# IST 707 Data Analytics HOMEWORK 1

## Rshiny Dashboard Link: https://bhavishkumar.shinyapps.io/IST707HW1/ (https://bhavishkumar.shinyapps.io/IST707HW1/)

```
setwd("C:/Users/bhavi/OneDrive/Desktop/SYR ADS/Sem 2/IST_707_Data_Analytics/HW")

getwd
```

```
## function ()
## .Internal(getwd())
## <bytecode: 0x0000000012ed2448>
## <environment: namespace:base>
```

```
emp_attr <- read.csv("employee_attrition.csv")
```

### Writing a function to obtain mode (most frequently occuring value)

*Match returns the first position of occurence of every element of 'x' in uniq These position match indexes are tabulated which generates number of times each index is present which.max provides the index of the maximum occuring match, i.e. the mode*

```
getmode <- function(x) {

  uniq <- unique(x)
  uniq[which.max(tabulate(match(x,uniq)))]
}
```

### Treating all missing values, by replacing with mean/median/mode

*Storing all missing values as NA 11 NAs in the table*

```
emp_attr[emp_attr == ""] <-NA
sum(is.na(emp_attr))
```

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

*1 NA in the Gender column Replace Gender NA with Mode of the column*

```
sum(is.na(emp_attr$Gender))
```

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

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

*1 NA in the OverTime column Replace OverTime NA with mode*

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

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

*2 NAs in DistanceFromHome Replace NA with mean*

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

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

*1 NA in JobLevel column Replace with mode*

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

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

*1 NA in percent salary hike Replace NA with mean*

```
emp_attr$PercentSalaryHike[is.na(emp_attr$PercentSalaryHike)] <- mean(emp_attr$PercentSalaryHike, na.rm = TRUE)
sum(is.na(emp_attr$PercentSalaryHike))
```

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

*1 NA in PerformanceRating Replace NA with mode*

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

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

*1 NA in RelationshipSatisfaction Replace NA with mode*

```
emp_attr$RelationshipSatisfaction[is.na(emp_attr$RelationshipSatisfaction)] <- getmode(emp_attr$RelationshipSatisfaction)
sum(is.na(emp_attr$RelationshipSatisfaction))
```

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

*2 NA in Total Working Years Replace NA with median*

```
emp_attr$TotalWorkingYears[is.na(emp_attr$TotalWorkingYears)] <- median(emp_attr$TotalWorkingYears, na.rm = TRUE)
sum(is.na(emp_attr$TotalWorkingYears))
```

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

*1 NA in YearsSinceLastPromotion*
*Replace NA with median*

```
emp_attr$YearsSinceLastPromotion[is.na(emp_attr$YearsSinceLastPromotion)] <- median(emp_attr$YearsSinceLastPromotion, na.rm
= TRUE)
sum(is.na(emp_attr$YearsSinceLastPromotion))
```

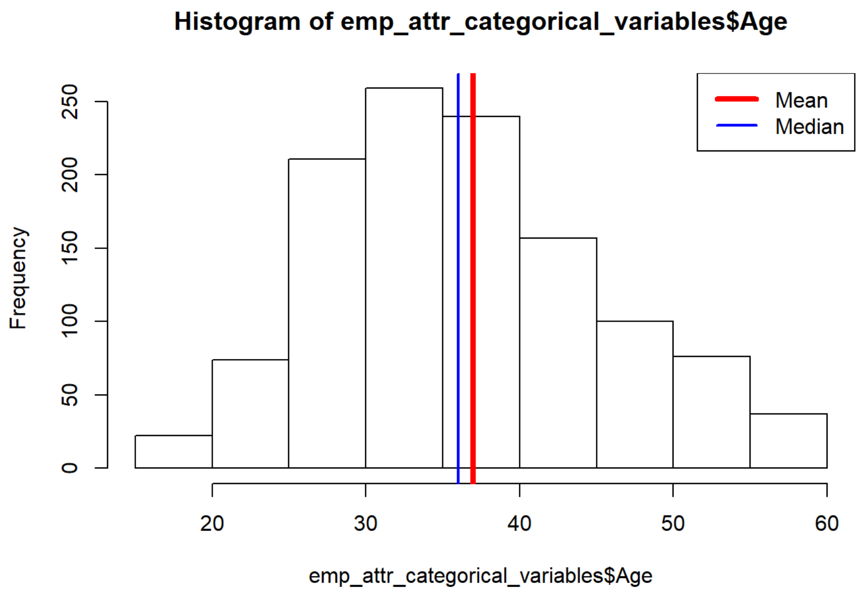
```
## [1] 0
```

```
emp_attr_categorical_variables<-emp_attr
```

## HISTOGRAMS on numeric columns to identify Outliers and look at frequency distribution and spread

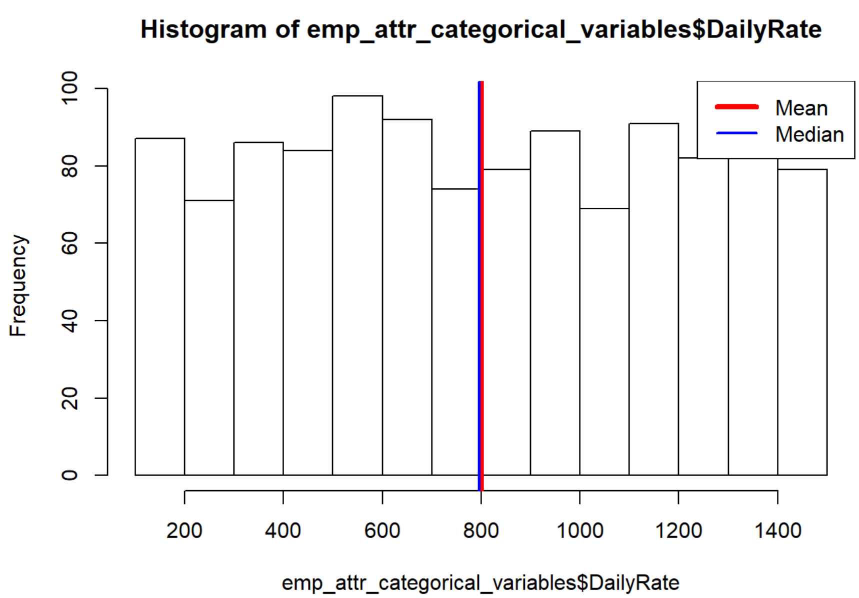
**HISTOGRAM of AGE column.Mean aproximately = Median, hence normal distribution**

```
hist(emp_attr_categorical_variables$Age)
abline(v=mean(emp_attr_categorical_variables$Age),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$Age),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```



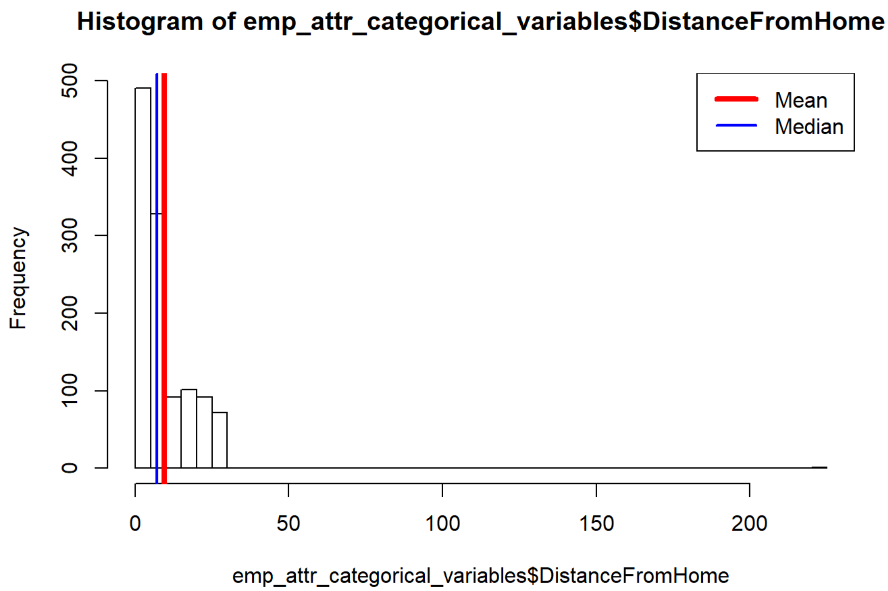

**HISTOGRAM of Daily Rate column. ## Mean = Median, with Uniform distribution**

```
hist(emp_attr_categorical_variables$DailyRate)
abline(v=mean(emp_attr_categorical_variables$DailyRate),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$DailyRate),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```



