Human Resources Project

Jihene Samet

09/02/2021

Data description

Data title: Human Resources Dataset

https://www.kaggle.com/rhuebner/human-resources-data-set

Codebook

https://rpubs.com/rhuebner/hr_codebook_v14

Context

This dataset was updated 10/19/2020. It was created by Dr. Carla Patalano and Dr. Rich Huebner , which is used in one of their graduate MSHRM courses called HR Metrics and Analytics, at New England College of Business. They use the data set to teach HR students how to use and analyze the data in Tableau Desktop.

The data provide information about employees and their characteristics.

What made me choose this data set is my new interest in Human Resources work and its effect on the contribution to the success of a Company or an organization.

In this project, I'm going to practice what I've learned in Data Analytics class. So let's get started.

Importing packages

```
library(magrittr)
library(dplyr)
library(tidyr) # to gather columns into key-value pairs
library(ggplot2) #for data viz
library(arules) # for association rules
library(arulesViz) #for viz association rules
library(grid) # for data viz
library(gridExtra) #for data viz
library(moments) # for skewness
library(nortest) # for pearson test
```

Importing data set

Once I've downloaded and installed my data set , I can import it into R to start doing some preliminary analysis and get a sense of what the data looks like.

```
HRdata <- read.csv("HRdataset.csv", sep=",", header = TRUE)
```

Exploring and describing

The following steps will lead you through the initial exploration of our data set, where we understand more the features and look for missing data.

```
str(HRdata)
```

```
## 'data.frame':    311 obs. of  36 variables:
##  $ ï..Employee_Name           : chr  "Adinolfi, Wilson  K" "Ait Sidi, Karthikeyan     " "Akinkuoli
e, Sarah" "Alagbe,Trina" ...
##  $ EmpID                      : int  10026 10084 10196 10088 10069 10002 10194 10062 10114 10250
...
##  $ MarriedID                  : int  0 1 1 1 0 0 0 0 0 0 ...
##  $ MaritalStatusID            : int  0 1 1 1 2 0 0 4 0 2 ...
##  $ GenderID                   : int  1 1 0 0 0 0 0 1 0 1 ...
##  $ EmpStatusID                : int  1 5 5 1 5 1 1 1 3 1 ...
##  $ DeptID                     : int  5 3 5 5 5 5 4 5 5 3 ...
##  $ PerfScoreID                : int  4 3 3 3 3 4 3 3 3 3 ...
##  $ FromDiversityJobFairID     : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ Salary                     : int  62506 104437 64955 64991 50825 57568 95660 59365 47837 5017
8 ...
##  $ Termd                      : int  0 1 1 0 1 0 0 0 0 0 ...
##  $ PositionID                 : int  19 27 20 19 19 19 24 19 19 14 ...
##  $ Position                   : chr  "Production Technician I" "Sr. DBA" "Production Technician
II" "Production Technician I" ...
##  $ State                      : chr  "MA" "MA" "MA" "MA" ...
##  $ Zip                        : int  1960 2148 1810 1886 2169 1844 2110 2199 1902 1886 ...
##  $ DOB                        : chr  "07/10/83" "05/05/75" "09/19/88" "09/27/88" ...
##  $ Sex                        : chr  "M " "M " "F" "F" ...
##  $ MaritalDesc                : chr  "Single" "Married" "Married" "Married" ...
##  $ CitizenDesc                : chr  "US Citizen" "US Citizen" "US Citizen" "US Citizen" ...
##  $ HispanicLatino             : chr  "No" "No" "No" "No" ...
##  $ RaceDesc                   : chr  "White" "White" "White" "White" ...
##  $ DateofHire                 : chr  "7/5/2011" "3/30/2015" "7/5/2011" "1/7/2008" ...
##  $ DateofTermination          : chr  "" "6/16/2016" "9/24/2012" "" ...
##  $ TermReason                 : chr  "N/A-StillEmployed" "career change" "hours" "N/A-StillEmplo
yed" ...
##  $ EmploymentStatus           : chr  "Active" "Voluntarily Terminated" "Voluntarily Terminated"
"Active" ...
##  $ Department                 : chr  "Production       " "IT/IS" "Production       " "Production
" ...
##  $ ManagerName                : chr  "Michael Albert" "Simon Roup" "Kissy Sullivan" "Elijiah Gra
y" ...
##  $ ManagerID                  : int  22 4 20 16 39 11 10 19 12 7 ...
##  $ RecruitmentSource          : chr  "LinkedIn" "Indeed" "LinkedIn" "Indeed" ...
##  $ PerformanceScore           : chr  "Exceeds" "Fully Meets" "Fully Meets" "Fully Meets" ...
##  $ EngagementSurvey           : num  4.6 4.96 3.02 4.84 5 5 3.04 5 4.46 5 ...
##  $ EmpSatisfaction            : int  5 3 3 5 4 5 3 4 3 5 ...
##  $ SpecialProjectsCount       : int  0 6 0 0 0 0 4 0 0 6 ...
##  $  LastPerformanceReview_Date:  chr  "1/17/2019" "2/24/2016" "5/15/2012" "1/3/2019" ...
##  $ DaysLateLast30             : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Absences                   : int  1 17 3 15 2 15 19 19 4 16 ...
```

Our data contains 311 observations(rows) and 36 variables(columns).It's has multiple types. We have numeric , binary and categorical variables. Here is a description of some variable:

Feature: DeptID, Description:Department ID code that matches the department the employee works in, DataType: Integer/

Feature: Termd, Description: Has this employee been terminated 1 or 0, DataType: Binary/

Feature: EngagementSurvey, Description: Results from the last engagement survey, DataType: numeric /

Feature: EmploymentStatus, Description: A description of the person's employment status. Anyone currently working full time = Active, DataType: character.

PerfScoreID, EngagementSurvey and EmpSatisfaction are score between 1 and 5.

Looking for NAs

```
sum(is.na(HRdata))
```

```
## [1] 8
```

There are 8 missing values. Let's see where they are.

```
colSums(is.na(HRdata))
```

```
##             ï..Employee_Name                      EmpID
##                            0                          0
##                    MarriedID             MaritalStatusID
##                            0                          0
##                     GenderID                 EmpStatusID
##                            0                          0
##                       DeptID                PerfScoreID
##                            0                          0
##       FromDiversityJobFairID                     Salary
##                            0                          0
##                        Termd                  PositionID
##                            0                          0
##                     Position                      State
##                            0                          0
##                          Zip                        DOB
##                            0                          0
##                          Sex                 MaritalDesc
##                            0                          0
##                 CitizenDesc              HispanicLatino
##                            0                          0
##                     RaceDesc                  DateofHire
##                            0                          0
##            DateofTermination                  TermReason
##                            0                          0
##             EmploymentStatus                  Department
##                            0                          0
##                  ManagerName                   ManagerID
##                            0                          8
##            RecruitmentSource            PerformanceScore
##                            0                          0
##             EngagementSurvey             EmpSatisfaction
##                            0                          0
##         SpecialProjectsCount  LastPerformanceReview_Date
##                            0                          0
##               DaysLateLast30                    Absences
##                            0                          0
```

