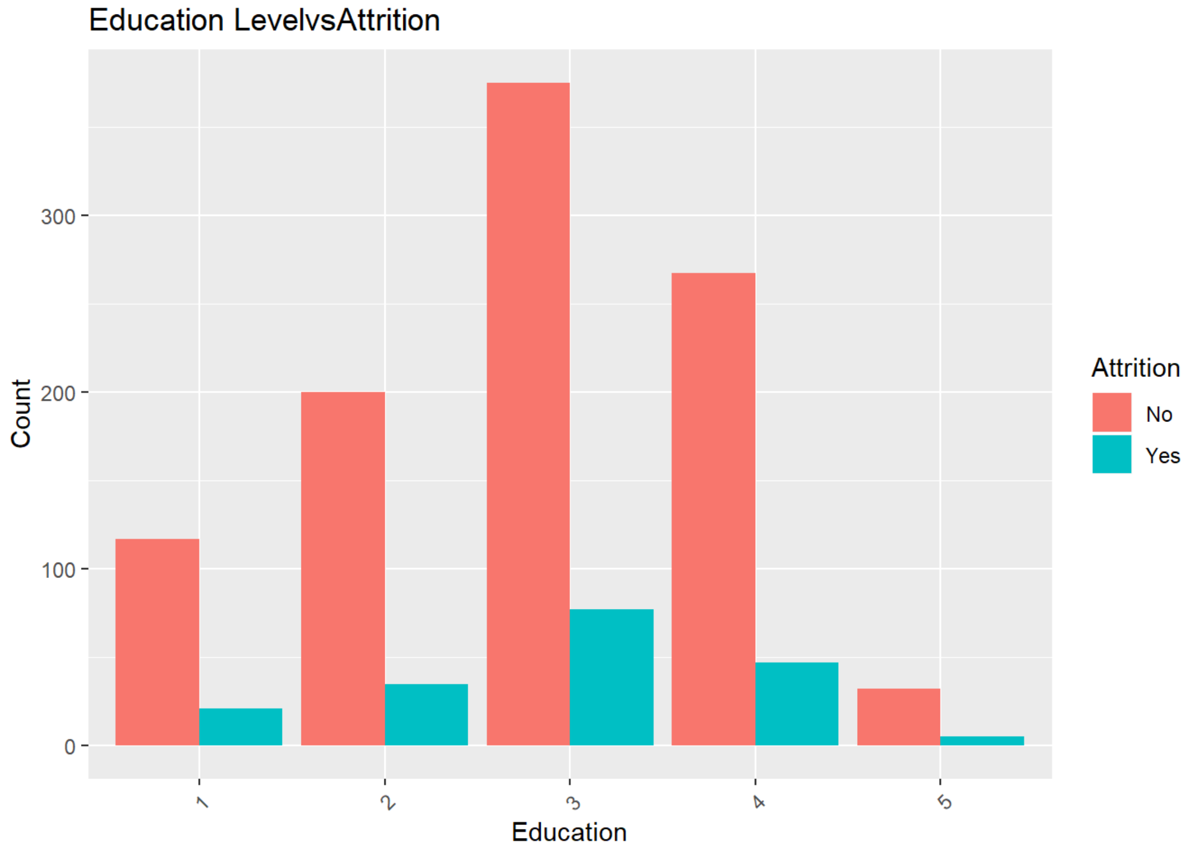
```
DeptvsAttrition <- ggplot(df1) +
  aes(x = df1$Department, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("DepartmentvsAttrition") +
  labs(x = "Department", y = "Count", fill = "Attrition")
DeptvsAttrition
```



Analysis: The above graph shows that Employees in sales field shows more employee attrition.

4

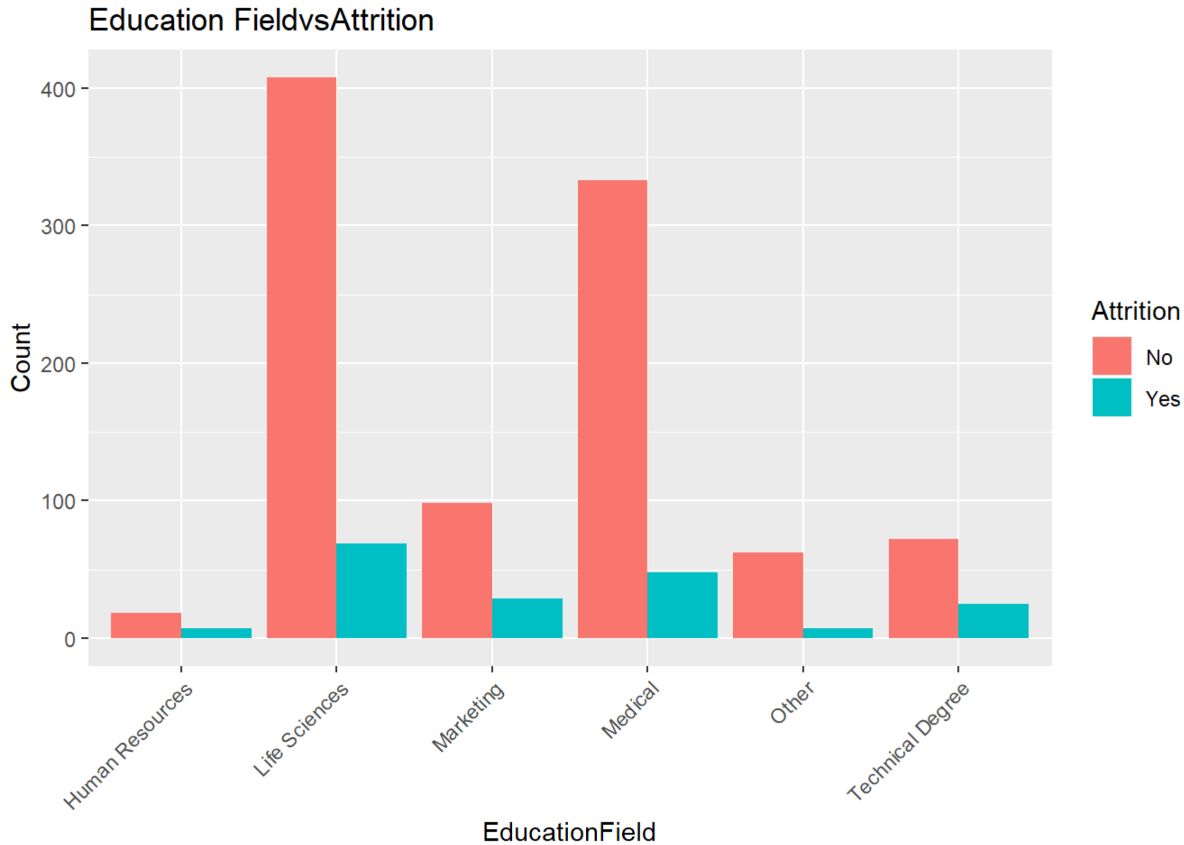
```
EdvsAttrition <- ggplot(df1) +
  aes(x = df1$Education, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Education LevelvsAttrition") +
  labs(x = "Education", y = "Count", fill = "Attrition")
EdvsAttrition
```



Analysis: The above graph shows that Employees with average education are more in employee attrition.