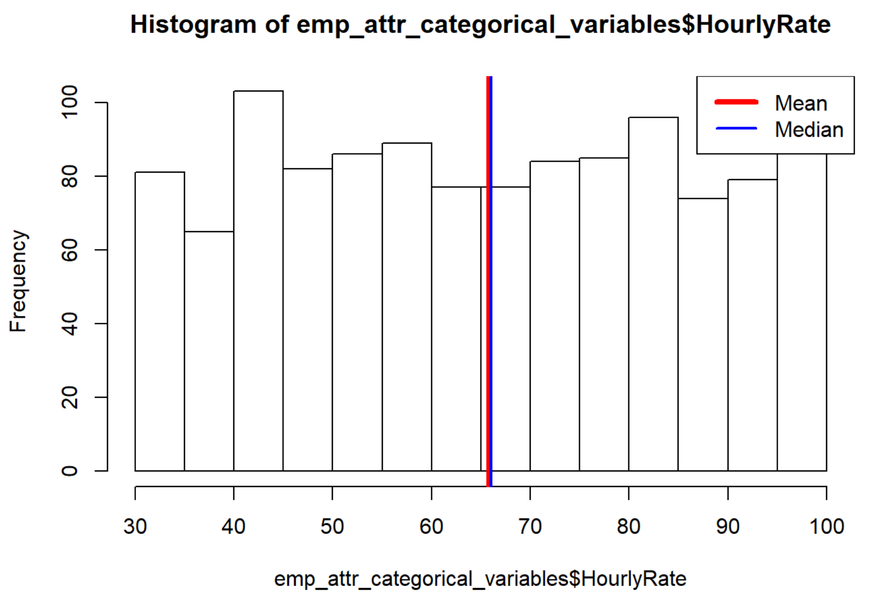

**HISTOGRAM of DistanceFromHome column. Mean > Median, hence right skewed distribution**

```
hist(emp_attr_categorical_variables$DistanceFromHome, breaks = 50)
abline(v=mean(emp_attr_categorical_variables$DistanceFromHome),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$DistanceFromHome),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```



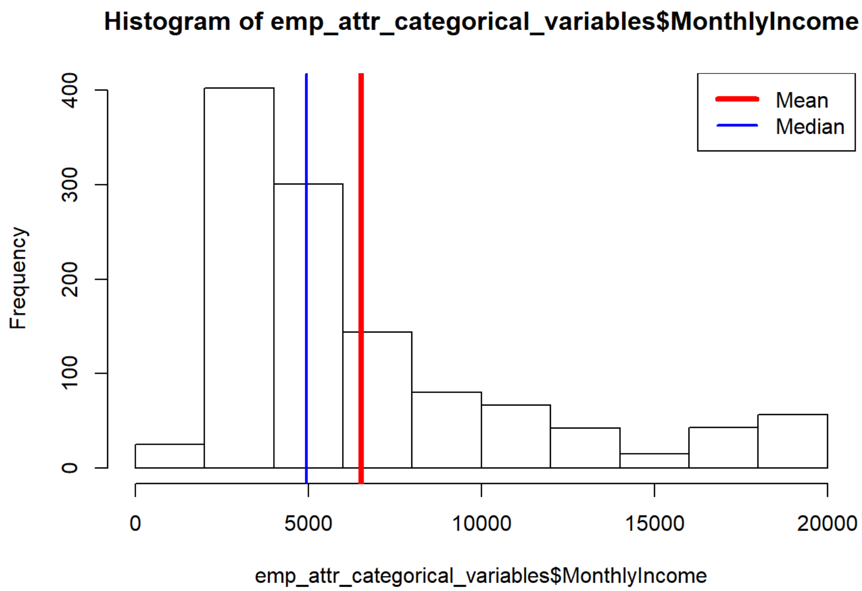

**HISTOGRAM of HourlyRate column. Mean = Median, with uniform distribution**

```
hist(emp_attr_categorical_variables$HourlyRate)
abline(v=mean(emp_attr_categorical_variables$HourlyRate),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$HourlyRate),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```



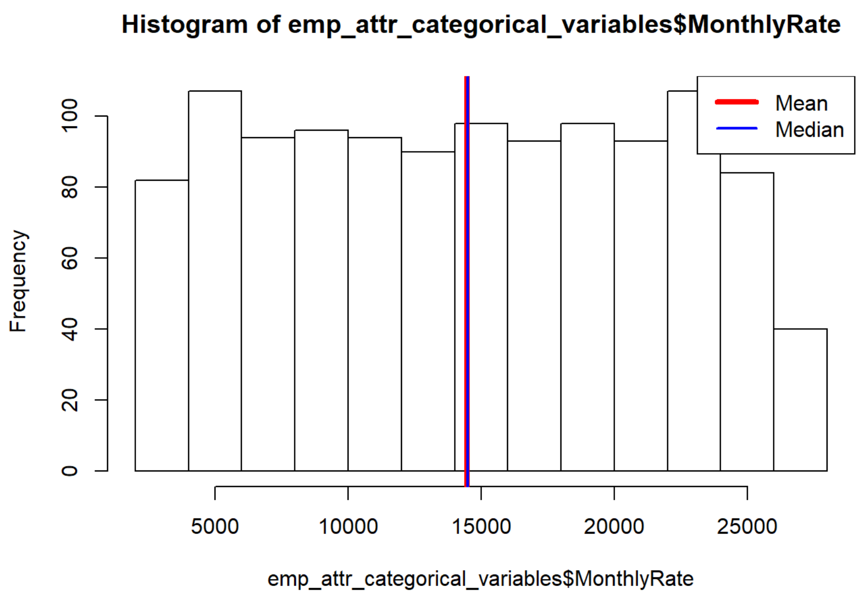

**HISTOGRAM of MonthlyIncome column. Mean > Median, Right Skewed with possibility of outliers**

```
hist(emp_attr_categorical_variables$MonthlyIncome)
abline(v=mean(emp_attr_categorical_variables$MonthlyIncome),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$MonthlyIncome),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```



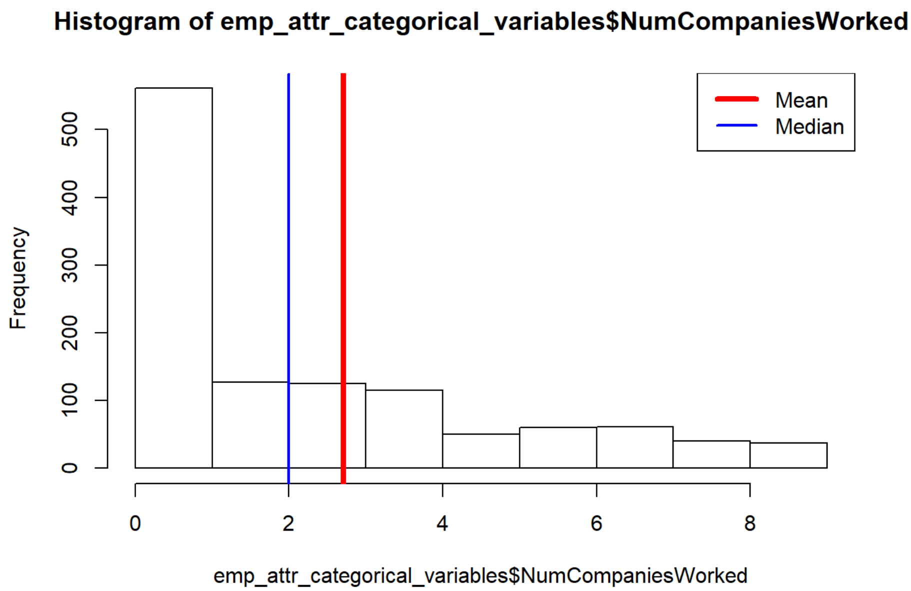

**HISTOGRAM of MonthlyRate column. Mean = Median with Uniform Distribution.**

```
hist(emp_attr_categorical_variables$MonthlyRate)
abline(v=mean(emp_attr_categorical_variables$MonthlyRate),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$MonthlyRate),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```



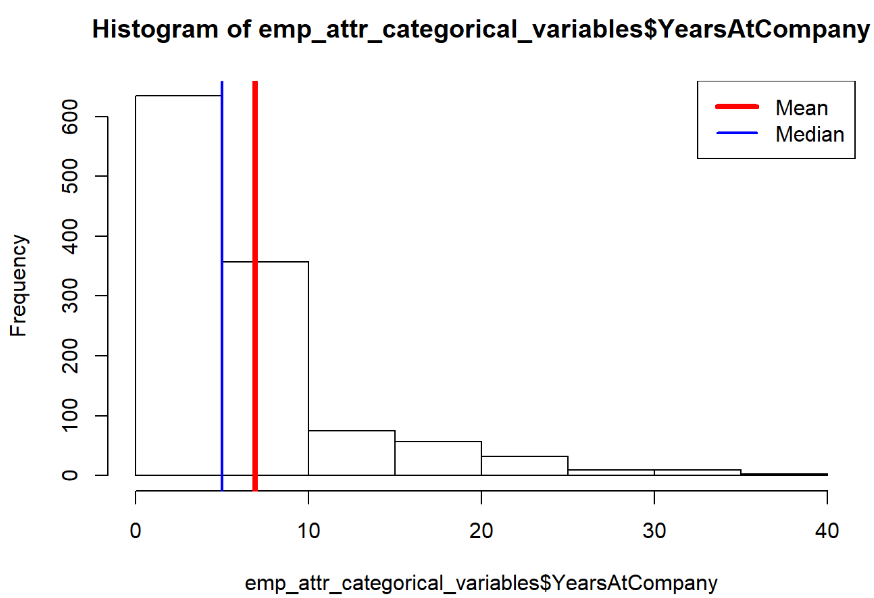

**HISTOGRAM of NumCompaniesWorked column. Mean > Median with right skewed Distribution.**

```
hist(emp_attr_categorical_variables$NumCompaniesWorked)
abline(v=mean(emp_attr_categorical_variables$NumCompaniesWorked),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$NumCompaniesWorked),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```




**Histogram of YearsAtCompany column. Mean > Median with right skewed Distribution**