These missing values needs to be taken to account and needs to be removed using na.omit() function, to prevent us from getting NA values when running calculations which it can affect our results.

```
HRdata = na.omit(HRdata)
sum(is.na(HRdata))
```

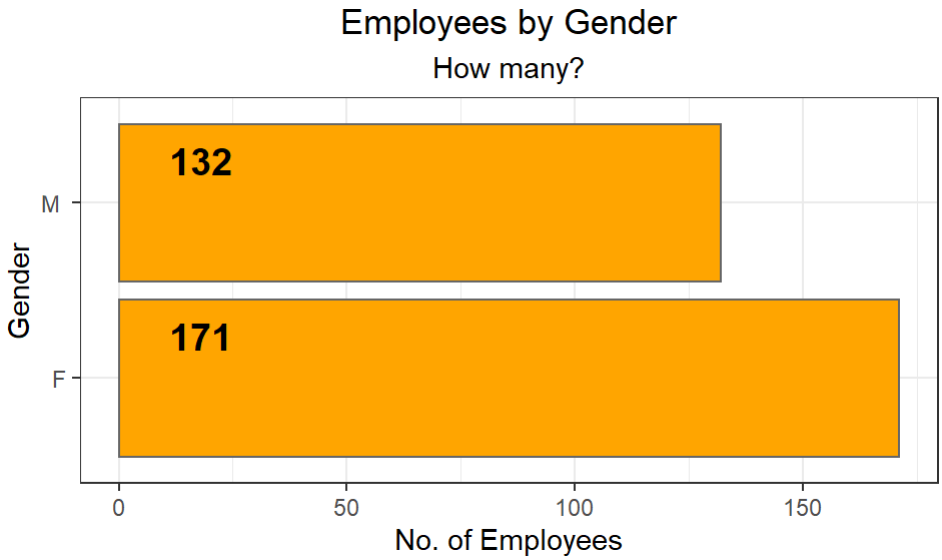
```
## [1] 0
```

The data now contains 308 observations(rows) and 36 variables(columns). No more missing data. Our data now is all good to work with.

Data visualization

Bar charts of employees by gender

```
options(repr.plot.width = 8, repr.plot.height = 4)
employeesnumbers <- HRdata %>%
group_by(Sex) %>%
summarise(count = n()) %>%
ggplot(aes(x=Sex, y=count)) +
geom_bar(stat="identity", fill = "orange", color = "grey40") +
theme_bw() +
coord_flip() +
geom_text(aes(x = Sex, y = 0.01, label = count), hjust = -0.8
          , vjust = -1, size = 5, color = "black", fontface = "bold", angle = 360) +
labs(title = "Employees by Gender", x = "Gender"
     , y = "No. of Employees", subtitle = "How many?") +
theme(plot.title=element_text(hjust=0.5), plot.subtitle=element_text(hjust=0.5))
employeesnumbers
```



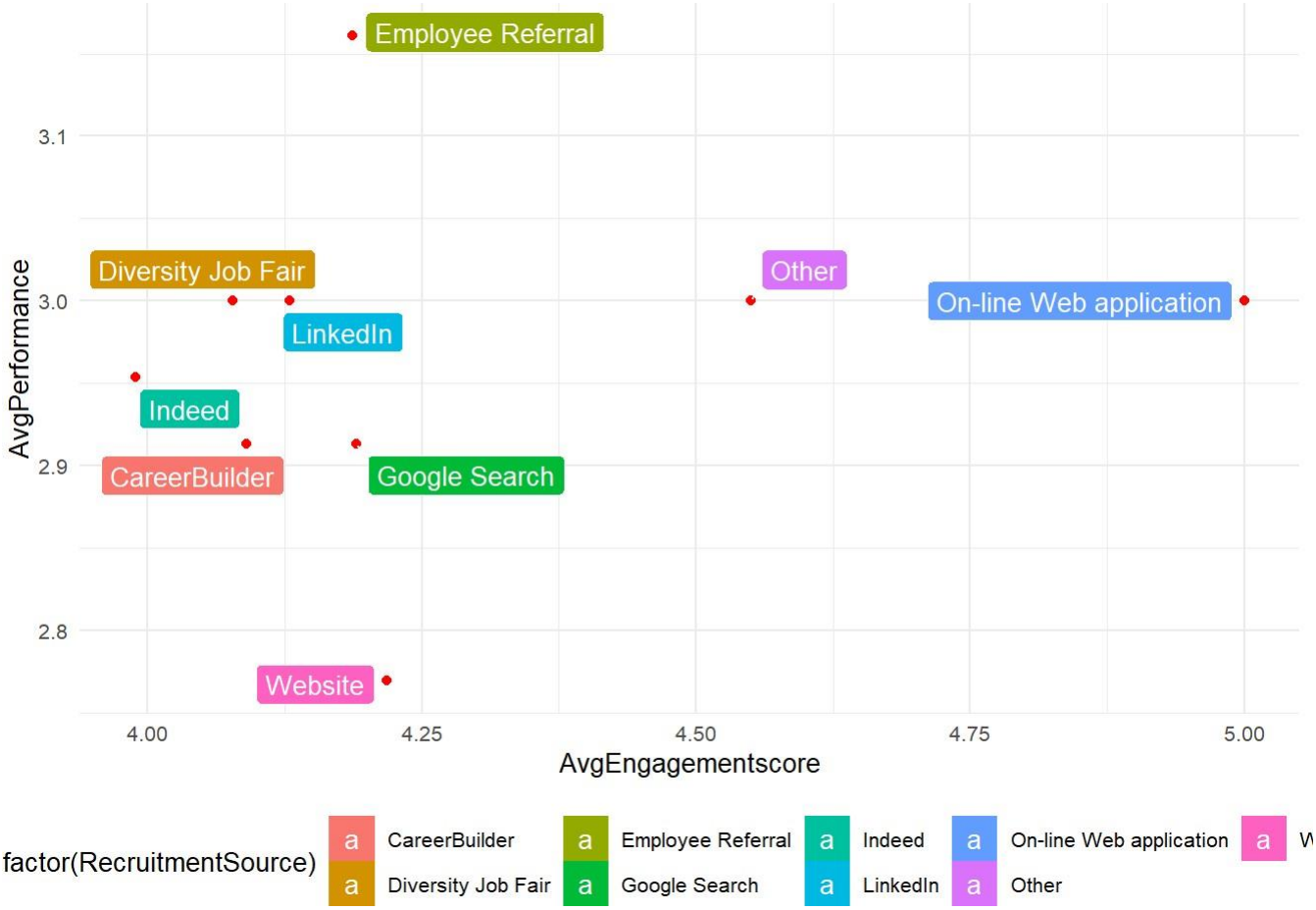
As you can see there is 57% of females work in the company and 43% males.