5

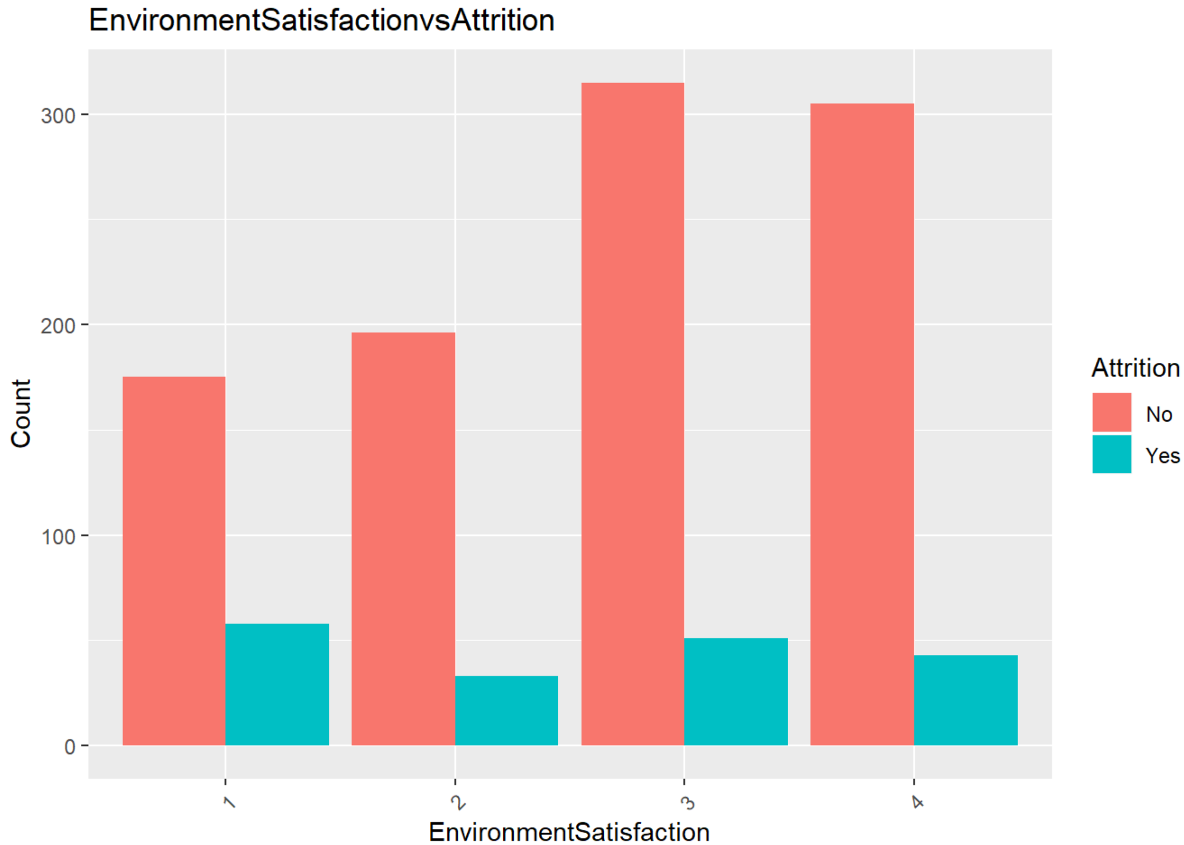
```
EdfvsAttrition <- ggplot(df1) +
  aes(x = df1$EducationField, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Education FieldvsAttrition") +
  labs(x = "EducationField", y = "Count", fill = "Attrition")
EdfvsAttrition
```



Analysis: The above graph shows that Employees from mediacal and life science field have highest no. of employees and large amount of employees tends toward employee attrition

6

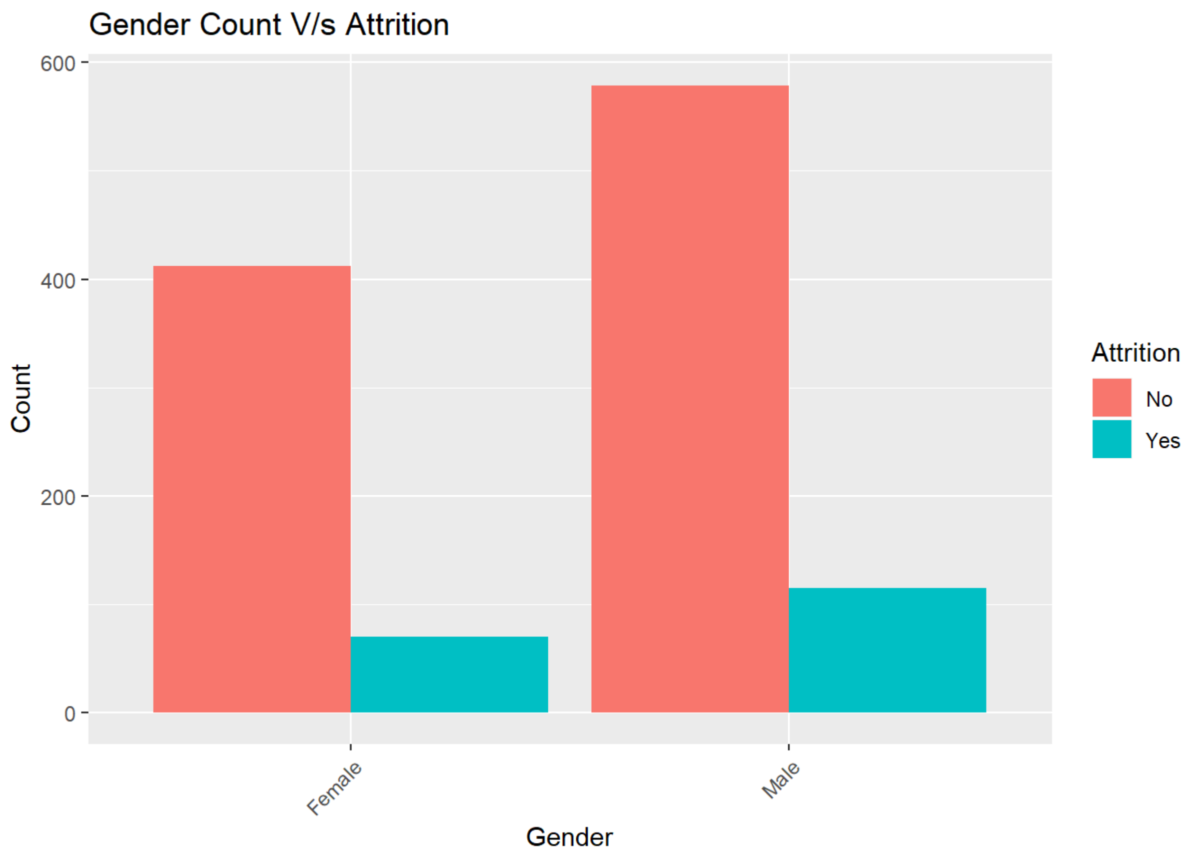
```
EvsvsAttrition <- ggplot(df1) +
  aes(x = df1$EnvironmentSatisfaction, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("EnvironmentSatisfactionvsAttrition") +
  labs(x = "EnvironmentSatisfaction", y = "Count", fill = "Attrition")
EvsvsAttrition
```



Analysis: The above graph shows that lower the Environment Satisfaction more the number of Attrition.