```
hist(emp_attr_categorical_variables$YearsAtCompany)
abline(v=mean(emp_attr_categorical_variables$YearsAtCompany),col = "red",lwd = 4)
abline(v=median(emp_attr_categorical_variables$YearsAtCompany),col = "blue", lwd = 2)
legend(x = "topright",
       c("Mean", "Median"),
       col = c("red", "blue"),lwd = c(4,2))
```




## Binning of numeric continuous variables to ordinal categorical variables for EDA & Association Rules creation

**Creating Age Groups**

```
library(stringr)
emp_attr_categorical_variables$age_groups <-cut(emp_attr_categorical_variables$Age, breaks = c(quantile(emp_attr_categorical_variables$Age, probs = c(0,0.25,0.5,0.75,1))),
     labels = c(str_c(quantile(emp_attr_categorical_variables$Age,probs = 0),quantile(emp_attr_categorical_variables$Age,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$Age,probs = 0.25),quantile(emp_attr_categorical_variables$Age,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$Age,probs = 0.5),quantile(emp_attr_categorical_variables$Age,probs = 0.75),sep = " to "),str_c(quantile(emp_attr_categorical_variables$Age,probs = 0.75),quantile(emp_attr_categorical_variables$Age,probs = 1),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$age_groups<-as.factor(emp_attr_categorical_variables$age_groups)
unique(emp_attr_categorical_variables$age_groups)
```

```
## [1] 30 to 36 43 to 60 36 to 43 18 to 30
## Levels: 18 to 30 30 to 36 36 to 43 43 to 60
```

**Creating Daily Rate Groups**

```
emp_attr_categorical_variables$DailyRate_groups <-cut(emp_attr_categorical_variables$DailyRate, breaks = c(quantile(emp_attr_categorical_variables$DailyRate, probs = c(0,0.25,0.5,0.75,1))),
  labels = c(str_c(quantile(emp_attr_categorical_variables$DailyRate,probs = 0),quantile(emp_attr_categorical_variables$DailyRate,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$DailyRate,probs = 0.25),quantile(emp_attr_categorical_variables$DailyRate,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$DailyRate,probs = 0.5),quantile(emp_attr_categorical_variables$DailyRate,probs = 0.75),sep = " to "),str_c(quantile(emp_attr_categorical_variables$DailyRate,probs = 0.75),quantile(emp_attr_categorical_variables$DailyRate,probs = 1),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$DailyRate_groups<-as.factor(emp_attr_categorical_variables$DailyRate_groups)
unique(emp_attr_categorical_variables$DailyRate_groups)
```

```
## [1] 1162 to 1499  461.75 to 796 102 to 461.75 796 to 1162
## Levels: 102 to 461.75 461.75 to 796 796 to 1162 1162 to 1499
```

**Creating DistanceFromHome Groups**

```
emp_attr_categorical_variables$DistanceFromHome_groups <-cut(emp_attr_categorical_variables$DistanceFromHome, breaks = c(quantile(emp_attr_categorical_variables$DistanceFromHome, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0),quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0.25),quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0.5),quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0.75),sep = " to "),str_c(quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0.75),quantile(emp_attr_categorical_variables$DistanceFromHome,probs = 0.99),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$DistanceFromHome_groups <- as.factor(emp_attr_categorical_variables$DistanceFromHome_groups)
unique(emp_attr_categorical_variables$DistanceFromHome_groups)
```

```
## [1] 14 to 29 7 to 14  1 to 2   2 to 7
## Levels: 1 to 2 2 to 7 7 to 14 14 to 29
```

**Creating HourlyRate Groups**

```
emp_attr_categorical_variables$HourlyRate_groups <- cut(emp_attr_categorical_variables$HourlyRate,breaks = c(quantile(emp_attr_categorical_variables$HourlyRate, probs = c(0,0.25,0.5,0.75,1))),
  labels = c(str_c(quantile(emp_attr_categorical_variables$HourlyRate,probs = 0),quantile(emp_attr_categorical_variables$HourlyRate,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$HourlyRate,probs = 0.25),quantile(emp_attr_categorical_variables$HourlyRate,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$HourlyRate,probs = 0.5),quantile(emp_attr_categorical_variables$HourlyRate,probs = 0.75),sep = " to "),str_c(quantile(emp_attr_categorical_variables$HourlyRate,probs = 0.75),quantile(emp_attr_categorical_variables$HourlyRate,probs = 1),sep = " to ")), right = FALSE,include.lowest=TRUE)
emp_attr_categorical_variables$HourlyRate_groups <- as.factor(emp_attr_categorical_variables$HourlyRate_groups)
unique(emp_attr_categorical_variables$HourlyRate_groups)
```

```
## [1] 83 to 100 66 to 83  30 to 48  48 to 66
## Levels: 30 to 48 48 to 66 66 to 83 83 to 100
```

**Creating MonthlyIncome Groups**

```
emp_attr_categorical_variables$MonthlyIncome_groups <- cut(emp_attr_categorical_variables$MonthlyIncome,breaks = c(quantile(emp_attr_categorical_variables$MonthlyIncome, probs = c(0,0.25,0.5,0.75,1))),
  labels = c(str_c(quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 0),quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 0.25),quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 0.5),quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 0.75),sep = " to "),str_c(quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 0.75),quantile(emp_attr_categorical_variables$MonthlyIncome,probs = 1),sep = " to ")), right = FALSE,include.lowest=TRUE)
emp_attr_categorical_variables$MonthlyIncome_groups <- as.factor(emp_attr_categorical_variables$MonthlyIncome_groups)
unique(emp_attr_categorical_variables$MonthlyIncome_groups)
```

```
## [1] 4950.5 to 8354.5 2954.5 to 4950.5 8354.5 to 19973  1009 to 2954.5
## Levels: 1009 to 2954.5 2954.5 to 4950.5 4950.5 to 8354.5 8354.5 to 19973
```

**Creating MonthlyRate Groups**

```
emp_attr_categorical_variables$MonthlyRate_groups <- cut(emp_attr_categorical_variables$MonthlyRate,breaks = c(quantile(emp_
attr_categorical_variables$MonthlyRate, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$MonthlyRate,probs = 0),quantile(emp_attr_categorical_variables$Mont
hlyRate,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$MonthlyRate,probs = 0.25),quantile(emp_att
r_categorical_variables$MonthlyRate,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$MonthlyRate,pro
bs = 0.5),quantile(emp_attr_categorical_variables$MonthlyRate,probs = 0.75),sep = " to "),str_c(quantile(emp_attr_categorica
l_variables$MonthlyRate,probs = 0.75),quantile(emp_attr_categorical_variables$MonthlyRate,probs = 1),sep = " to ")), right =
FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$MonthlyRate_groups <- as.factor(emp_attr_categorical_variables$MonthlyRate_groups)
unique(emp_attr_categorical_variables$MonthlyRate_groups)
```

```
## [1] 2094 to 8275      20627.25 to 26999 14488 to 20627.25 8275 to 14488
## Levels: 2094 to 8275 8275 to 14488 14488 to 20627.25 20627.25 to 26999
```

**Creating NumCompaniesWorked Groups**

```
emp_attr_categorical_variables$NumCompaniesWorked_groups <- cut(emp_attr_categorical_variables$NumCompaniesWorked,breaks = c
(quantile(emp_attr_categorical_variables$NumCompaniesWorked, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$NumCompaniesWorked,probs = 0),quantile(emp_attr_categorical_variabl
es$NumCompaniesWorked,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$NumCompaniesWorked,probs =
0.25),quantile(emp_attr_categorical_variables$NumCompaniesWorked,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categori
cal_variables$NumCompaniesWorked,probs = 0.5),quantile(emp_attr_categorical_variables$NumCompaniesWorked,probs = 0.75),sep =
" to "),str_c(quantile(emp_attr_categorical_variables$NumCompaniesWorked,probs = 0.75),quantile(emp_attr_categorical_variabl
es$NumCompaniesWorked,probs = 1),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$NumCompaniesWorked_groups <- as.factor(emp_attr_categorical_variables$NumCompaniesWorked_grou
ps)
unique(emp_attr_categorical_variables$NumCompaniesWorked_groups)
```

```
## [1] 4 to 9 1 to 2 2 to 4 0 to 1
## Levels: 0 to 1 1 to 2 2 to 4 4 to 9
```

**Creating PercentSalaryHike Groups**