Identifying the best recruiting source

We will use three metrics to determine the best recruitment source. We use geom_point() to visualize it so that we can easily see the average performance and engagement for each recruitment source.

```
library(ggrepel)
HRdata %>%
```

```
  group_by(RecruitmentSource)  %>%
  summarise(AvgPerformance = mean(PerfScoreID), AvgEngagementscore = mean(EngagementSurvey)) %>%
  arrange(AvgPerformance,AvgEngagementscore) %>%
  ggplot(., aes(x = AvgEngagementscore, y = AvgPerformance))+
geom_point(color = 'red') +
theme_minimal(base_size = 10)+ geom_label_repel(aes(label = RecruitmentSource,
fill = factor(RecruitmentSource)), color = 'white',
size = 3.5) +
theme(legend.position = "bottom")
```



Now is the time for us to draw conclusions from the data analysis we have done. From the data visualization we created, we can see a clear picture of the best recruiting sources to ensure a diverse organization are Online web application , Employee referral and Other. They have the highest score of engagement and performance.

Department analysis

As HR analysis I want to see the performance of every department and help to improve it.

First I'll look for which department have the lowest engagement score, which is a score less than 2. Then I'm going to group department and gender(I want to see the performance of each gender in the department) by percentage of disengagement , salary and absences.

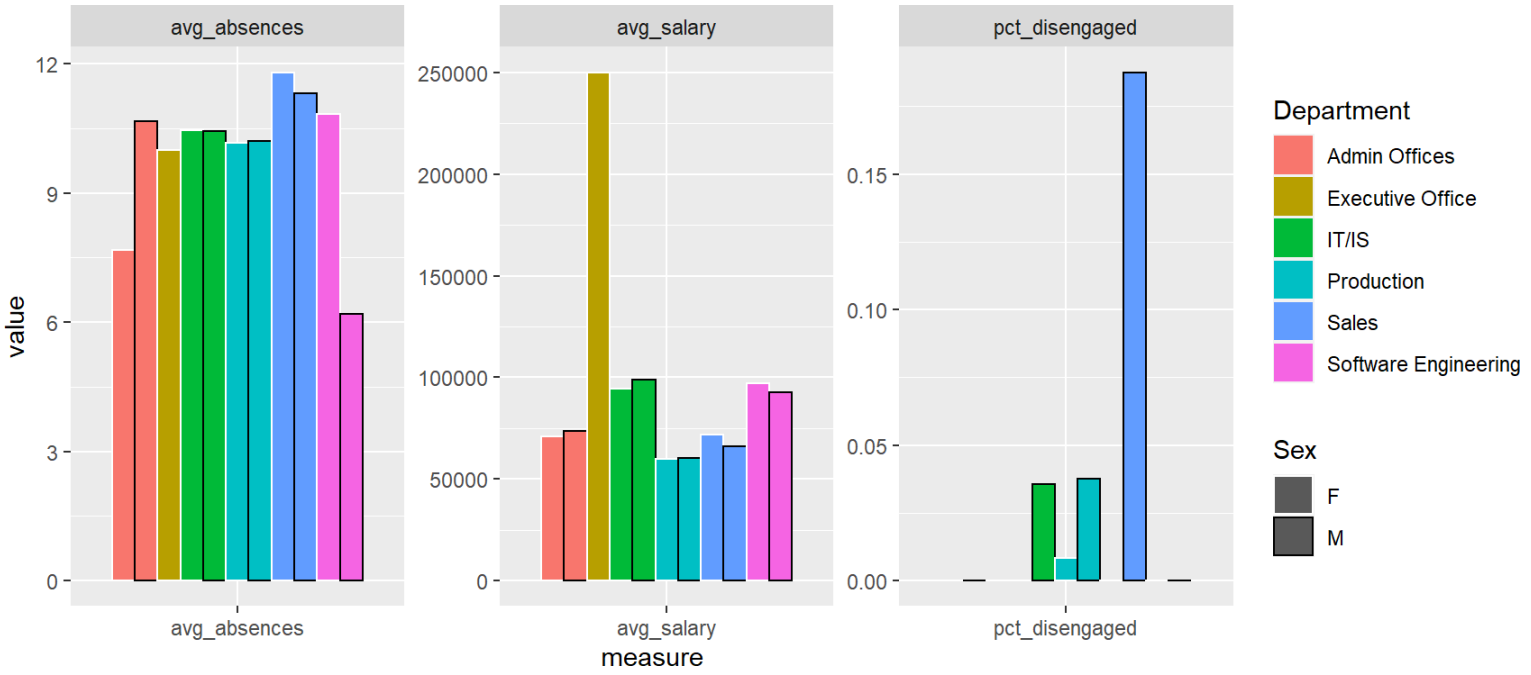
```
# first prepare  Least Engaged column
disengaged <- HRdata %>%
  mutate(LeastEngaged = ifelse(EngagementSurvey <= 2, 1, 0))

Dep_summary <- disengaged %>%
  group_by(Department,Sex) %>%
  summarize(pct_disengaged = mean(LeastEngaged),
```

```
            avg_salary = mean(Salary),
            avg_absences = mean(Absences))
```

```
## `summarise()` has grouped output by 'Department'. You can override using the `.groups` argumen
t.
```

```
Dep_gathered <- Dep_summary %>%
  gather(key = "measure", value = "value",
         pct_disengaged, avg_salary, avg_absences)
ggplot(Dep_gathered, aes(measure, value, fill = Department, color = Sex)) +
  geom_col(position = "dodge") +
  facet_wrap(~ measure, scales = "free") +
 scale_color_manual(values=c("white","black"))
```



The bars framed in white represent Female and the one framed in black represent Male.

The bars are filled with departments. Each color represent a department.

This is interesting. It turnout that Executive office have only Female executives.

In the first plot we see that females are often more absent than males especially in Software Eng department and the average absences in Sales department is little higher then the others.

The second plot show us that Executive office department has the highest salary, which is normal and IT/IS is second highest, and Sales and Productions departments have the lowest salary. We notice that in 3 out of 5 departments, males have high salary than females: imbalance of payment between gender(we'll test this later).

In the third plot we notice that Sales department is the most disengaged. Going back to plot one and two, it has the most absences and the second worst average salary. Therefore Sales department needs to be taking under consideration and find a way to cheer team spirit up. Also we see that Males are the ones who show disengagement. Maybe the Executive office having only females executives may have an effect on this.

Association Rules Minig

I'm applying association rule to find the relative between the features of my data set. For HR analysis one of the main tasks is to find the degree of satisfaction of the employees. This method will help us see what makes an employee satisfied and other

dissatisfied.

Converting numeric variables into categorical

I'm going to do some changes to my data to prepare it for association rules mining. First I'm using discretize() function from {arules} package to convert a numeric variable into a categorical variable.