7

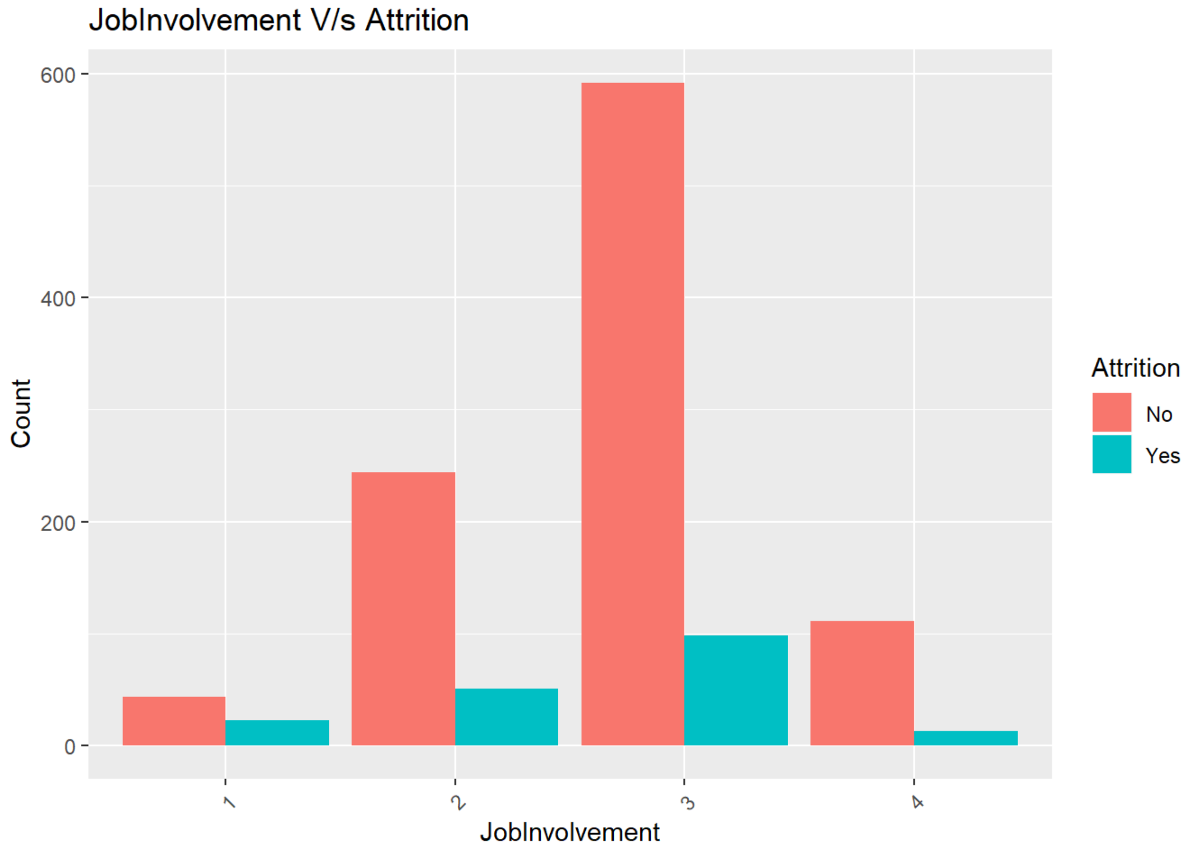
```
genderVsAttrition <- ggplot(df1) +
  aes(x = df1$Gender, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("Gender Count V/s Attrition") +
  labs(x = "Gender", y = "Count", fill = "Attrition")
genderVsAttrition
```



Analysis: The above graph shows that Male Employees tends more towards employee attrition.

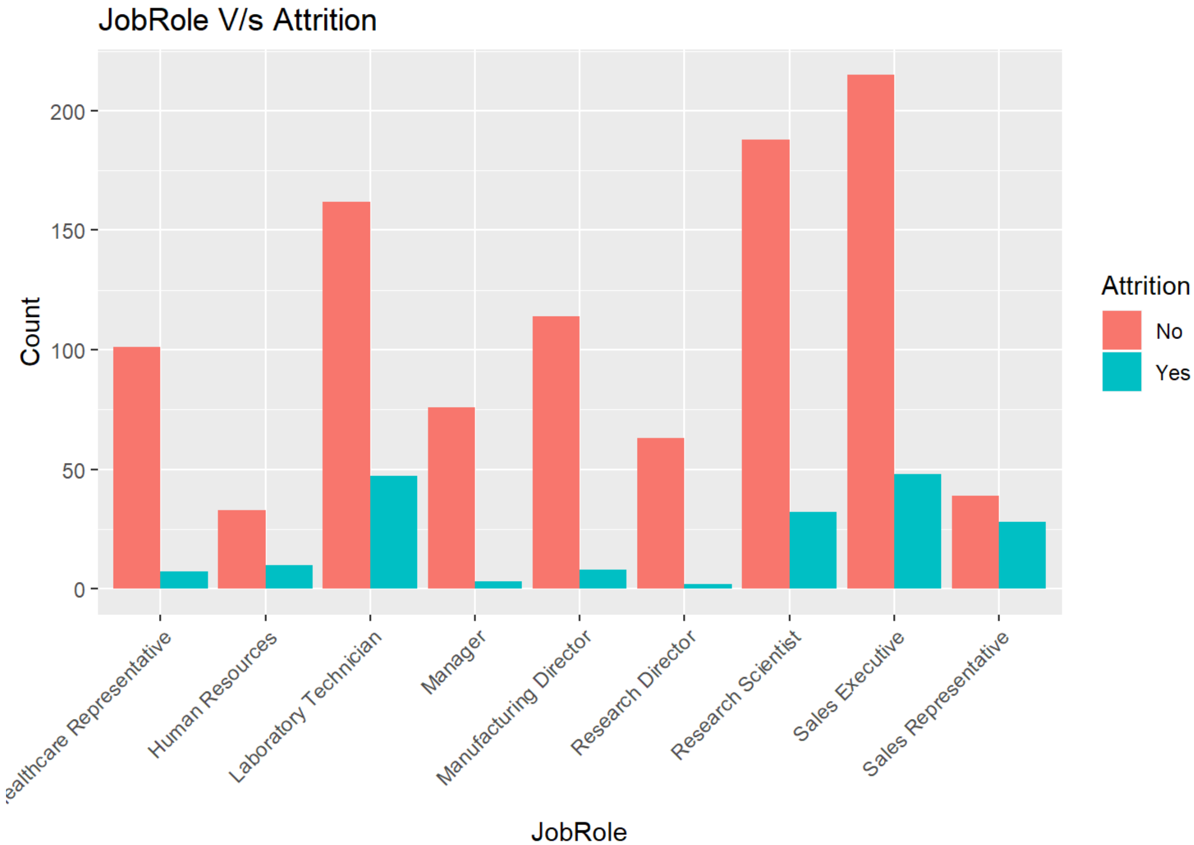
8

```
JobInvolvementVsAttrition <- ggplot(df1) +
  aes(x = df1$JobInvolvement, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("JobInvolvement V/s Attrition") +
  labs(x = "JobInvolvement", y = "Count", fill = "Attrition")
JobInvolvementVsAttrition
```



9

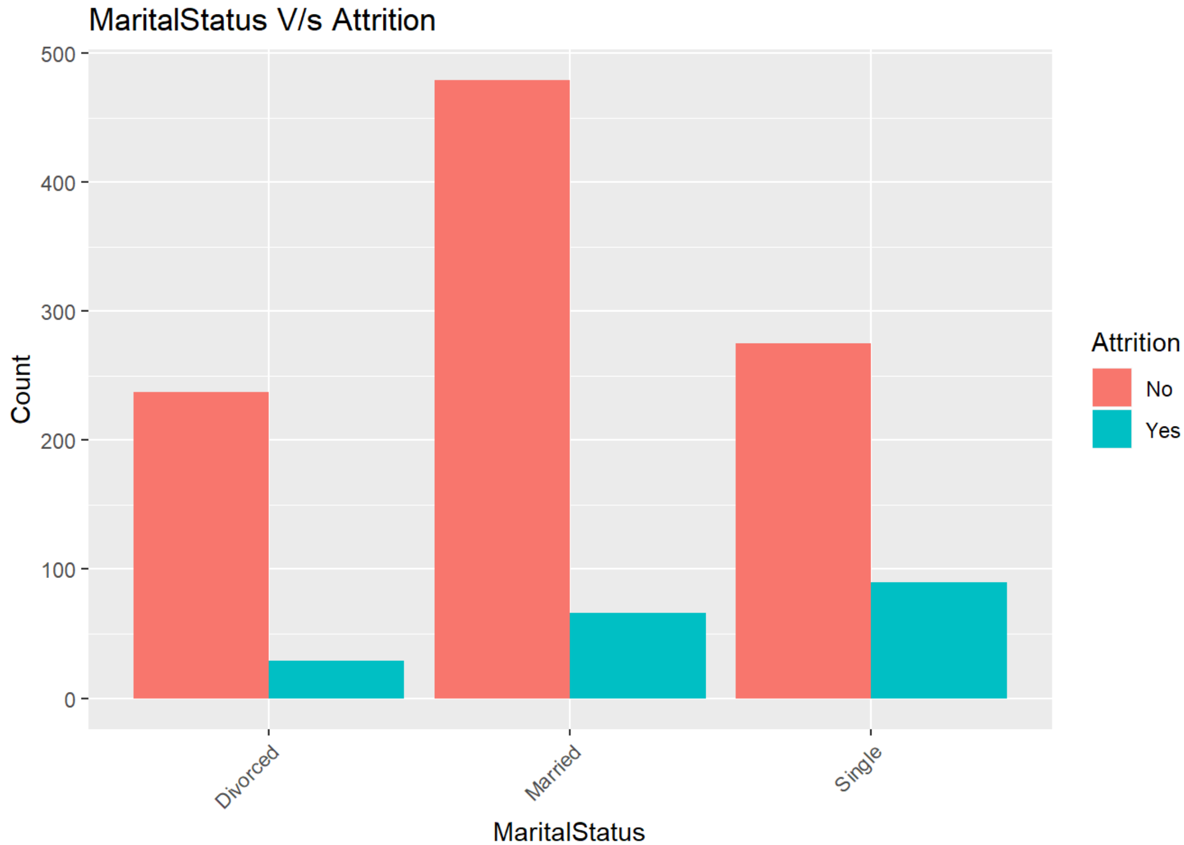
```
JobRoleVsAttrition <- ggplot(df1) +
  aes(x = df1$JobRole, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("JobRole V/s Attrition") +
  labs(x = "JobRole", y = "Count", fill = "Attrition")
JobRoleVsAttrition
```



Analysis: The above graph shows that Employees who are sales executive/Sales Representative tends more towards employee attrition.