```
emp_attr_categorical_variables$PercentSalaryHike_groups <- cut(emp_attr_categorical_variables$PercentSalaryHike,breaks = c(q
uantile(emp_attr_categorical_variables$PercentSalaryHike, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$PercentSalaryHike,probs = 0),quantile(emp_attr_categorical_variable
s$PercentSalaryHike,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$PercentSalaryHike,probs = 0.25
),quantile(emp_attr_categorical_variables$PercentSalaryHike,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_v
ariables$PercentSalaryHike,probs = 0.5),quantile(emp_attr_categorical_variables$PercentSalaryHike,probs = 0.75),sep = " to "
),str_c(quantile(emp_attr_categorical_variables$PercentSalaryHike,probs = 0.75),quantile(emp_attr_categorical_variables$Perc
entSalaryHike,probs = 1),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$PercentSalaryHike_groups <- as.factor(emp_attr_categorical_variables$PercentSalaryHike_group
s)
unique(emp_attr_categorical_variables$PercentSalaryHike_groups)
```

```
## [1] 14 to 18 18 to 25 12 to 14 11 to 12
## Levels: 11 to 12 12 to 14 14 to 18 18 to 25
```

**Creating TotalWorkingYears Groups**

```
emp_attr_categorical_variables$TotalWorkingYears_groups <- cut(emp_attr_categorical_variables$TotalWorkingYears,breaks = c(q
uantile(emp_attr_categorical_variables$TotalWorkingYears, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$TotalWorkingYears,probs = 0),quantile(emp_attr_categorical_variable
s$TotalWorkingYears,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$TotalWorkingYears,probs = 0.25
),quantile(emp_attr_categorical_variables$TotalWorkingYears,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_v
ariables$TotalWorkingYears,probs = 0.5),quantile(emp_attr_categorical_variables$TotalWorkingYears,probs = 0.75),sep = " to "
),str_c(quantile(emp_attr_categorical_variables$TotalWorkingYears,probs = 0.75),quantile(emp_attr_categorical_variables$Tota
lWorkingYears,probs = 0.99),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$TotalWorkingYears_groups <- as.factor(emp_attr_categorical_variables$TotalWorkingYears_group
s)
unique(emp_attr_categorical_variables$TotalWorkingYears_groups)
```

```
## [1] 0 to 6      6 to 10     10 to 15    15 to 35.25
## Levels: 0 to 6 6 to 10 10 to 15 15 to 35.25
```

**Creating YearsAtCompany Groups**

```
emp_attr_categorical_variables$YearsAtCompany_groups <- cut(emp_attr_categorical_variables$YearsAtCompany,breaks = c(quantil
e(emp_attr_categorical_variables$YearsAtCompany, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$YearsAtCompany,probs = 0),quantile(emp_attr_categorical_variables$Y
earsAtCompany,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$YearsAtCompany,probs = 0.25),quantil
e(emp_attr_categorical_variables$YearsAtCompany,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$Yea
rsAtCompany,probs = 0.5),quantile(emp_attr_categorical_variables$YearsAtCompany,probs = 0.75),sep = " to "),str_c(quantile(e
mp_attr_categorical_variables$YearsAtCompany,probs = 0.75),quantile(emp_attr_categorical_variables$YearsAtCompany,probs = 1
),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$YearsAtCompany_groups <- as.factor(emp_attr_categorical_variables$YearsAtCompany_groups)
unique(emp_attr_categorical_variables$YearsAtCompany_groups)
```

```
## [1] 0 to 3  5 to 9  9 to 40 3 to 5
## Levels: 0 to 3 3 to 5 5 to 9 9 to 40
```

**Creating YearsInCurrentRole Groups**

```
emp_attr_categorical_variables$YearsInCurrentRole_groups <- cut(emp_attr_categorical_variables$YearsInCurrentRole,breaks = c
(quantile(emp_attr_categorical_variables$YearsInCurrentRole, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$YearsInCurrentRole,probs = 0),quantile(emp_attr_categorical_variabl
es$YearsInCurrentRole,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$YearsInCurrentRole,probs =
0.25),quantile(emp_attr_categorical_variables$YearsInCurrentRole,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categori
cal_variables$YearsInCurrentRole,probs = 0.5),quantile(emp_attr_categorical_variables$YearsInCurrentRole,probs = 0.75),sep =
" to "),str_c(quantile(emp_attr_categorical_variables$YearsInCurrentRole,probs = 0.75),quantile(emp_attr_categorical_variabl
es$YearsInCurrentRole,probs = 1),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$YearsInCurrentRole_groups <- as.factor(emp_attr_categorical_variables$YearsInCurrentRole_grou
ps)
unique(emp_attr_categorical_variables$YearsInCurrentRole_groups)
```

```
## [1] 0 to 2  2 to 3  3 to 7  7 to 18
## Levels: 0 to 2 2 to 3 3 to 7 7 to 18
```

**Creating YearsSinceLastPromotion Groups**

```
emp_attr_categorical_variables$YearsSinceLastPromotion_groups <- cut(emp_attr_categorical_variables$YearsSinceLastPromotion,
breaks = c(quantile(emp_attr_categorical_variables$YearsSinceLastPromotion, probs = c(0,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$YearsSinceLastPromotion,probs = 0),quantile(emp_attr_categorical_va
riables$YearsSinceLastPromotion,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_categorical_variables$YearsSinceLastPromo
tion,probs = 0.5),quantile(emp_attr_categorical_variables$YearsSinceLastPromotion,probs = 0.75),sep = " to "),str_c(quantile
(emp_attr_categorical_variables$YearsSinceLastPromotion,probs = 0.75),quantile(emp_attr_categorical_variables$YearsSinceLast
Promotion,probs = 1),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$YearsSinceLastPromotion_groups <- as.factor(emp_attr_categorical_variables$YearsSinceLastProm
otion_groups)
unique(emp_attr_categorical_variables$YearsSinceLastPromotion_groups)
```

```
## [1] 2 to 15 1 to 2  0 to 1
## Levels: 0 to 1 1 to 2 2 to 15
```

**Creating YearsWithCurrManager Groups**

```
emp_attr_categorical_variables$YearsWithCurrManager_groups <- cut(emp_attr_categorical_variables$YearsWithCurrManager,breaks
= c(quantile(emp_attr_categorical_variables$YearsWithCurrManager, probs = c(0,0.25,0.5,0.75,1))),
labels = c(str_c(quantile(emp_attr_categorical_variables$YearsWithCurrManager,probs = 0),quantile(emp_attr_categorical_varia
bles$YearsWithCurrManager,probs = 0.25),sep = " to "),str_c(quantile(emp_attr_categorical_variables$YearsWithCurrManager,pro
bs = 0.25),quantile(emp_attr_categorical_variables$YearsWithCurrManager,probs = 0.5),sep = " to "),str_c(quantile(emp_attr_c
ategorical_variables$YearsWithCurrManager,probs = 0.5),quantile(emp_attr_categorical_variables$YearsWithCurrManager,probs =
0.75),sep = " to "),str_c(quantile(emp_attr_categorical_variables$YearsWithCurrManager,probs = 0.75),quantile(emp_attr_categ
orical_variables$YearsWithCurrManager,probs = 0.99),sep = " to ")), right = FALSE, include.lowest=TRUE)
emp_attr_categorical_variables$YearsWithCurrManager_groups <- as.factor(emp_attr_categorical_variables$YearsWithCurrManager_
groups)
unique(emp_attr_categorical_variables$YearsWithCurrManager_groups)
```

```
## [1] 2 to 3  3 to 7  0 to 2  7 to 15
## Levels: 0 to 2 2 to 3 3 to 7 7 to 15
```

*CAN EXCLUDE EmployeeNumber, EmployeeCount, Over18 & StandardHours variables as they have 0 variance*

```
emp_attr_categorical_variables$EmployeeNumber <- NULL
emp_attr_categorical_variables$EmployeeCount <- NULL
emp_attr_categorical_variables$Over18 <- NULL
emp_attr_categorical_variables$StandardHours <- NULL
```

# PERFORM EDA to identify interesting trends in data

```
library(dplyr)
```

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

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

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

```
## -- Attaching packages ---------------------------------------------------------------------------------------------------
------- tidyverse 1.3.0 --
```

```
## <U+2713> ggplot2 3.2.1     <U+2713> readr   1.3.1
## <U+2713> tibble  2.1.3     <U+2713> purrr   0.3.3
## <U+2713> tidyr   1.0.0     <U+2713> forcats 0.4.0
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

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

```
library(plotly)
```

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

```
##
## Attaching package: 'plotly'
```