Converting salary, absences and employees satisafation by discretization

So here I'm going to prepare a new data set by converting Engagement Survey, Salary, Absences and Employees Satisfaction from numeric to categorical .

```
HR.new <- discretizeDF(HRdata, methods = list(
  EngagementSurvey = list(method = "frequency", breaks = 3,
     labels = c( "least engaged", "engaged", "most engaged")),

   Salary = list(method = "frequency", breaks = 4,
     labels = c("below average", "average", "good", "high")),
   Absences = list(method = "frequency", breaks = 3,
     labels = c("perfect", "good","concerned" )),
   EmpSatisfaction = list(method = "frequency", breaks= 2,
     labels = c( "dissatisfied" , "satisfied" ))

  ),
  default = list(method = "none")
  )
```

Converting Employee position

First let's take a look at employee's position.

```
HRdata %>%
  count(Position)
```

```
##                          Position   n
## 1                   Accountant I   3
## 2       Administrative Assistant   3
## 3             Area Sales Manager  27
## 4                   BI Developer   4
## 5                    BI Director   1
## 6                            CIO   1
## 7                   Data Analyst   7
## 8                   Data Analyst    1
## 9                  Data Architect   1
## 10        Database Administrator   5
## 11        Director of Operations   1
## 12             Director of Sales   1
## 13          Enterprise Architect   1
## 14                   IT Director   1
## 15              IT Manager - DB   2
## 16            IT Manager - Infra   1
## 17          IT Manager - Support   1
## 18                   IT Support   8
## 19              Network Engineer   5
```

```
## 20               President & CEO   1
## 21      Principal Data Architect   1
## 22            Production Manager  14
## 23       Production Technician I 133
## 24      Production Technician II  53
## 25                 Sales Manager   3
## 26           Senior BI Developer   3
## 27       Shared Services Manager   1
## 28             Software Engineer  10
## 29  Software Engineering Manager   1
## 30                Sr. Accountant   2
## 31                       Sr. DBA   2
## 32          Sr. Network Engineer   5
```

As you can see there are a lot of categories. So it would be interesting to group them by level of responsibility: personnel, manager, director.

```
Personnel <- c('Accountant I', 'Administrative Assistant','BI Developer', 'Data Analyst', 'Data An
alyst ', 'Data Architect', 'Database Administrator', 'Enterprise Architect', 'IT Support', 'Networ
k Engineer', 'Principal Data Architect', 'Production Technician I', 'Production Technician II', 'S
enior BI Developer', 'Software Engineer', 'Sr. Accountant', 'Sr. DBA', 'Sr. Network Engineer')
managers <- c('Area Sales Manager', 'IT Manager - DB', 'IT Manager - Infra', 'IT Manager - Suppor
t', 'Production Manager', 'Sales Manager', 'Shared Services Manager', 'Software Engineering Manage
r')
directors.CEOs <- c('BI Director', 'CIO', 'Director of Operations', 'Director of Sales', 'IT Direc
tor')

HR.new$Position<- factor(HR.new$Position, levels = c(Personnel, managers, directors.CEOs ), labels
= c(rep("Personnel", 18), rep("Manager", 8), rep("Director or CEO ", 5)))

levels(HR.new$Position)
```

```
## [1] "Personnel"         "Manager"           "Director or CEO "
```

Done.

```
tapply(HRdata$Absences,HR.new$Absences,summary)
```

```
## $perfect
##    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   1.000    2.000    3.000    3.414    5.000    6.000
##
## $good
##    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##   7.000    8.000   10.000    9.828   12.000   13.000
##
## $concerned
##    Min. 1st Qu.   Median     Mean  3rd Qu.     Max.
##   14.00   15.00    16.00    16.75    19.00    20.00
```

For better explanation, an employee with less than 6 absences he's a perfect employee, an employee with a absences between 7 and 13 he's a good employee and and employee with absences more than 13 he's a concerned employee.

```
tapply(HRdata$EmpSatisfaction,HR.new$EmpSatisfaction,summary)
```

```
## $dissatisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   2.888   3.000   3.000
##
## $satisfied
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.000   4.000   5.000   4.508   5.000   5.000
```

A rate between 1 and 3 considered as dissatisfied and rate between 4 and 5 considered as satisfied.

Salary is a contentious variable, so I've breaking it into a 4 levels categorical variable by order. let's take a look at salary to understand how the division is made.

```
median(HRdata$Salary)
```

```
## [1] 62910
```

```
tapply(HRdata$Salary,HR.new$Salary,summary)
```

```
## $`below average`
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   45046   47671   50935   50542   53029   55578
##
## $average
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   55688   57907   59369   59466   61352   62810
##
## $good
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   62910   63746   65151   66203   68238   72202
##
## $high
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   72460   81316   92659  100829  105691  250000
```

So here the minimum salary is 45046 and the highest Salary is 250000. What to consider below average salary is a wage between 45046 and 55578, an average salary is wage between 55688 and 62810 , a good salary between 62910 and 72202 and a high salary is between 72460 and 250000.

Presenting the new data set

I made all the changes that I need. Now my data is ready. let's take a look at it.

```
head(HR.new)
```

```
##            ï..Employee_Name EmpID  MarriedID  MaritalStatusID  GenderID  EmpStatusID
## 1      Adinolfi, Wilson  K 10026          0                0         1            1
## 2 Ait Sidi, Karthikeyan     10084          1                1         1            5
## 3         Akinkuolie, Sarah 10196          1                1         0            5
## 4              Alagbe,Trina 10088          1                1         0            1
## 5          Anderson, Carol  10069          0                2         0            5
```

```
## 6        Anderson, Linda   10002          0               0        0           1
##   DeptID  PerfScoreID FromDiversityJobFairID          Salary Termd  PositionID
## 1      5           4                       0         average     0          19
## 2      3           3                       0            high     1          27
## 3      5           3                       0            good     1          20
## 4      5           3                       0            good     0          19
## 5      5           3                       0   below average     1          19
## 6      5           4                       0         average     0          19
##     Position State   Zip       DOB  Sex MaritalDesc  CitizenDesc  HispanicLatino
## 1  Personnel    MA  1960  07/10/83   M       Single   US Citizen              No
## 2  Personnel    MA  2148  05/05/75   M      Married   US Citizen              No
## 3  Personnel    MA  1810  09/19/88    F     Married   US Citizen              No
## 4  Personnel    MA  1886  09/27/88    F     Married   US Citizen              No
## 5  Personnel    MA  2169  09/08/89    F    Divorced   US Citizen              No
## 6  Personnel    MA  1844  05/22/77    F      Single   US Citizen              No
##   RaceDesc  DateofHire  DateofTermination         TermReason
## 1    White   7/5/2011                     N/A-StillEmployed
## 2    White  3/30/2015          6/16/2016      career change
## 3    White   7/5/2011          9/24/2012              hours
## 4    White   1/7/2008                     N/A-StillEmployed
## 5    White  7/11/2011           9/6/2016   return to school
## 6    White   1/9/2012                     N/A-StillEmployed
##          EmploymentStatus         Department     ManagerName  ManagerID
## 1                  Active Production          Michael Albert         22
## 2 Voluntarily Terminated              IT/IS      Simon Roup          4
## 3 Voluntarily Terminated Production          Kissy Sullivan         20
## 4                  Active Production            Elijiah Gray         16
## 5 Voluntarily Terminated Production          Webster Butler         39
## 6                  Active Production                Amy Dunn         11
##   RecruitmentSource  PerformanceScore  EngagementSurvey  EmpSatisfaction
## 1          LinkedIn           Exceeds      most engaged        satisfied
## 2            Indeed       Fully Meets      most engaged     dissatisfied
## 3          LinkedIn       Fully Meets     least engaged     dissatisfied
## 4            Indeed       Fully Meets      most engaged        satisfied
## 5     Google Search       Fully Meets      most engaged        satisfied
## 6          LinkedIn           Exceeds      most engaged        satisfied
##    SpecialProjectsCount  LastPerformanceReview_Date  DaysLateLast30   Absences
## 1                     0                   1/17/2019               0    perfect
## 2                     6                   2/24/2016               0  concerned
## 3                     0                   5/15/2012               0    perfect
## 4                     0                    1/3/2019               0  concerned
## 5                     0                    2/1/2016               0    perfect
## 6                     0                    1/7/2019               0  concerned
```