10

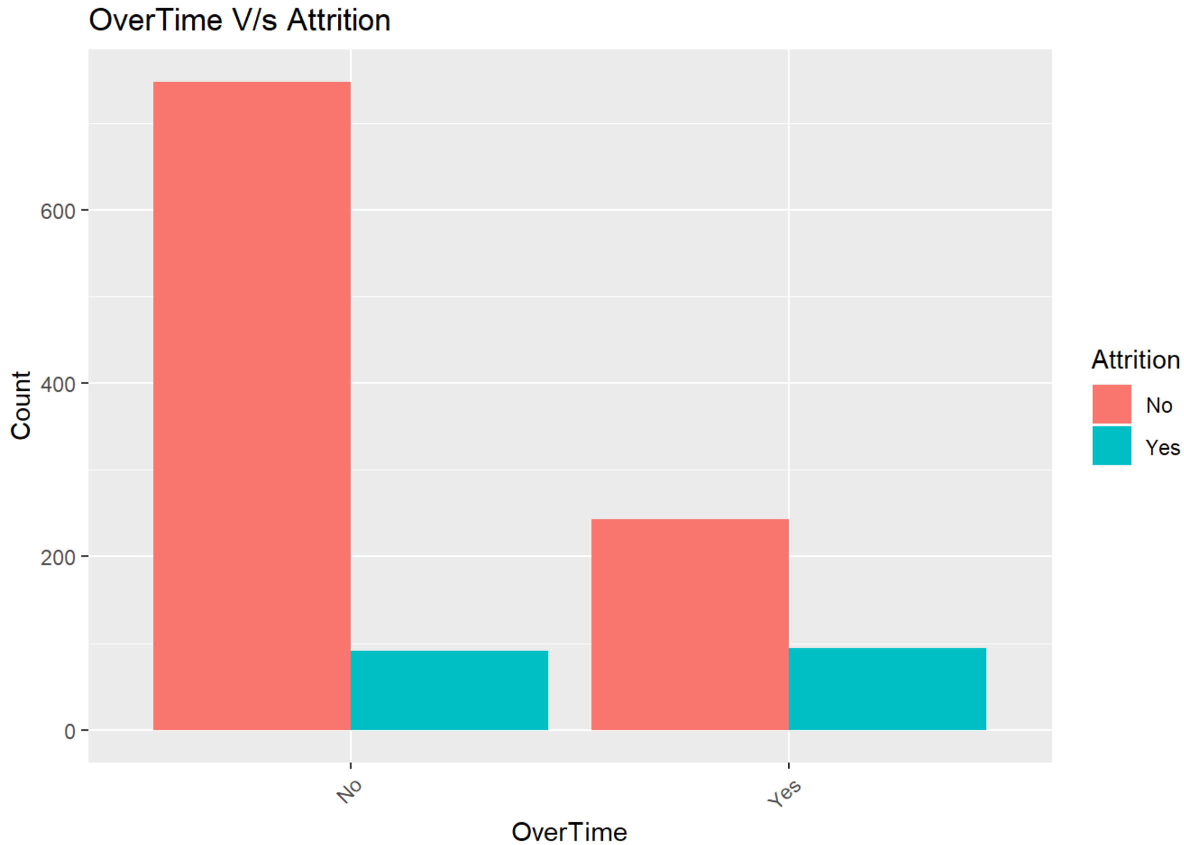
```
MaritalStatusVsAttrition <- ggplot(df1) +
  aes(x = df1$MaritalStatus, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("MaritalStatus V/s Attrition") +
  labs(x = "MaritalStatus", y = "Count", fill = "Attrition")
MaritalStatusVsAttrition
```



Analysis: The above graph shows that Employees who are single tends to show employee attrition.

11

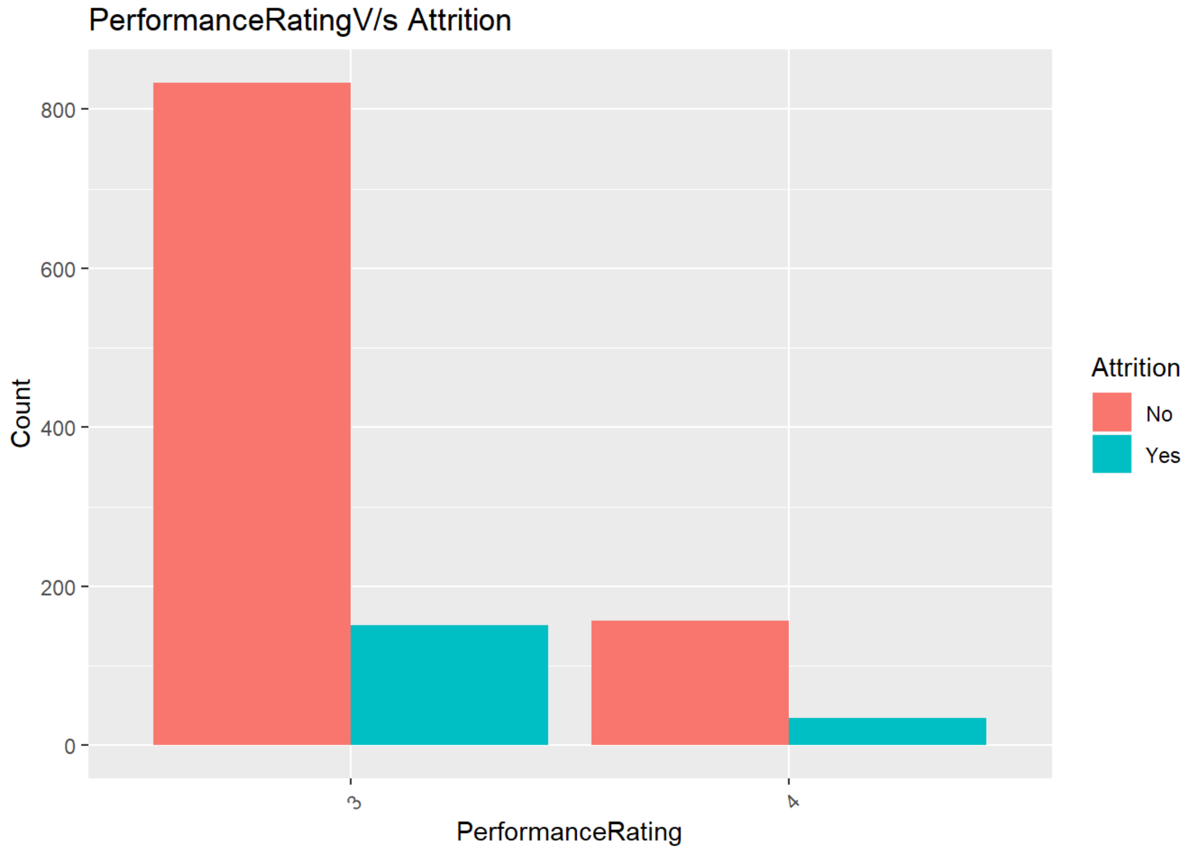
```
OverTimeVsAttrition <- ggplot(df1) +
  aes(x = df1$OverTime, fill = df1$Attrition) +
  geom_bar(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle("OverTime V/s Attrition") +
  labs(x = "OverTime", y = "Count", fill = "Attrition")
OverTimeVsAttrition
```



Analysis: The above graph shows that there are few Employees who work overtime and among them almost half of employees tends towards Employee Attrition.