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

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(ggplot2)
```

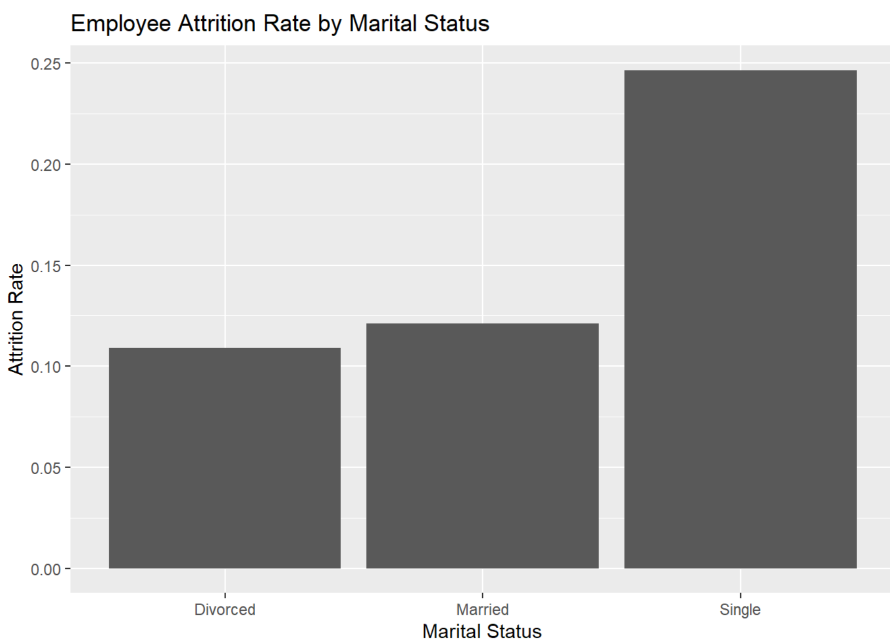
## HYPOTHESIS

## 1. Attrition Rate is very low amongst married/divorced employees and very high amongst single employees

```
df_marital_status <- emp_attr_categorical_variables %>%
  group_by(MaritalStatus,Attrition) %>%
  summarise(count = n()) %>%
    spread(Attrition,count) %>%
    ungroup %>%
    transmute(MaritalStatus=MaritalStatus,attr_rate = Yes/(Yes+No))
  df_marital_status
```

```
## # A tibble: 3 x 2
##   MaritalStatus attr_rate
##   <fct>             <dbl>
## 1 Divorced          0.109
## 2 Married           0.121
## 3 Single            0.247
```

```
  colnames(df_marital_status) <- c("Marital_Status","Attrition_Rate")
marital_status_attrition_plot <- ggplot(df_marital_status,aes(x=Marital_Status,y=Attrition_Rate))+
  geom_col(show.legend=TRUE,position = "dodge")+
  xlab("Marital Status")+ ylab("Attrition Rate")+
  ggtitle("Employee Attrition Rate by Marital Status")
marital_status_attrition_plot
```

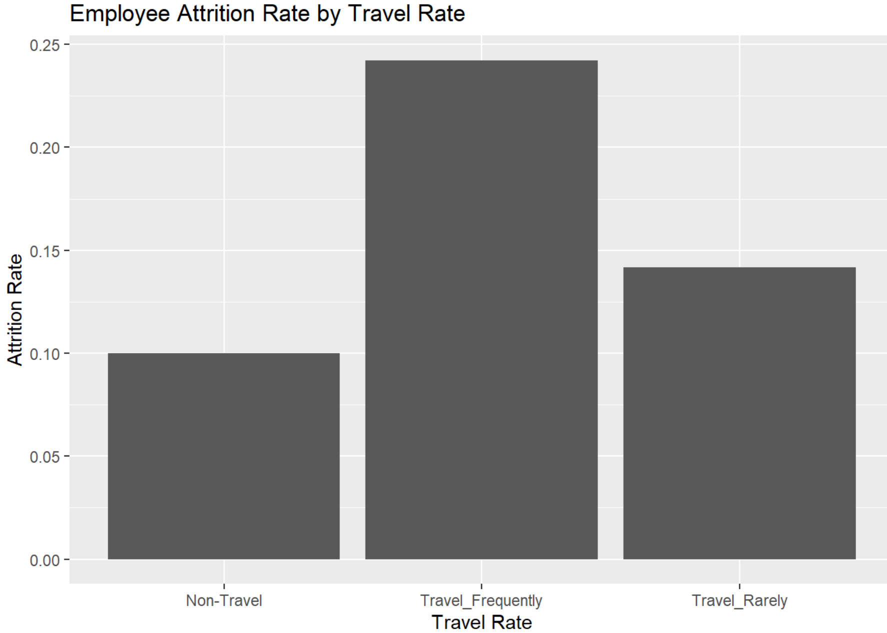



## 2. Attrition Rate is very low amongst Non Travellers and higher amongst frequent travellers

```
df_travel <- emp_attr_categorical_variables %>%
  group_by(BusinessTravel,Attrition) %>%
  summarise(count = n()) %>%
  spread(Attrition,count) %>%
  ungroup %>%
  transmute(BusinessTravel=BusinessTravel,attr_rate = Yes/(Yes+No))
df_travel
```

```
## # A tibble: 3 x 2
##   BusinessTravel    attr_rate
##   <fct>                 <dbl>
## 1 Non-Travel            0.1
## 2 Travel_Frequently     0.242
## 3 Travel_Rarely         0.142
```

```
colnames(df_travel) <- c("BusinessTravel","Attrition_Rate")
business_travel_attrition_plot <- ggplot(df_travel,aes(x=BusinessTravel,y=Attrition_Rate))+
  geom_col(show.legend=TRUE,position = "dodge")+
  xlab("Travel Rate")+ ylab("Attrition Rate")+
  ggtitle("Employee Attrition Rate by Travel Rate")
business_travel_attrition_plot
```

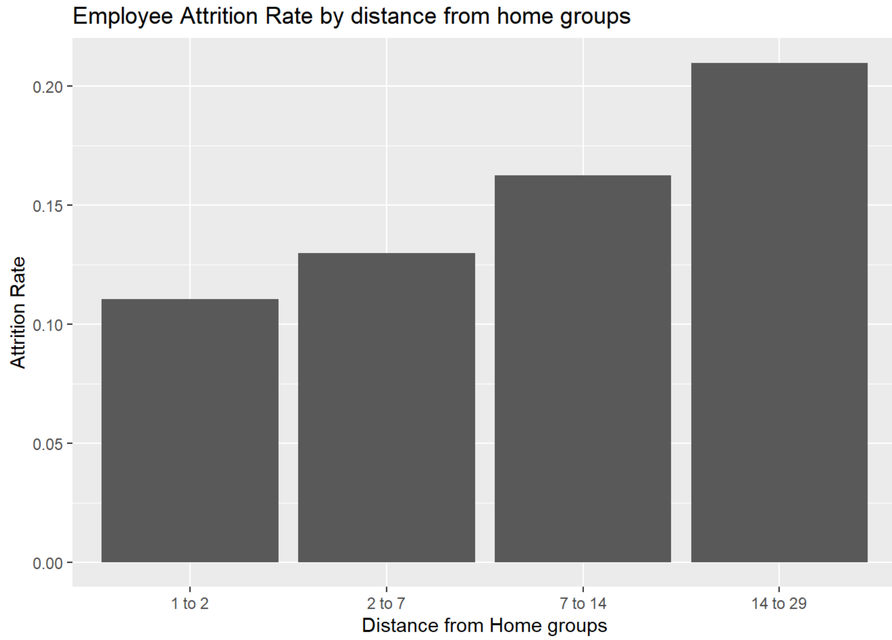



## 3. Attrition rate is lower amongst employees living near by and higher amongst employees staying far off

```
df_distance <- emp_attr_categorical_variables %>%
  group_by(DistanceFromHome_groups,Attrition) %>%
  summarise(count = n()) %>%
  spread(Attrition,count) %>%
  ungroup %>%
  transmute(DistanceFromHome_groups=DistanceFromHome_groups,attr_rate = Yes/(Yes+No))
df_distance
```

```
## # A tibble: 4 x 2
##   DistanceFromHome_groups attr_rate
##   <fct>                       <dbl>
## 1 1 to 2                      0.110
## 2 2 to 7                      0.130
## 3 7 to 14                     0.163
## 4 14 to 29                    0.210
```

```
colnames(df_distance) <- c("DistanceFromHome_groups","Attrition_Rate")
distance_attrition_plot <- ggplot(df_distance,aes(x=DistanceFromHome_groups,y=Attrition_Rate))+
  geom_col(show.legend=TRUE,position = "dodge")+
  xlab("Distance from Home groups")+ ylab("Attrition Rate")+
  ggtitle("Employee Attrition Rate by distance from home groups")
distance_attrition_plot
```

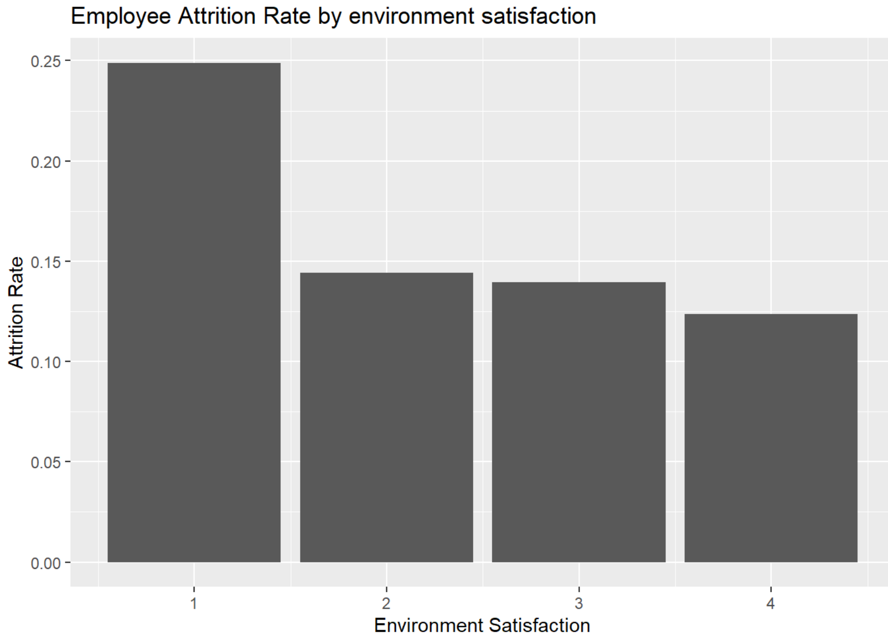



## 4. Attrition Rate is lower amongst employees with high environment satisfaction and is higher amongst employees with low environment satisfaction