Starting association rules

First, I will get rid of the unnecessary variables. We want to find the relative between some variables and satisfaction. These variables are salary,Position, Department, performance score , results from the last engagement survey,Employee Satisfaction and Absences. The data set that I will be using for Association rule is as follow:

```
datarule <- HR.new[c(10,13,26,30,31,32,36)]
head(datarule)
```

```
##          Salary  Position        Department PerformanceScore EngagementSurvey
## 1       average Personnel Production                  Exceeds      most engaged
## 2          high Personnel             IT/IS      Fully Meets      most engaged
## 3          good Personnel Production             Fully Meets     least engaged
## 4          good Personnel Production             Fully Meets      most engaged
## 5 below average Personnel Production             Fully Meets      most engaged
## 6       average Personnel Production                  Exceeds      most engaged
##   EmpSatisfaction  Absences
## 1       satisfied   perfect
## 2    dissatisfied concerned
## 3    dissatisfied   perfect
## 4       satisfied concerned
## 5       satisfied   perfect
## 6       satisfied concerned
```

```
str(datarule)
```

```
## 'data.frame':    303 obs. of  7 variables:
##  $ Salary          : Factor w/ 4 levels "below average",..: 2 4 3 3 1 2 4 2 1 1 ...
##   ..- attr(*, "discretized:breaks")= num [1:5] 45046 55633 62910 72331 250000
##   ..- attr(*, "discretized:method")= chr "frequency"
##  $ Position        : Factor w/ 3 levels "Personnel","Manager",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Department      : chr  "Production       " "IT/IS" "Production       " "Production       "
...
##  $ PerformanceScore: chr  "Exceeds" "Fully Meets" "Fully Meets" "Fully Meets" ...
##  $ EngagementSurvey: Factor w/ 3 levels "least engaged",..: 3 3 1 3 3 3 1 3 2 3 ...
##   ..- attr(*, "discretized:breaks")= num [1:4] 1.12 3.98 4.5  5
##   ..- attr(*, "discretized:method")= chr "frequency"
##  $ EmpSatisfaction : Factor w/ 2 levels "dissatisfied",..: 2  1 1 2 2 2 1 2 1 2 ...
##   ..- attr(*, "discretized:breaks")= num [1:3] 1  4  5
##   ..- attr(*, "discretized:method")= chr "frequency"
##  $ Absences        : Factor w/ 3 levels "perfect","good",..:  1 3 1 3 1 3 3 3 1 3 ...
##   ..- attr(*, "discretized:breaks")= num [1:4] 1  7  14  20
##   ..- attr(*, "discretized:method")= chr "frequency"
```

Converting my variables to Factor

I have 2 variables that are characters. I need to convert them to Factor in order to apply association rules.

```
datarule$Department <- as.factor(datarule$Department)
datarule$PerformanceScore <- as.factor(datarule$PerformanceScore)
str(datarule)
```

```
## 'data.frame':    303 obs. of  7 variables:
##  $ Salary          : Factor w/ 4 levels "below average",..: 2 4 3 3 1 2 4 2 1 1 ...
##   ..- attr(*, "discretized:breaks")= num [1:5] 45046 55633 62910 72331 250000
##   ..- attr(*, "discretized:method")= chr "frequency"
##  $ Position        : Factor w/ 3 levels "Personnel","Manager",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Department      : Factor w/ 6 levels "Admin Offices",..: 4 3 4 4 4 4 6 4 4 3 ...
##  $ PerformanceScore: Factor w/ 4 levels "Exceeds","Fully Meets",..: 1 2 2 2 2 1 2 2 2 2 ...
##  $ EngagementSurvey: Factor w/ 3 levels "least engaged",..: 3 3 1 3 3 3 1 3 2 3 ...
##   ..- attr(*, "discretized:breaks")= num [1:4] 1.12 3.98 4.5 5
##   ..- attr(*, "discretized:method")= chr "frequency"
##  $ EmpSatisfaction : Factor w/ 2 levels "dissatisfied",..: 2 1 1 2 2 2 1 2 1 2 ...
```

```
##   ..- attr(*, "discretized:breaks")= num [1:3] 1 4 5
##   ..- attr(*, "discretized:method")= chr "frequency"
##  $ Absences         : Factor w/ 3 levels "perfect","good",..: 1 3 1 3 1 3 3 3 1 3 ...
##   ..- attr(*, "discretized:breaks")= num [1:4] 1 7 14 20
##   ..- attr(*, "discretized:method")= chr "frequency"
```

My variables are all Factor. So I'm good to go.