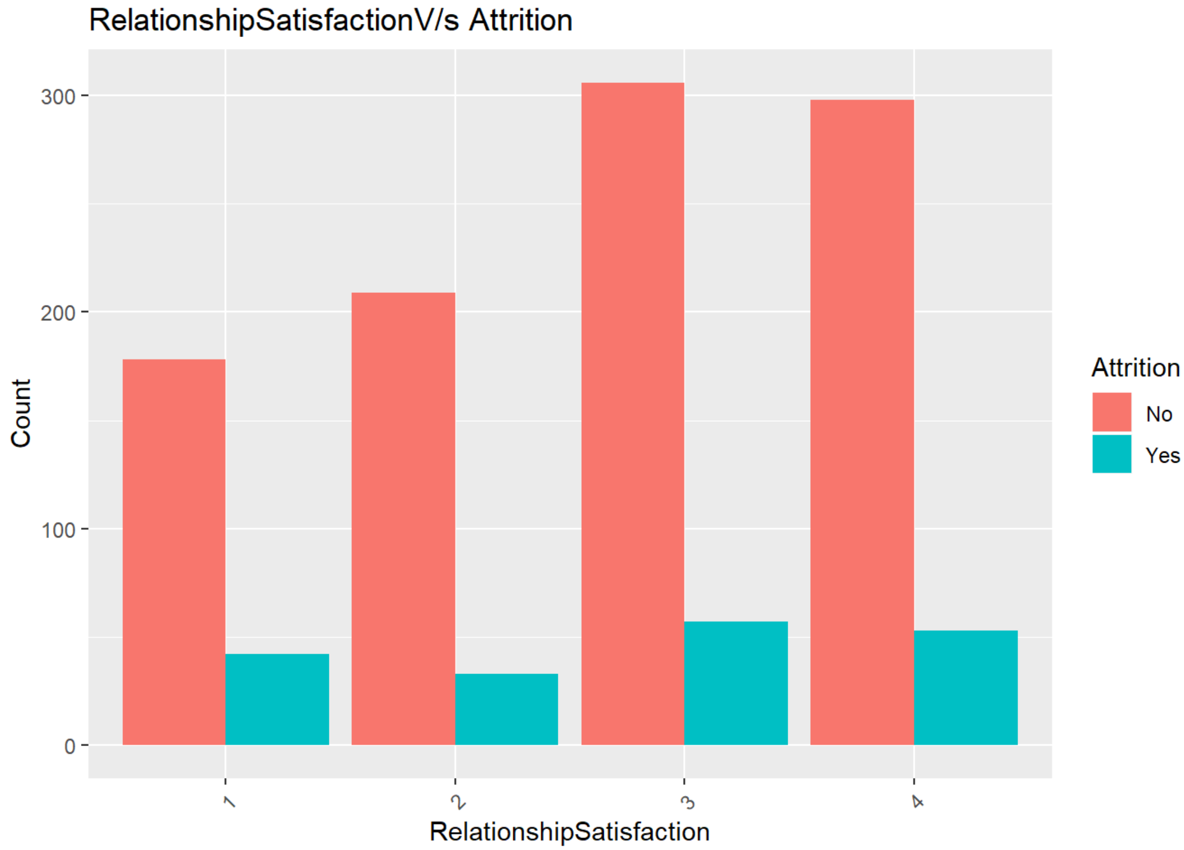
12

```
PerformanceRatingVsAttrition <- ggplot(df1) +
  aes(x = df1$PerformanceRating, fill = df1$Attrition)+
  geom_bar(position = "dodge")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ggtitle("PerformanceRatingV/s Attrition")+
  labs(x = "PerformanceRating", y = "Count", fill = "Attrition")
PerformanceRatingVsAttrition
```



13

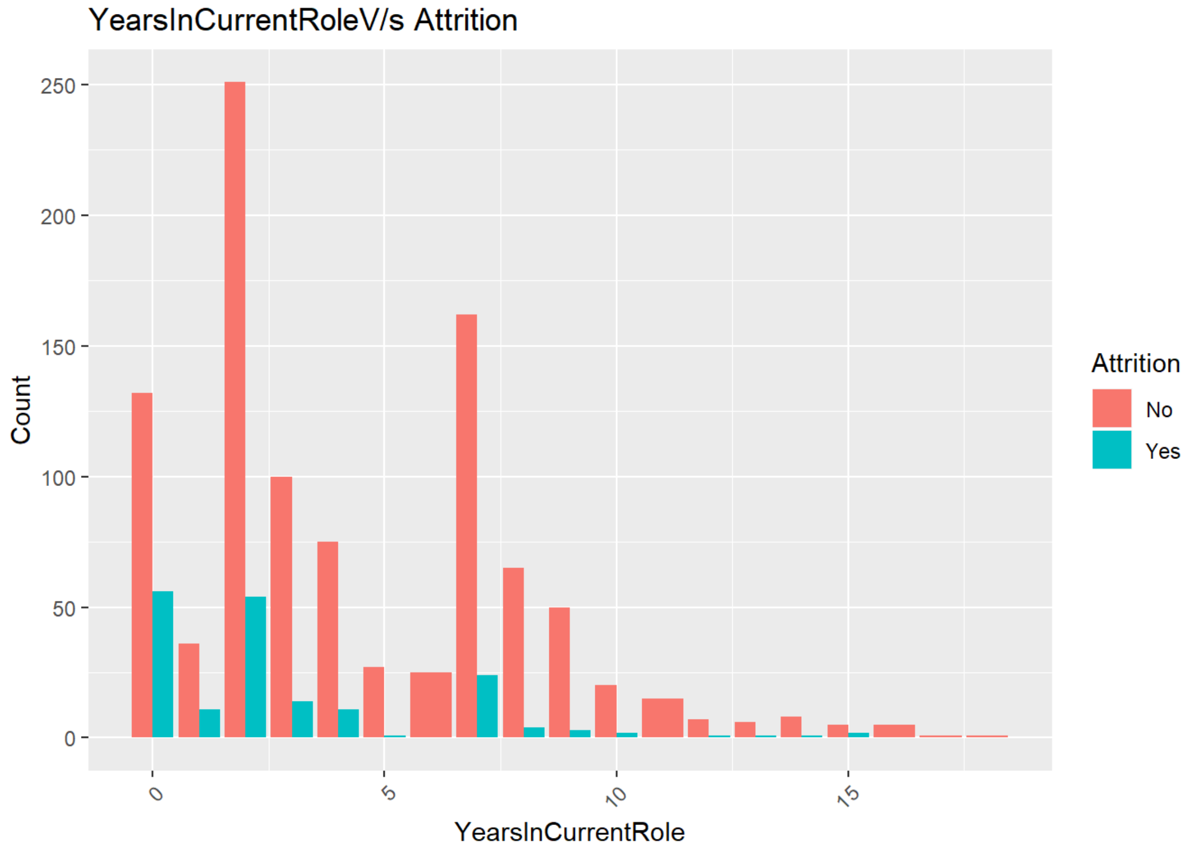
```
RelationshipSatisfactionVsAttrition <- ggplot(df1) +
  aes(x = df1$RelationshipSatisfaction, fill = df1$Attrition)+
  geom_bar(position = "dodge")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ggtitle("RelationshipSatisfactionV/s Attrition")+
  labs(x = "RelationshipSatisfaction", y = "Count", fill = "Attrition")
RelationshipSatisfactionVsAttrition
```



Analysis: The above two graphs shows that Employees with low performance rating and Relationship Satisfaction tends towards Employee Attrition.