```
df_envt <- emp_attr_categorical_variables %>%
  group_by(EnvironmentSatisfaction,Attrition) %>%
  summarise(count = n()) %>%
  spread(Attrition,count) %>%
  ungroup %>%
  transmute(EnvironmentSatisfaction=EnvironmentSatisfaction,attr_rate = Yes/(Yes+No))
df_envt
```

```
## # A tibble: 4 x 2
##   EnvironmentSatisfaction attr_rate
##                     <int>     <dbl>
## 1                       1     0.249
## 2                       2     0.144
## 3                       3     0.139
## 4                       4     0.124
```

```
colnames(df_envt) <- c("EnvironmentSatisfaction","Attrition_Rate")
envt_attrition_plot <- ggplot(df_envt,aes(x=EnvironmentSatisfaction,y=Attrition_Rate))+
  geom_col(show.legend=TRUE,position = "dodge")+
  xlab("Environment Satisfaction")+ ylab("Attrition Rate")+
  ggtitle("Employee Attrition Rate by environment satisfaction")
envt_attrition_plot
```

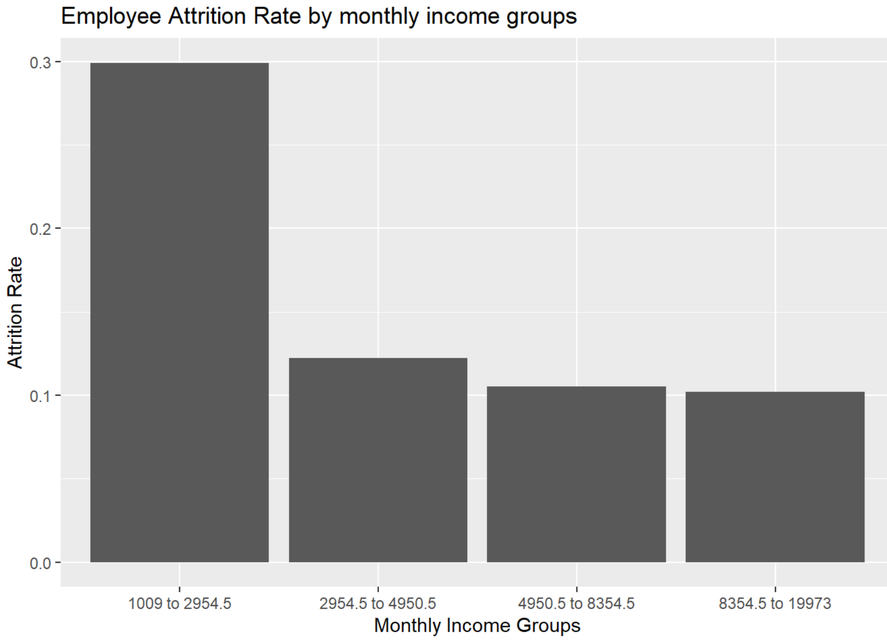



## 5. Attrition Rate is lower amongst high Monthly Income groups

```
df_monthly_income <- emp_attr_categorical_variables %>%
  group_by(MonthlyIncome_groups,Attrition) %>%
  summarise(count = n()) %>%
  spread(Attrition,count) %>%
  ungroup %>%
  transmute(MonthlyIncome_groups=MonthlyIncome_groups,attr_rate = Yes/(Yes+No))
df_monthly_income
```

```
## # A tibble: 4 x 2
##   MonthlyIncome_groups attr_rate
##   <fct>                    <dbl>
## 1 1009 to 2954.5           0.299
## 2 2954.5 to 4950.5         0.122
## 3 4950.5 to 8354.5         0.105
## 4 8354.5 to 19973          0.102
```

```
colnames(df_monthly_income) <- c("MonthlyIncome_groups","Attrition_Rate")
monthly_income_attrition_plot <- ggplot(df_monthly_income,aes(x=MonthlyIncome_groups,y=Attrition_Rate))+
  geom_col(show.legend=TRUE,position = "dodge")+
  xlab("Monthly Income Groups")+ ylab("Attrition Rate")+
  ggtitle("Employee Attrition Rate by monthly income groups")
monthly_income_attrition_plot
```

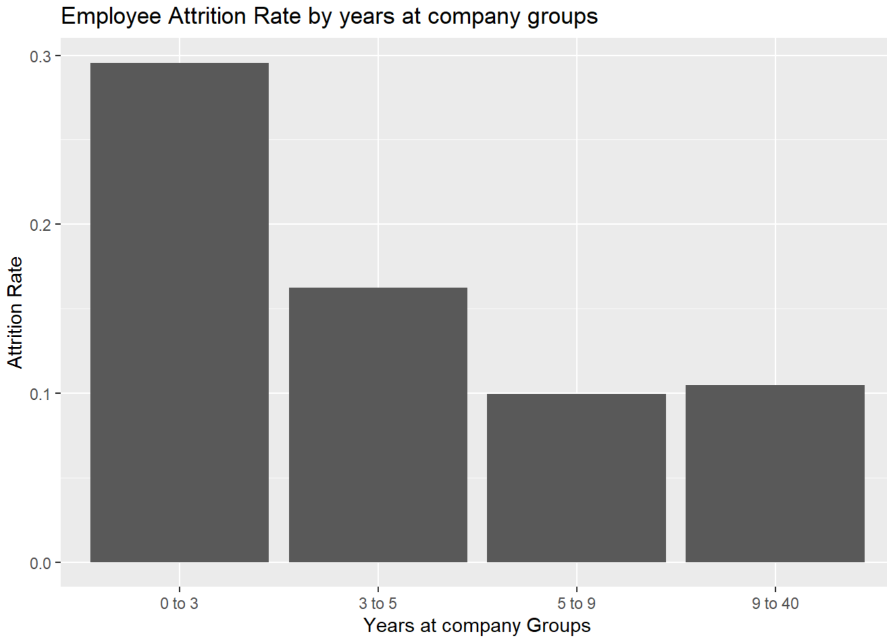



## 6. Attrition Rate is higher amongst employees who have worked for fewer years at the company

```
df_yearsatcompany <- emp_attr_categorical_variables %>%
group_by(YearsAtCompany_groups,Attrition) %>%
summarise(count = n()) %>%
spread(Attrition,count) %>%
ungroup %>%
transmute(YearsAtCompany_groups=YearsAtCompany_groups,attr_rate = Yes/(Yes+No))
df_yearsatcompany
```

```
## # A tibble: 4 x 2
##   YearsAtCompany_groups attr_rate
##   <fct>                     <dbl>
## 1 0 to 3                   0.296
## 2 3 to 5                   0.162
## 3 5 to 9                   0.0994
## 4 9 to 40                  0.105
```

```
colnames(df_yearsatcompany) <- c("YearsAtCompany_groups","Attrition_Rate")
yearsatcompany_plot <- ggplot(df_yearsatcompany,aes(x=YearsAtCompany_groups,y=Attrition_Rate))+
geom_col(show.legend=TRUE,position = "dodge")+
xlab("Years at company Groups")+ ylab("Attrition Rate")+
ggtitle("Employee Attrition Rate by years at company groups")
yearsatcompany_plot
```

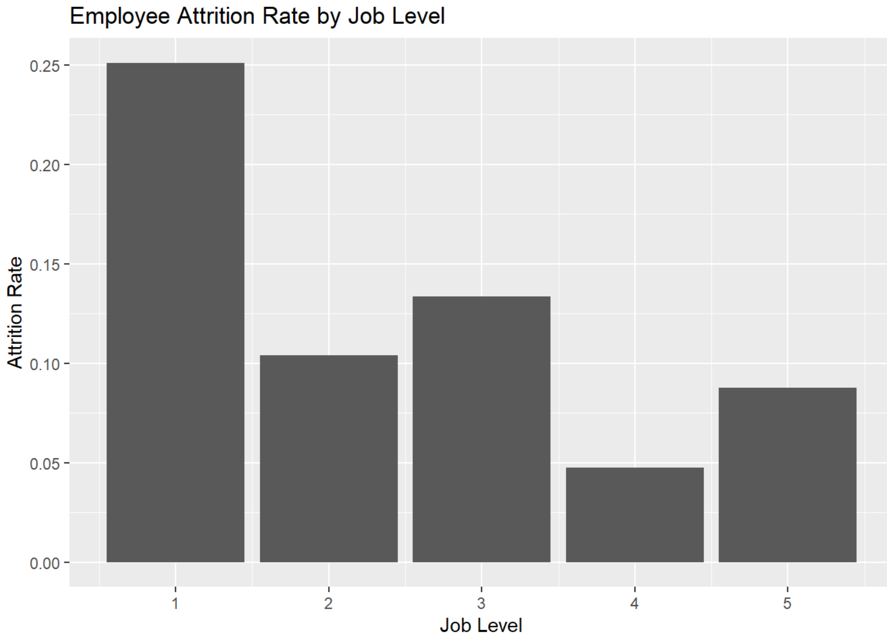



## 7. Attrition Rate is higher amongst employees with lower job levels