Converting data from dataframe to transactions

```
## transactions as itemMatrix in sparse format with
##  303 rows (elements/itemsets/transactions) and
##  25 columns (items) and a density of 0.279868
##
## most frequent items:
##           Position=Personnel PerformanceScore=Fully Meets
##                          247                          236
## Department=Production            EmpSatisfaction=satisfied
##                          201                          187
## EmpSatisfaction=dissatisfied                        (Other)
##                          116                         1133
##
## element (itemset/transaction) length distribution:
## sizes
##   6   7
##   1 302
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   6.000   7.000   7.000   6.997   7.000   7.000
##
## includes extended item information - examples:
##                 labels  variables        levels
## 1 Salary=below average    Salary below average
## 2       Salary=average     Salary       average
## 3          Salary=good    Salary          good
##
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

The matrix has a density of 27.98%, which represents the proportion of non-zero cells. Our sparse matrix' summary gives information about the transaction's sizes. one transaction have size of 6 items and 302 transactions have a size of 7 items. All employees satisfy 7 features expect 1 that satisfy only 6 features.

The summary lists the most frequent items found in the matrix.

PerformanceScore=Fully Meets, 236 times, which means that , we can determine that performanceScore Fully Meets appeared in 76% of transactions.

Position=Personnel, 247 times, which means that 77.88% of employees are stuff.

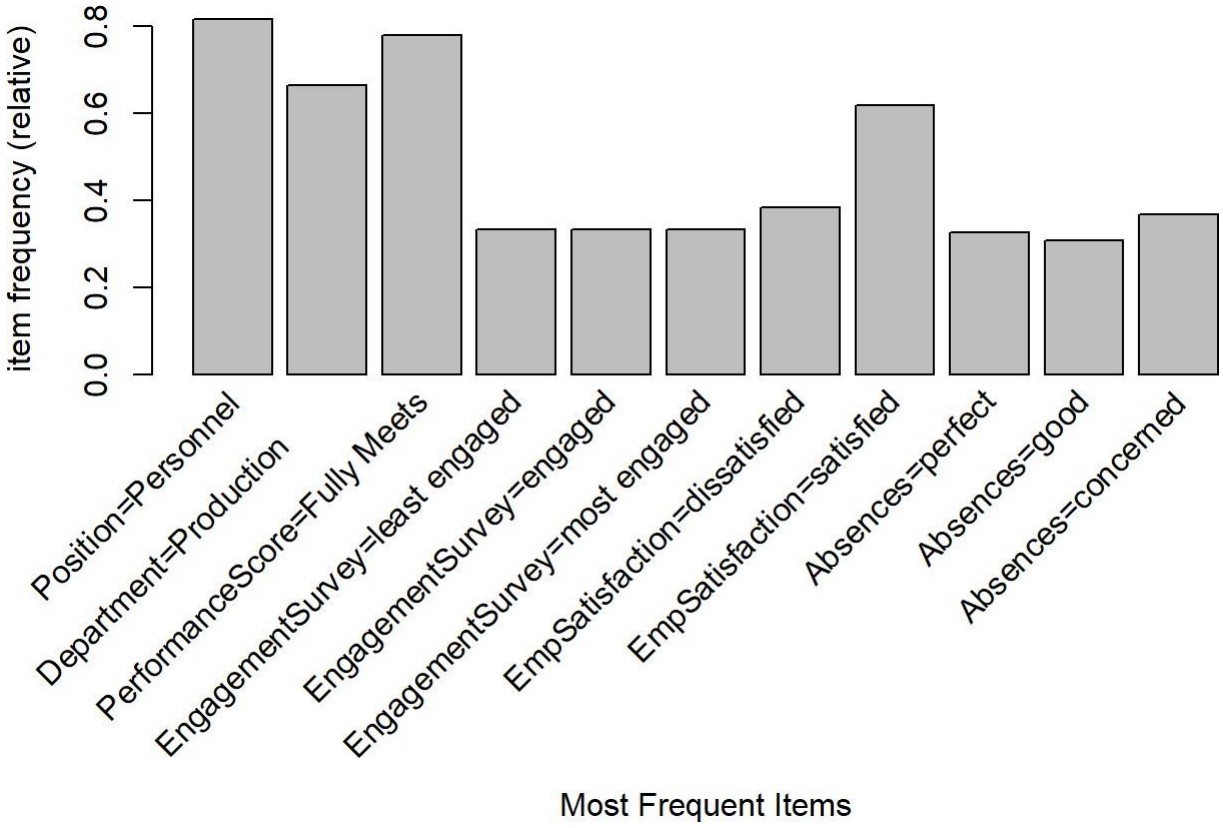
EmpSatisfaction=satisfied, 187 times, which means that 61.72% of employees are satisfied.

EmpSatisfaction=dissatisfied, 116 times, which means that 38.28% of employees are diassatisfied.

Department=Production, 201 times, which means that 66.36% of employees work in production department.

We can visualize most frequent items by using itemFrequencyPlot() function, with minimum support of 0.3. As showns in the following plot, this results in a histogram showing eight items in the data with at least 30 percent support:

```
itemFrequencyPlot(datatrans, support = 0.3, xlab= "Most Frequent Items")
```



Let's have a look at the first 3 elements of our sparse matrix, using the inspect() function from arules package .

```
inspect(datatrans[1:3])
```

```
##     items                                transactionID
## [1] {Salary=average,
##      Position=Personnel,
##      Department=Production        ,
##      PerformanceScore=Exceeds,
##      EngagementSurvey=most  engaged,
##      EmpSatisfaction=satisfied,
##      Absences=perfect}                    1
## [2] {Salary=high,
##      Position=Personnel,
##      Department=IT/IS,
##      PerformanceScore=Fully  Meets,
##      EngagementSurvey=most  engaged,
##      EmpSatisfaction=dissatisfied,
##      Absences=concerned}                  2
## [3] {Salary=good,
##      Position=Personnel,
##      Department=Production        ,
```

```
##       PerformanceScore=Fully Meets,
##       EngagementSurvey=least engaged,
##       EmpSatisfaction=dissatisfied,
##       Absences=perfect}                              3
```

These transactions match the first 3 rows of our data set.

Training a model on the data

Below we demonstrate association rule mining with apriori() function from arules package which we already called. Apriori counts transactions to find frequent itemsets and then derive association rules from them. With settings of: supp=0.05, which is the minimum support of rules; conf=0.25, which is the minimum confidence of rules; and minlen=2 & maxlen=5, which are the minimum and the maximum length of rules, here is our rule:

```
myrule <- apriori(datatrans, parameter = list(minlen=2, maxlen=5,support=0.05,   confidence=0.25))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.25    0.1    1 none FALSE            TRUE       5    0.05      2
##  maxlen target  ext
##       5  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort  verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[25 item(s), 303 transaction(s)] done [0.00s].
## sorting and recoding items ... [20 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5
```

```
## Warning in apriori(datatrans, parameter = list(minlen = 2, maxlen = 5, support
## = 0.05, : Mining stopped (maxlen reached). Only patterns up to a length of 5
## returned!
```

```
##  done [0.00s].
## writing ... [1632 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
myrule
```

```
## set of 1632 rules
```

myrule object contains a set of 1632 association rules.

Evaluating model performance

```
summary(myrule)
```

```
## set of 1632 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4   5
## 192 619 622 199
##
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.000   3.000   4.000   3.507   4.000   5.000
##
## summary of quality measures:
##     support           confidence         coverage             lift
##  Min.   :0.05281   Min.   :0.2500   Min.   :0.05281   Min.   :0.6613
##  1st Qu.:0.05941   1st Qu.:0.3556   1st Qu.:0.09571   1st Qu.:0.9914
##  Median :0.07591   Median :0.6137   Median :0.16832   Median :1.0834
##  Mean   :0.10156   Mean   :0.5870   Mean   :0.20117   Mean   :1.1612
##  3rd Qu.:0.11634   3rd Qu.:0.8000   3rd Qu.:0.24092   3rd Qu.:1.2000
##  Max.   :0.64686   Max.   :1.0000   Max.   :0.81518   Max.   :7.8194
##      count
##  Min.   : 16.00
##  1st Qu.: 18.00
##  Median : 23.00
##  Mean   : 30.77
##  3rd Qu.: 35.25
##  Max.   :196.00
##
## mining info:
##       data ntransactions support confidence
##  datatrans            303    0.05       0.25
```

In the part of 'summary of quality measures' we can see that the max support is 0.646 which represent the most frequent item. In our case it's Position=Personnel. The max count is 196 which also present the number of employees who are in Personnel position. The lift is a measure of how much more likely one item is to be present(exist) relative to its typical present(existing) rate. In this rule the maximum lift is 7.

Determine the factors that influence the variable EmpSatisfaction

We are interested in only rules with rhs indicating employee satisfaction , so I set ' rhs=c("EmpSatisfaction=dissatisfied", "EmpSatisfaction=satisfied") ' in appearance to make sure that only "EmpSatisfaction=dissatisfied" and "EmpSatisfaction=satisfied" will appear in the right hand side of rules. All other items can appear in the left hand side, as set with default="lhs". Rules are sorted by lift to make high-lift rules appear first.