14

```
YearsInCurrentRoleVsAttrition <- ggplot(df1) +
  aes(x = df1$YearsInCurrentRole, fill = df1$Attrition)+
  geom_bar(position = "dodge")+
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ggtitle("YearsInCurrentRoleV/s Attrition")+
  labs(x = "YearsInCurrentRole", y = "Count", fill = "Attrition")
YearsInCurrentRoleVsAttrition
```



Analysis: The above graphs shows that Employees new Employees tends more towards Employee Attrition.

Discretization

```
df1$Age <- discretize(df1$Age, method = "frequency", breaks = 3,
                       labels = c("young", "adult", "old"), order = T)
df1$DailyRate <- discretize(df1$DailyRate, method = "frequency", breaks = 4,
                       labels = c("low", "Medium", "High","Higher"), order = T)
df1$DistanceFromHome<-discretize(df1$DistanceFromHome,method = "frequency", breaks = 4,
                       labels = c("low", "Medium", "High","Higher"), order = T)
df1$HourlyRate<-discretize(df1$HourlyRate,method = "frequency", breaks = 4,
                       labels = c("low", "Medium", "High","Higher"), order = T)
df1$MonthlyIncome<-discretize(df1$MonthlyIncome,method = "frequency", breaks = 4,
                       labels = c("low", "Medium", "High","Higher"), order = T)
df1$MonthlyRate<-discretize(df1$MonthlyRate,method = "frequency", breaks = 4,
                       labels = c("low", "Medium", "High","Higher"), order = T)
df1$PercentSalaryHike<-discretize(df1$PercentSalaryHike,method = "frequency", breaks = 4,
                       labels = c("<5%", "5%<hike<10%", "10%<Hike<20%",">20%"), order = T)
df1$YearsAtCompany<-discretize(df1$YearsAtCompany,method = "frequency", breaks = 4,
                       labels = c("<5years", "5<Years<10", "10<Years<20",">20"), order = T)
df1$YearsInCurrentRole<-discretize(df1$YearsInCurrentRole,method = "frequency", breaks = 4,
                       labels = c("<5years", "5<Years<10", "10<Years<20",">20"), order = T)
df1$TotalWorkingYears<-cut(df1$TotalWorkingYears, breaks = 5,
                                 labels = c("<5years", "5<Years<10", "10<Years<20","20<years<25",">
25"), order = T)
df1$YearsSinceLastPromotion<-cut(df1$YearsSinceLastPromotion, breaks = 5,
                       labels = c("<5years", "5<Years<10", "10<Years<20","20<years<25",">25"), orde
r = T)
df1$YearsWithCurrManager<-discretize(df1$YearsWithCurrManager,method = "frequency", breaks = 4,
                       labels = c("<5years", "5<Years<10", "10<Years<20",">20"), order = T)
```

Transforming Dataframe into Transaction Matrix

```
SS<-as(df1,"transactions")
```

ARM with default settings displaying top 10 rules with the high confidence

```
Attrition_rules <- apriori(data=SS)
```

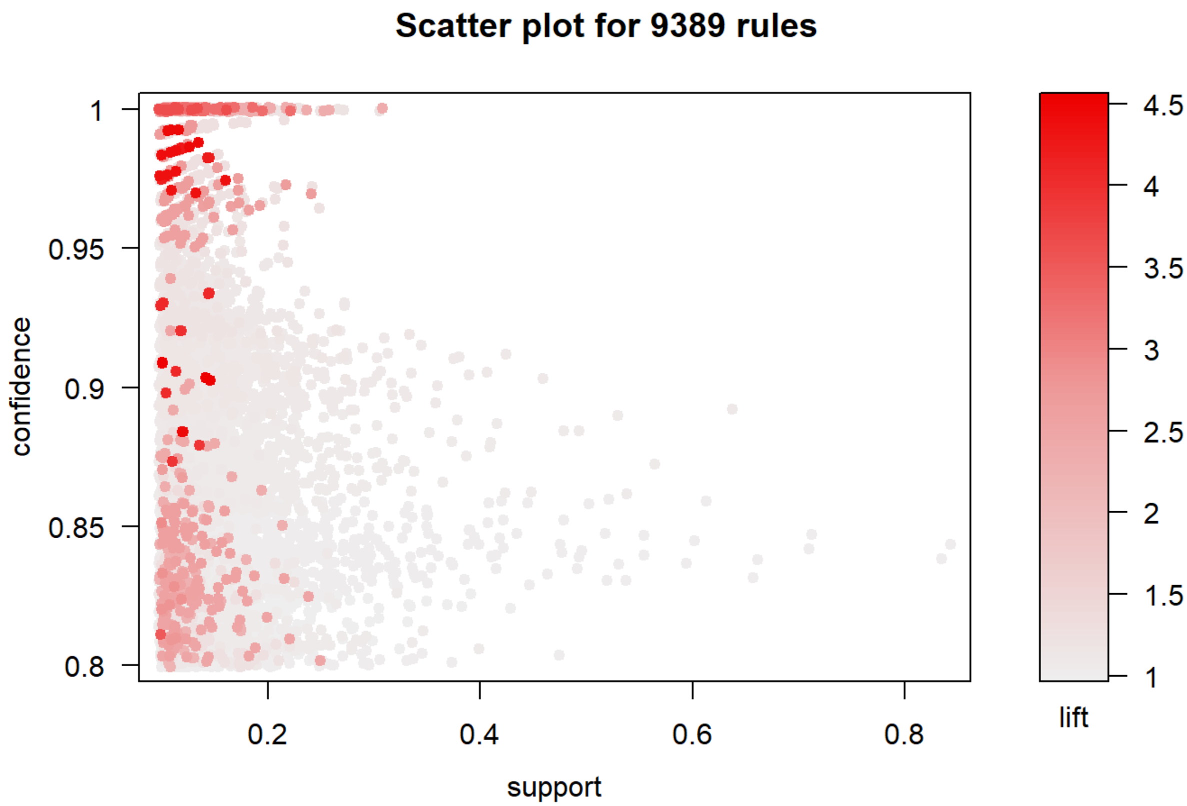
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 117
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[132 item(s), 1176 transaction(s)] done [0.01s].
## sorting and recoding items ... [99 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.07s].
## writing ... [9389 rule(s)] done [0.02s].
## creating S4 object  ... done [0.01s].
```

```
inspect(head(sort(Attrition_rules, by='confidence'),5))
```

```
##     lhs                                 rhs                                    support confidence
lift count
## [1] {JobRole=Manufacturing Director} => {Department=Research & Development} 0.1037415          1
1.539267   122
## [2] {EducationField=Marketing}       => {Department=Sales}                  0.1079932          1
3.284916   127
## [3] {PercentSalaryHike=<5%}          => {PerformanceRating=3}               0.1445578          1
1.193909   170
## [4] {PerformanceRating=4}            => {PercentSalaryHike=>20%}            0.1624150          1
3.618462   191
## [5] {YearsAtCompany=5<Years<10}      => {YearsSinceLastPromotion=<5years}   0.1675170          1
1.268608   197
```

```
plot(Attrition_rules)
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Analysis: From the above plot it is clear that, with decrease in support both the confidence and lift increases. Going forward, lets fine tune the function.