```
df_joblevel <- emp_attr_categorical_variables %>%
  group_by(JobLevel,Attrition) %>%
  summarise(count = n()) %>%
  spread(Attrition,count) %>%
  ungroup %>%
  transmute(JobLevel=JobLevel,attr_rate = Yes/(Yes+No))
df_joblevel
```

```
## # A tibble: 5 x 2
##   JobLevel attr_rate
##      <int>     <dbl>
## 1        1    0.251
## 2        2    0.104
## 3        3    0.134
## 4        4    0.0476
## 5        5    0.0877
```

```
colnames(df_joblevel) <- c("JobLevel","Attrition_Rate")
JobLevel_plot <- ggplot(df_joblevel,aes(x=JobLevel,y=Attrition_Rate))+
  geom_col(show.legend=TRUE,position = "dodge")+
  xlab("Job Level")+ ylab("Attrition Rate")+
  ggtitle("Employee Attrition Rate by Job Level")
JobLevel_plot
```

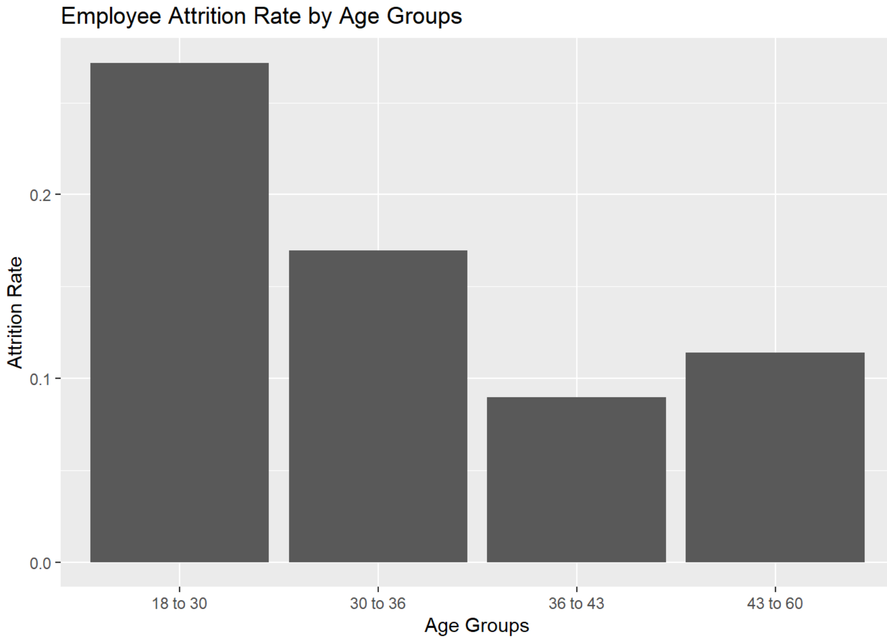



## 8. Attrition Rate is higher amongst lower age group employees

```
df_age_groups <- emp_attr_categorical_variables %>%
  group_by(age_groups,Attrition) %>%
  summarise(count = n()) %>%
  spread(Attrition,count) %>%
  ungroup %>%
  transmute(age_groups=age_groups,attr_rate = Yes/(Yes+No))
df_age_groups
```

```
## # A tibble: 4 x 2
##   age_groups attr_rate
##   <fct>          <dbl>
## 1 18 to 30      0.272
## 2 30 to 36      0.169
## 3 36 to 43      0.0897
## 4 43 to 60      0.114
```

```
colnames(df_age_groups) <- c("AgeGroups","Attrition_Rate")
age_group_plot <- ggplot(df_age_groups,aes(x=AgeGroups,y=Attrition_Rate))+
  geom_col(show.legend=TRUE,position = "dodge")+
  xlab("Age Groups")+ ylab("Attrition Rate")+
  ggtitle("Employee Attrition Rate by Age Groups")
age_group_plot
```




## Removing Continuous numeric variables from dataframe for running Association Rules Mining

```
emp_attr_arm <- emp_attr_categorical_variables
emp_attr_arm$Age<-NULL
emp_attr_arm$DailyRate<-NULL
emp_attr_arm$DistanceFromHome<-NULL
emp_attr_arm$HourlyRate<-NULL
emp_attr_arm$MonthlyIncome<-NULL
emp_attr_arm$MonthlyRate<-NULL
emp_attr_arm$NumCompaniesWorked<-NULL
emp_attr_arm$PercentSalaryHike<-NULL
emp_attr_arm$TotalWorkingYears<-NULL
emp_attr_arm$YearsAtCompany<-NULL
emp_attr_arm$YearsInCurrentRole<-NULL
emp_attr_arm$YearsSinceLastPromotion<-NULL
emp_attr_arm$YearsWithCurrManager<-NULL
emp_attr_arm$Education <-as.factor(emp_attr_arm$Education)
emp_attr_arm$EnvironmentSatisfaction <- as.factor(emp_attr_arm$EnvironmentSatisfaction)
emp_attr_arm$JobInvolvement <- as.factor(emp_attr_arm$JobInvolvement)
emp_attr_arm$JobLevel <- as.factor(emp_attr_arm$JobLevel)
emp_attr_arm$JobSatisfaction <- as.factor(emp_attr_arm$JobSatisfaction)
emp_attr_arm$PerformanceRating <- as.factor(emp_attr_arm$PerformanceRating)
emp_attr_arm$RelationshipSatisfaction <- as.factor(emp_attr_arm$RelationshipSatisfaction)
emp_attr_arm$StockOptionLevel <- as.factor(emp_attr_arm$StockOptionLevel)
emp_attr_arm$TrainingTimesLastYear <- as.factor(emp_attr_arm$TrainingTimesLastYear)
emp_attr_arm$WorkLifeBalance <- as.factor(emp_attr_arm$WorkLifeBalance)
```

# ASSOCIATION RULES MINING, using APRIORI

**CREATING TRANSACTIONS MATRIX**

```
#install.packages("arules")
library(arules)
```

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

```
## 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
```

```
#install.packages("arulesViz")
library(arulesViz)
```

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

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

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

```
emp_attr_armX <- as(emp_attr_arm,"transactions")
```

# BASELINE MODEL

**BASELINE MODEL RULES for ATTRITION “NO”, i.e. employees who have not attritioned**

```
ruleset_attrition_no_baseline <- apriori(emp_attr_armX,
                   appearance = list(default = "lhs", rhs=("Attrition=No")))
```

```
## 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 ...[1 item(s)] done [0.00s].
## set transactions ...[124 item(s), 1176 transaction(s)] done [0.00s].
## sorting and recoding items ... [102 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.06s].
## writing ... [2527 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

**INSPECT RULES for Attrition = ‘No’ Baseline model**

```
inspect(head(sort (ruleset_attrition_no_baseline, by="lift", decreasing=TRUE),5))
```

```
##     lhs                                        rhs              support confidence     lift count
## [1] {Department=Research & Development,
##      JobLevel=2,
##      PerformanceRating=3,
##      WorkLifeBalance=3}                     => {Attrition=No} 0.1003401  0.9752066 1.157258   118
## [2] {StockOptionLevel=1,
##      age_groups=36 to 43}                   => {Attrition=No} 0.1164966  0.9716312 1.153015   137
## [3] {Department=Research & Development,
##      OverTime=No,
##      StockOptionLevel=1,
##      WorkLifeBalance=3}                     => {Attrition=No} 0.1105442  0.9701493 1.151257   130
## [4] {Department=Research & Development,
##      JobLevel=2,
##      MonthlyIncome_groups=4950.5 to 8354.5} => {Attrition=No} 0.1062925  0.9689922 1.149884   125
## [5] {JobInvolvement=3,
##      OverTime=No,
##      TotalWorkingYears_groups=15 to 35.25}  => {Attrition=No} 0.1054422  0.9687500 1.149596   124
```

**BASELINE MODEL RULES for ATTRITION “YES”, i.e. employees who have attritioned**

```
ruleset_attrition_yes_baseline <- apriori(emp_attr_armX,
                   appearance = list(default = "lhs", rhs=("Attrition=Yes")))
```

```
## 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 ...[1 item(s)] done [0.00s].
## set transactions ...[124 item(s), 1176 transaction(s)] done [0.00s].
## sorting and recoding items ... [102 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.02s].
## writing ... [0 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

**INSPECT RULES for Attrition = ‘YES’ Baseline model** *No rules returned for baseline model*