```
rule <- apriori(datatrans, parameter = list(minlen=2, maxlen=5,support=0.05,  confidence=0.25),app
earance= list(rhs=c("EmpSatisfaction=dissatisfied", "EmpSatisfaction=satisfied"), default="lhs"),
control = list(verbose=F))
rule
```

```
## set of 245 rules
```

```
inspect(sort(rule, by = "lift")[1:9]) #Sorting the set of association rules
```

```
##     lhs                                    rhs                                    support confidence
coverage      lift count
## [1] {Salary=average,
##      EngagementSurvey=least engaged} => {EmpSatisfaction=dissatisfied} 0.05940594   0.6206897 0.
```

```
09570957 1.621284     18
## [2] {EngagementSurvey=least  engaged,
##      Absences=perfect}               =>  {EmpSatisfaction=dissatisfied} 0.06600660  0.5882353 0.
11221122 1.536511     20
## [3] {Position=Personnel,
##      EngagementSurvey=least  engaged,
##      Absences=perfect}               =>  {EmpSatisfaction=dissatisfied} 0.05280528  0.5333333 0.
09900990 1.393103     16
## [4] {Salary=below average,
##      Department=Production        ,
##      PerformanceScore=Fully  Meets,
##      Absences=good}                  => {EmpSatisfaction=satisfied}     0.05280528  0.8000000 0.
06600660 1.296257     16
## [5] {Salary=average,
##      EngagementSurvey=most engaged}  => {EmpSatisfaction=satisfied}     0.06270627  0.7916667 0.
07920792 1.282754     19
## [6] {Salary=average,
##      Position=Personnel,
##      EngagementSurvey=most  engaged}  => {EmpSatisfaction=satisfied}     0.06270627  0.7916667 0.
07920792 1.282754     19
## [7] {Salary=average,
##      Department=Production        ,
##      EngagementSurvey=most  engaged}  => {EmpSatisfaction=satisfied}     0.05940594  0.7826087 0.
07590759 1.268077     18
## [8] {Salary=average,
##      Position=Personnel,
##      Department=Production        ,
##      EngagementSurvey=most  engaged}  => {EmpSatisfaction=satisfied}     0.05940594  0.7826087 0.
07590759 1.268077     18
## [9] {PerformanceScore=Fully Meets,
##      EngagementSurvey=engaged,
##      Absences=good}                  => {EmpSatisfaction=satisfied}     0.08250825  0.7812500 0.
10561056 1.265876     25
```

245 rules satisfy those criteria.

From the first rule we can be 62% sure that when an employee is least engaged and have an average salary is dissatisfaction. This rule is involved in 5.94% of the entire transactions, and the lift implies that an employees who have these characteristics are 1.62 times more likely to be dissatisfied than the typical employee.

In the above result, the 7th and 8th rules show that we can be 78,2% sure that an employees who are most engaged, have and average salary and work in production department are of the same satisfaction degree, with lift of 1,268 which indicates employees with same characteristics are 1,268 likely to be satisfied.

Taking subsets of association rules

Satisfied employees

```
SatisfiedRule=subset(rule, items  %in% "EmpSatisfaction=satisfied")
inspect(sort(SatisfiedRule, by = "lift")[7:11])
```

```
##     lhs                                rhs                                support confidence    cover
age      lift count
## [1] {Salary=average,
##      Absences=good}                 => {EmpSatisfaction=satisfied} 0.05610561  0.7727273 0.07260
726 1.252066     17
## [2] {Salary=average,
```

```
##       EngagementSurvey=engaged}      => {EmpSatisfaction=satisfied} 0.05610561   0.7727273 0.07260
726 1.252066     17
## [3] {Position=Personnel,
##       PerformanceScore=Fully  Meets,
##       EngagementSurvey=engaged,
##       Absences=good}                 =>  {EmpSatisfaction=satisfied}  0.06600660   0.7692308 0.08580
858 1.246401     20
## [4] {Salary=below average,
##       PerformanceScore=Fully  Meets,
##       Absences=good}                 =>  {EmpSatisfaction=satisfied}  0.05280528   0.7619048 0.06930
693 1.234530     16
## [5] {Salary=below average,
##       Position=Personnel,
##       PerformanceScore=Fully  Meets,
##       Absences=good}                 =>  {EmpSatisfaction=satisfied}  0.05280528   0.7619048 0.06930
693 1.234530     16
```

The 4th and the 5th rules indicates that a employees with below average salary, with great performance and a good absences(he's been absent less than 6 days), are 76% of time satisfied, with lift of 1.234 which means that these characteristics are dependent. We can conclude that these employees are still young, newly hired and eager to work.

Dissatisfied employees

```
DisatRule=subset(rule, items %in% "EmpSatisfaction=dissatisfied")
DisatRule
```