ARM fine tuned

```
Attrition_rules <- apriori(data=SS, parameter=list (supp=0.3,conf =0.5, minlen= 3, maxtime=10, targ
et = "rules"))
```

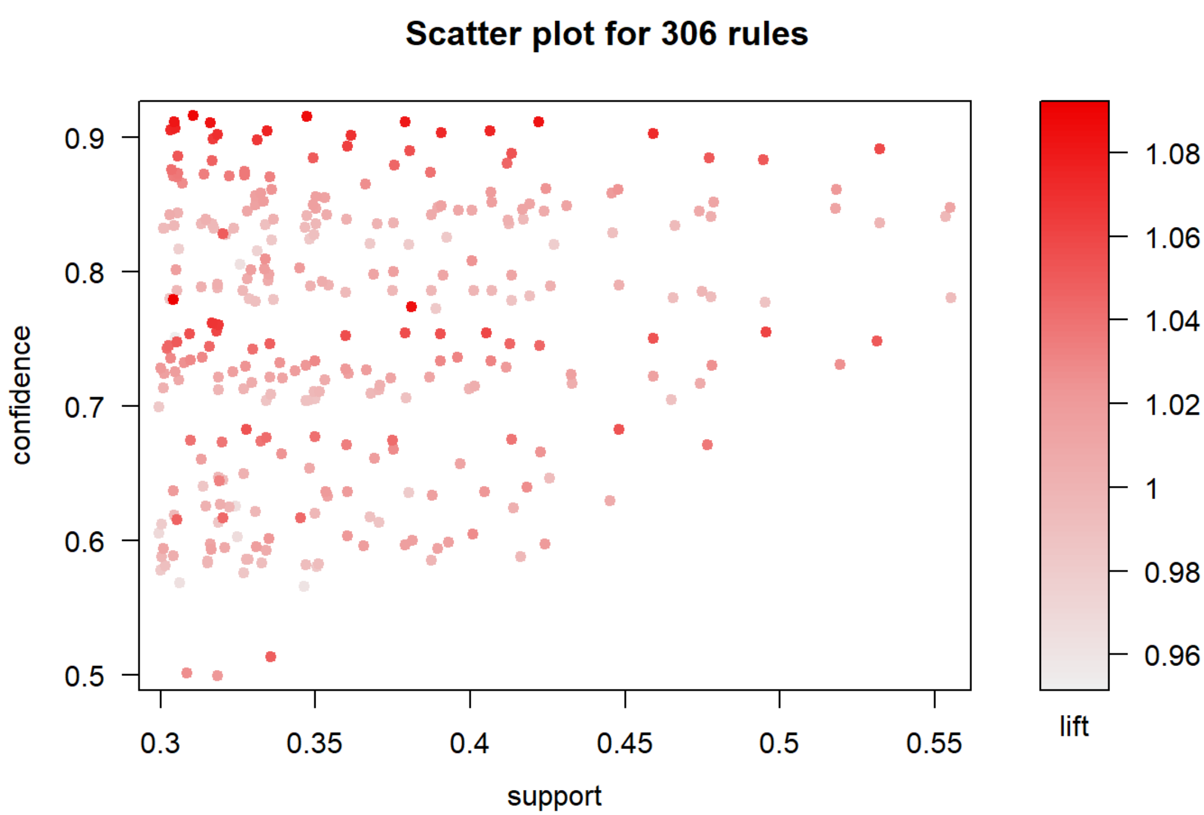
```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE      10     0.3      3
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 352
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[132 item(s), 1176 transaction(s)] done [0.01s].
## sorting and recoding items ... [36 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [306 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(Attrition_rules, by='confidence'),5))
```

```
##     lhs                                  rhs              support  confidence     lift  count
## [1] {BusinessTravel=Travel_Rarely,
##      Department=Research & Development,
##      OverTime=No}                     => {Attrition=No} 0.3103741   0.9170854 1.088287   365
## [2] {Department=Research & Development,
##      OverTime=No,
##      PerformanceRating=3}             => {Attrition=No} 0.3477891   0.9149888 1.085799   409
## [3] {Department=Research & Development,
##      OverTime=No}                     => {Attrition=No} 0.4226190   0.9119266 1.082165   497
## [4] {JobInvolvement=3,
##      OverTime=No,
##      PerformanceRating=3}             => {Attrition=No} 0.3163265   0.9117647 1.081973   372
## [5] {MaritalStatus=Married,
##      OverTime=No}                     => {Attrition=No} 0.3052721   0.9111675 1.081264   359
```

```
plot(Attrition_rules)
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Analysis: With the minimum Suppport and Confidence set to 0.5, we set the minimum rule length to 3 and maximum amount of time allowed to check for subsets to 10 we get 306 rules. Most of which are in to left corner the low support, high confidence and lift area

The goal of this assignment is to use the Association Rule Mining to predict when employee Attrition would be Yes / No. So, let us set the rhs to the values of the

Attrition variable and the target to “rules”

ARM to predict Attrition =Yes

```
Association_rules1 <- apriori(data=SS, parameter=list (supp=0.046,conf =0.25, minlen= 3, maxtime=19
, maxlen=7, target = "rules"), appearance = list(rhs=c("Attrition=Yes")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.25    0.1    1 none FALSE            TRUE      19   0.046      3
##  maxlen target   ext
##       7  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 54
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[132 item(s), 1176 transaction(s)] done [0.00s].
## sorting and recoding items ... [121 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7
```

```
## Warning in apriori(data = SS, parameter = list(supp = 0.046, conf =
## 0.25, : Mining stopped (maxlen reached). Only patterns up to a length of 7
## returned!
```

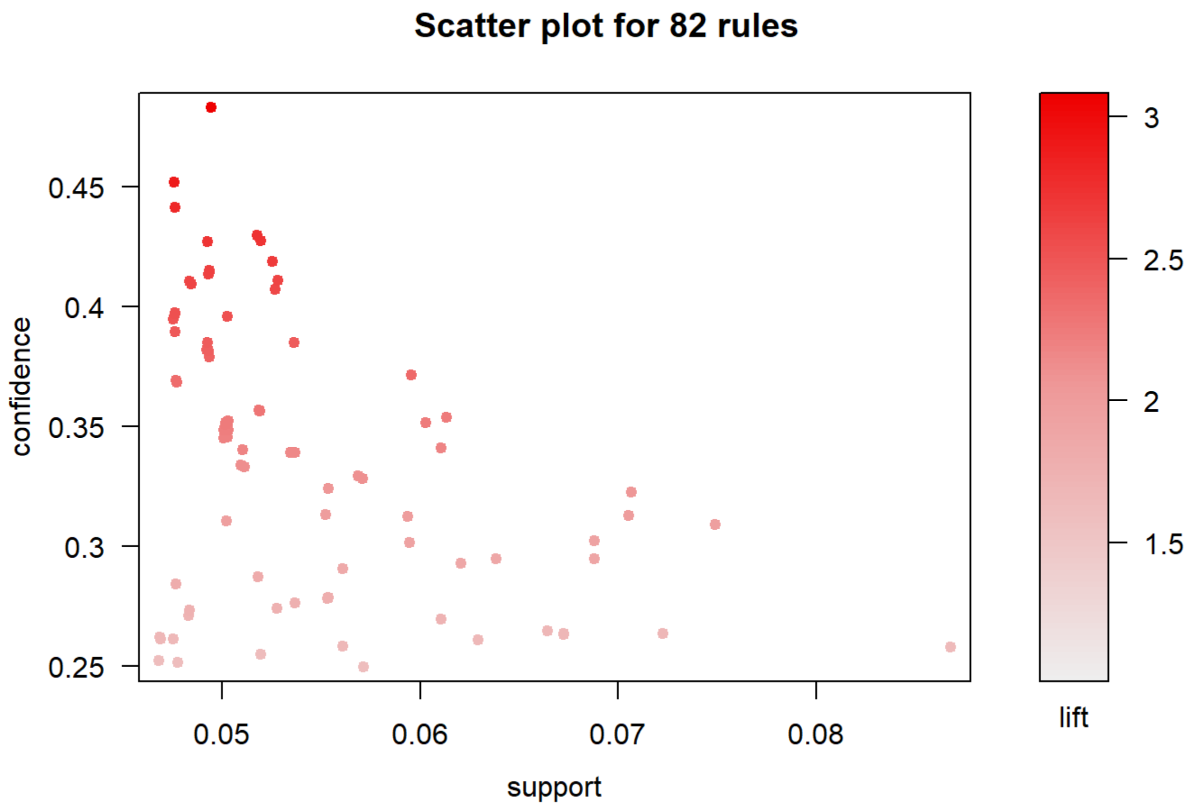
```
##  done [0.42s].
## writing ... [82 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].
```