```
inspect(head(sort (ruleset_attrition_yes_baseline, by="lift", decreasing=TRUE),5))
```

# FINE TUNED MODELs by adjusting HYPERPARAMETERS

**RULES for ATTRITION “NO”, i.e. employees who have not attritioned**

```
ruleset_attrition_no <- apriori(emp_attr_armX,
                   parameter = list(support = 0.35, confidence = 0.7, minlen = 3, maxlen = 15),
                   appearance = list(default = "lhs", rhs=("Attrition=No")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.35      3
##  maxlen target   ext
##      15  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 411
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[124 item(s), 1176 transaction(s)] done [0.00s].
## sorting and recoding items ... [19 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [17 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

**RULES for ATTRITION “Yes”, i.e. employees who have attritioned**

```
ruleset_attrition_yes <- apriori(emp_attr_armX,
                                parameter = list(support = 0.03, confidence = 0.6,  minlen = 3, maxlen = 15),
                                appearance = list(default = "lhs", rhs=("Attrition=Yes")))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5    0.03      3
##  maxlen target   ext
##      15  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 35
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[124 item(s), 1176 transaction(s)] done [0.00s].
## sorting and recoding items ... [123 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 done [0.62s].
## writing ... [8 rule(s)] done [0.02s].
## creating S4 object  ... done [0.01s].
```

*By Fine Tuning the Hyperparameters, we can filter out the best rules, based on confidence and lift. Overfitting and Underfitting does not exist for Association Rules Mining, as it is an unsupervised learning algorithm. High Confidence implies, that number of times RHS has occured, given LHS has occured is high. Moreover, a high Lift along with high confidence makes the rule significant and interesting because, it not only tells us that out of all the times, LHS has occured, even RHS has occured, but also that, whenever RHS has occured, LHS has occured with it. i.e. High Confidence and High Lift means that product of support of RHS and support of LHS independently is lower than support of (LHS and RHS) together*

## Using the inspect() command, review the “ruleset” for Attrition = No, i.e. people who stay

*These are the top 5 most interesting and significant rules, because they are having the highest lift. Hence these rules can predict people who stay Since lift is > 1, the LHS & RHS are positively correlated.*

```
inspect(head(sort (ruleset_attrition_no, by="lift", decreasing=TRUE),5))
```

```
##     lhs                                    rhs              support confidence     lift count
## [1] {Department=Research & Development,
##      OverTime=No}                       => {Attrition=No} 0.4226190  0.9119266 1.082165   497
## [2] {JobInvolvement=3,
##      OverTime=No}                       => {Attrition=No} 0.3792517  0.9102041 1.080121   446
## [3] {OverTime=No,
##      WorkLifeBalance=3}                 => {Attrition=No} 0.4056122  0.9051233 1.074092   477
## [4] {BusinessTravel=Travel_Rarely,
##      OverTime=No,
##      PerformanceRating=3}               => {Attrition=No} 0.3903061  0.9035433 1.072217   459
## [5] {BusinessTravel=Travel_Rarely,
##      OverTime=No}                       => {Attrition=No} 0.4600340  0.9031720 1.071776   541
```

**Using the inspect() command, review the "ruleset" for Attrition = Yes, i.e. people who leave** *These are the top 5 most interesting and significant rules, because they are having the highest lift. Hence these rules can predict people who leave Since lift is > 1, the LHS & RHS are positively correlated.*

```
inspect(head(sort (ruleset_attrition_yes, by="lift", decreasing=TRUE),5))
```

```
##     lhs                                       rhs               support confidence     lift count
## [1] {StockOptionLevel=0,
##      age_groups=18 to 30,
##      YearsAtCompany_groups=0 to 3}        => {Attrition=Yes} 0.03061224  0.6315789 4.014794    36
## [2] {JobLevel=1,
##      OverTime=Yes,
##      StockOptionLevel=0}                  => {Attrition=Yes} 0.03146259  0.6271186 3.986441    37
## [3] {MaritalStatus=Single,
##      TotalWorkingYears_groups=0 to 6,
##      YearsAtCompany_groups=0 to 3}        => {Attrition=Yes} 0.03061224  0.6206897 3.945573    36
## [4] {MaritalStatus=Single,
##      StockOptionLevel=0,
##      TotalWorkingYears_groups=0 to 6,
##      YearsAtCompany_groups=0 to 3}        => {Attrition=Yes} 0.03061224  0.6206897 3.945573    36
## [5] {JobLevel=1,
##      MaritalStatus=Single,
##      MonthlyIncome_groups=1009 to 2954.5,
##      TotalWorkingYears_groups=0 to 6}     => {Attrition=Yes} 0.03061224  0.6206897 3.945573    36
```

## Experiment with the interactive ruleset interface by running the inspectDT() command

**For Attrition No**

```
ruleset_attrition_no_sort <- sort (ruleset_attrition_no, by="lift", decreasing=TRUE)
inspectDT(ruleset_attrition_no_sort)
```

**Show** 10 ▼ **entries** **Search:**

| | LHS | RHS | support | confidence | lift | count |
|---|---|---|---|---|---|---|
| | All | A | | All | | |
| [1] | {Department=Research & Development,OverTime=No} | {Attrition=No} | 0.423 | 0.912 | 1.082 | 497.000 |
| [2] | {JobInvolvement=3,OverTime=No} | {Attrition=No} | 0.379 | 0.910 | 1.080 | 446.000 |
| [3] | {OverTime=No,WorkLifeBalance=3} | {Attrition=No} | 0.406 | 0.905 | 1.074 | 477.000 |
| [4] | {BusinessTravel=Travel_Rarely,OverTime=No,PerformanceRating=3} | {Attrition=No} | 0.390 | 0.904 | 1.072 | 459.000 |
| [5] | {BusinessTravel=Travel_Rarely,OverTime=No} | {Attrition=No} | 0.460 | 0.903 | 1.072 | 541.000 |
| [6] | {Department=Research & Development,WorkLifeBalance=3} | {Attrition=No} | 0.361 | 0.895 | 1.062 | 424.000 |
| [7] | {OverTime=No,PerformanceRating=3} | {Attrition=No} | 0.531 | 0.890 | 1.057 | 625.000 |
| [8] | {Gender=Male,OverTime=No} | {Attrition=No} | 0.381 | 0.889 | 1.055 | 448.000 |
| [9] | {Department=Research & Development,PerformanceRating=3} | {Attrition=No} | 0.477 | 0.885 | 1.050 | 561.000 |
| [10] | {BusinessTravel=Travel_Rarely,Department=Research & Development,PerformanceRating=3} | {Attrition=No} | 0.350 | 0.884 | 1.049 | 412.000 |

Showing 1 to 10 of 17 entries Previous 1 2 Next

**For Attrition Yes**

```
ruleset_attrition_yes_sort <- sort (ruleset_attrition_yes, by="lift", decreasing=TRUE)
inspectDT(ruleset_attrition_yes_sort)
```

**Show** 10 ▼ **entries** **Search:**

| | LHS | RHS | support | confidence | lift |
|---|---|---|---|---|---|
| | All | Al | | All | |
| [1] | {StockOptionLevel=0,age_groups=18 to 30,YearsAtCompany_groups=0 to 3} | {Attrition=Yes} | 0.031 | 0.632 | 4.015 |

| | LHS | RHS | support | confidence | lift |
|---|---|---|---|---|---|
| | All | Al | | All | |
| [2] | {JobLevel=1,OverTime=Yes,StockOptionLevel=0} | {Attrition=Yes} | 0.031 | 0.627 | 3.986 |
| [3] | {MaritalStatus=Single,TotalWorkingYears_groups=0 to 6,YearsAtCompany_groups=0 to 3} | {Attrition=Yes} | 0.031 | 0.621 | 3.946 |
| [4] | {MaritalStatus=Single,StockOptionLevel=0,TotalWorkingYears_groups=0 to 6,YearsAtCompany_groups=0 to 3} | {Attrition=Yes} | 0.031 | 0.621 | 3.946 |
| [5] | {JobLevel=1,MaritalStatus=Single,MonthlyIncome_groups=1009 to 2954.5,TotalWorkingYears_groups=0 to 6} | {Attrition=Yes} | 0.031 | 0.621 | 3.946 |
| [6] | {JobLevel=1,MaritalStatus=Single,StockOptionLevel=0,MonthlyIncome_groups=1009 to 2954.5,TotalWorkingYears_groups=0 to 6} | {Attrition=Yes} | 0.031 | 0.621 | 3.946 |
| [7] | {MaritalStatus=Single,MonthlyIncome_groups=1009 to 2954.5,TotalWorkingYears_groups=0 to 6} | {Attrition=Yes} | 0.031 | 0.610 | 3.879 |
| [8] | {MaritalStatus=Single,StockOptionLevel=0,MonthlyIncome_groups=1009 to 2954.5,TotalWorkingYears_groups=0 to 6} | {Attrition=Yes} | 0.031 | 0.610 | 3.879 |

Showing 1 to 8 of 8 entries

Previous 1 Next