```
inspect(head(sort(Association_rules1, by='confidence'),10))
```

```
##      lhs                                     rhs               support confidence     lift count
## [1]  {JobLevel=1,
##       OverTime=Yes}                       => {Attrition=Yes} 0.04931973  0.4833333 3.072432    58
## [2]  {JobLevel=1,
##       MonthlyIncome=low,
##       StockOptionLevel=0,
##       YearsSinceLastPromotion=<5years} => {Attrition=Yes} 0.04761905  0.4516129 2.870793    56
## [3]  {MonthlyIncome=low,
##       StockOptionLevel=0,
##       YearsSinceLastPromotion=<5years} => {Attrition=Yes} 0.04761905  0.4409449 2.802979    56
## [4]  {Age=young,
##       JobLevel=1,
##       MonthlyIncome=low,
##       YearsSinceLastPromotion=<5years} => {Attrition=Yes} 0.05187075  0.4295775 2.730719    61
## [5]  {Age=young,
##       MonthlyIncome=low,
##       YearsSinceLastPromotion=<5years} => {Attrition=Yes} 0.05187075  0.4265734 2.711624    61
## [6]  {JobLevel=1,
##       MonthlyIncome=low,
##       StockOptionLevel=0}              => {Attrition=Yes} 0.04931973  0.4264706 2.710970    58
## [7]  {OverTime=Yes,
##       StockOptionLevel=0}              => {Attrition=Yes} 0.05272109  0.4189189 2.662966    62
## [8]  {StockOptionLevel=0,
##       YearsAtCompany=<5years}          => {Attrition=Yes} 0.04931973  0.4142857 2.633514    58
## [9]  {MonthlyIncome=low,
##       StockOptionLevel=0}              => {Attrition=Yes} 0.04931973  0.4142857 2.633514    58
## [10] {StockOptionLevel=0,
##       YearsAtCompany=<5years,
##       YearsSinceLastPromotion=<5years} => {Attrition=Yes} 0.04931973  0.4142857 2.633514    58
```

```
plot(Association_rules1)
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Analysis: Keeping Rhs as attrition = yes we get 82 rules with maximum confidence as 0.4833 amd the coressponding support as 0.04931. The Employee will mostly tend towards attrition when job level =1/ overtime = yes / low monthly income / 0 stock option / years since last promotion is less than 5 years / new employee.

ARM to predict Attrition = NO

```
Association_rules2 <- apriori(data=SS, parameter=list (supp=0.25,conf =0.25, minlen= 3, maxtime=19,
maxlen=7, target = "rules"), appearance = list(rhs=c("Attrition=No")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.25    0.1    1 none FALSE            TRUE      19    0.25      3
##  maxlen target   ext
##       7  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 294
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[132 item(s), 1176 transaction(s)] done [0.00s].
## sorting and recoding items ... [59 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.01s].
## writing ... [118 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

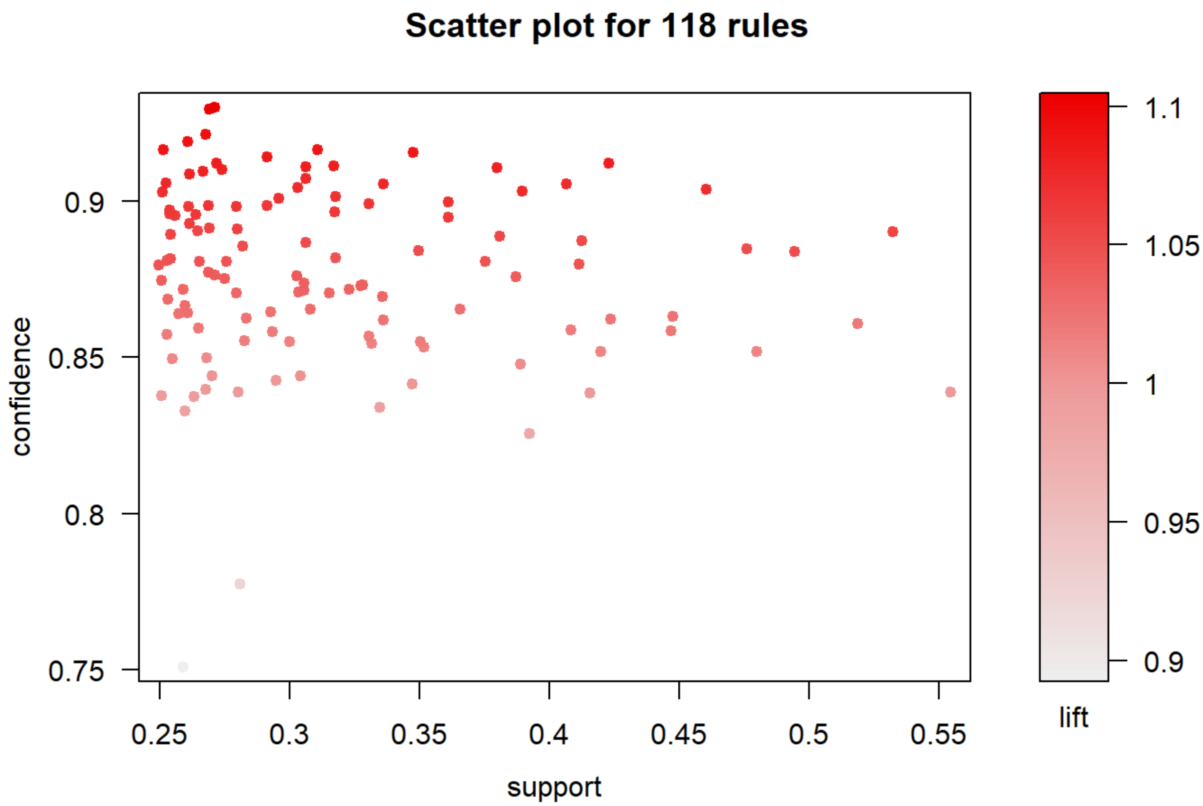
```
inspect(head(sort(Association_rules2, by='confidence'),10))
```

```
##      lhs                                  rhs              support confidence     lift count
## [1]  {OverTime=No,
##       StockOptionLevel=1}              => {Attrition=No} 0.2712585  0.9300292 1.103647   319
## [2]  {Department=Research & Development,
##       OverTime=No,
##       WorkLifeBalance=3}               => {Attrition=No} 0.2695578  0.9296188 1.103160   317
## [3]  {BusinessTravel=Travel_Rarely,
##       StockOptionLevel=1}              => {Attrition=No} 0.2678571  0.9210526 1.092995   315
## [4]  {BusinessTravel=Travel_Rarely,
##       Department=Research & Development,
##       OverTime=No,
##       PerformanceRating=3}             => {Attrition=No} 0.2619048  0.9194030 1.091037   308
## [5]  {BusinessTravel=Travel_Rarely,
##       Department=Research & Development,
##       OverTime=No}                     => {Attrition=No} 0.3103741  0.9170854 1.088287   365
## [6]  {MaritalStatus=Married,
##       StockOptionLevel=1}              => {Attrition=No} 0.2508503  0.9161491 1.087176   295
## [7]  {Department=Research & Development,
##       OverTime=No,
##       PerformanceRating=3}             => {Attrition=No} 0.3477891  0.9149888 1.085799   409
## [8]  {BusinessTravel=Travel_Rarely,
##       OverTime=No,
##       WorkLifeBalance=3}               => {Attrition=No} 0.2916667  0.9146667 1.085417   343
## [9]  {Department=Research & Development,
##       OverTime=No,

##       PerformanceRating=3,
##       YearsSinceLastPromotion=<5years}   => {Attrition=No} 0.2729592  0.9119318 1.082171   321
## [10] {Department=Research & Development,
##       OverTime=No}                     => {Attrition=No} 0.4226190  0.9119266 1.082165   497
```

```
plot(Association_rules2)
```

```
## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Analysis: Keeping Rhs as attrition = No we get 118 rules with maximum confidence as 0.93940 amd the coressponding support as 0.0271. The Employee will not tend towards attrition when overtime = no / high monthly income / good stock option / have good work life balance/ Business travel = rarely / department is Research and development / married.

#Shiny App:

Lets us now change the hyperparameters in apriori rules using Shiny App: (please put support below o.o5 for rules=yes)

https://akshaybhala.shinyapps.io/HW01/ (https://akshaybhala.shinyapps.io/HW01/)