```
## set of 87 rules
```

```
inspect(sort(DisatRule, by = "lift")[9:13])
```

```
##     lhs                                   rhs                                 support confidence  c
overage     lift count
## [1] {EngagementSurvey=least engaged,
##       Absences=concerned}              => {EmpSatisfaction=dissatisfied} 0.06600660   0.4444444 0.
1485149 1.160920     20
## [2] {Position=Personnel,
##       PerformanceScore=Fully  Meets,
##       EngagementSurvey=least engaged} => {EmpSatisfaction=dissatisfied} 0.08250825   0.4385965 0.
1881188 1.145644     25
## [3] {Position=Personnel,
##       Department=IT/IS,
##       PerformanceScore=Fully  Meets}    => {EmpSatisfaction=dissatisfied} 0.05610561   0.4358974 0.
1287129 1.138594     17
## [4] {Salary=good,
##       Department=Production        }    => {EmpSatisfaction=dissatisfied} 0.07590759   0.4339623 0.
1749175 1.133539     23
## [5] {Salary=high,
##       Position=Personnel,
##       PerformanceScore=Fully  Meets}    => {EmpSatisfaction=dissatisfied} 0.06270627   0.4318182 0.
1452145 1.127939     19
```

Rule 2 show s that we can be 43.85 % sure that employees who are in Personnel position and their performance fully meets are of the same dissatisfaction degree, with lift of 1.14 which indicates employees with same characteristics are 1,14 likely to be dissatisfied.

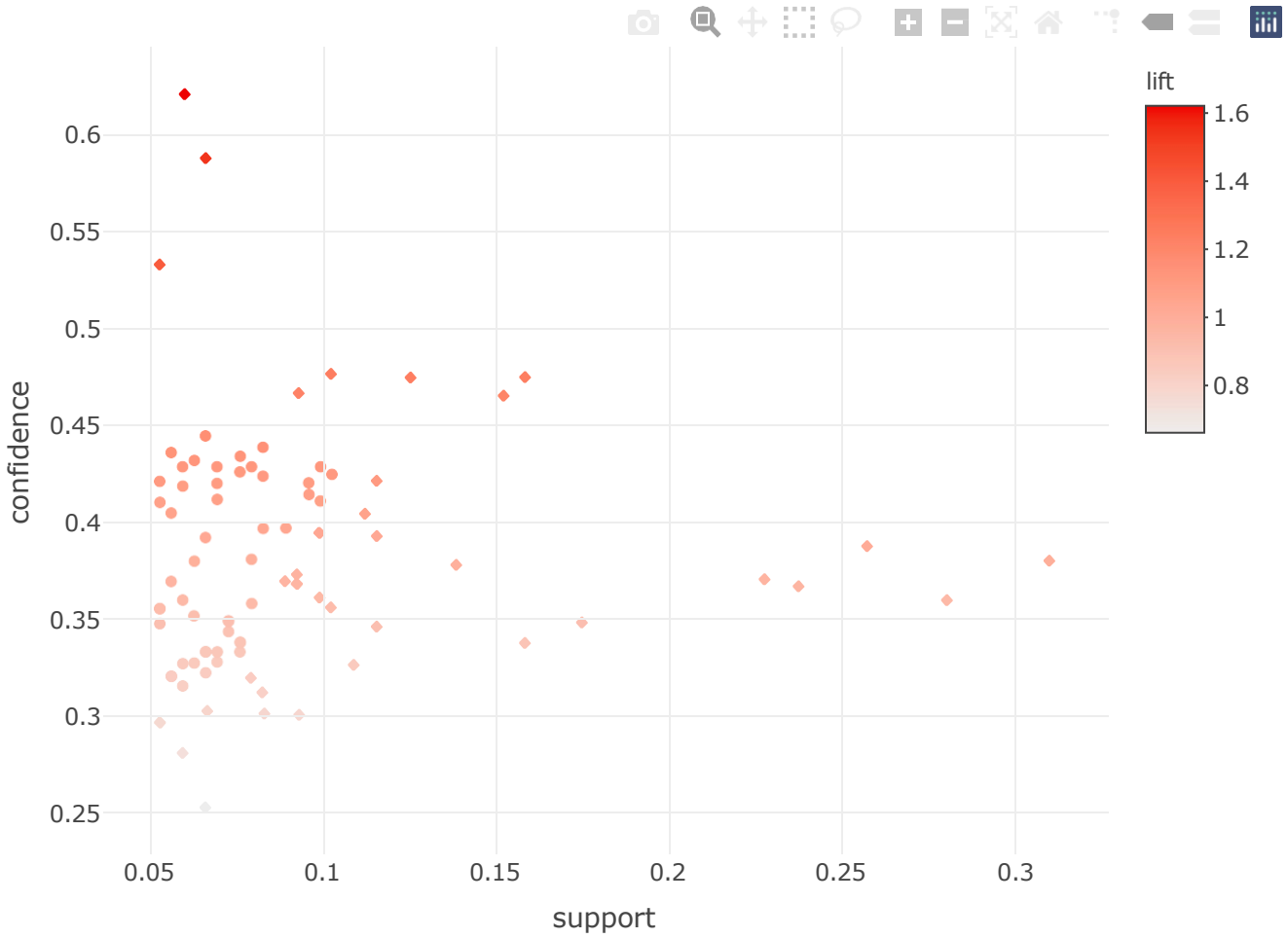
Rules visualization for:

Dissatiesied rules

In this section, we introduce arulesViz, a package dedicated to plot association rules, generated by the arules package.

```
plot(DisatRule, engine = "htmlwidget")
```

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

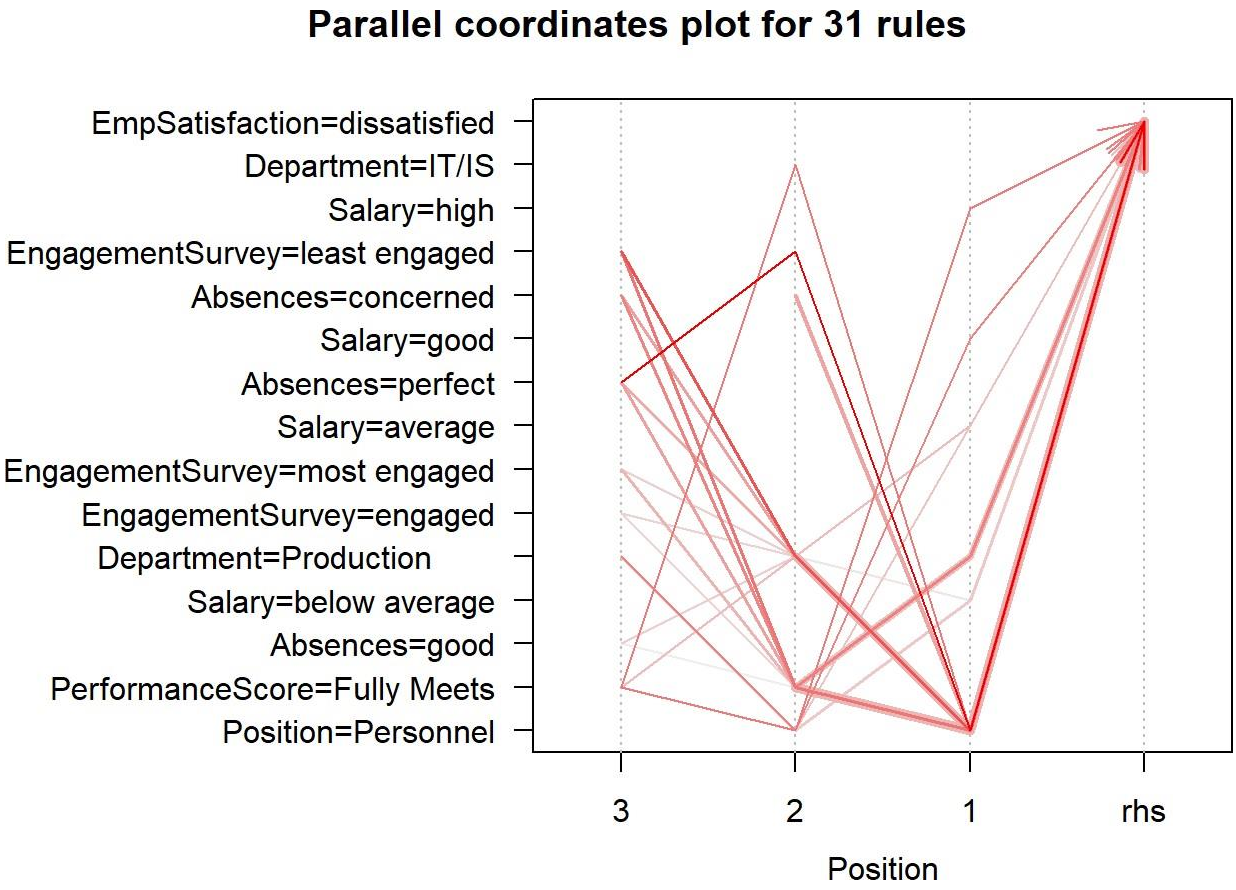


We can notice most of items have support between 0.05 and 0.1. The darker points the higher the lift and the confident.

Rule 18:{Salary=average, EngagementSurvey= least engaged} => dissatisfied, have high lift(1.6) and high confidence(0.62). So it's a reliable rule.

Rule 42: {performaceScore =fully meets, EngagemntSurvey= engaged} => dissatisfied, have low lift(0.787) and confidence(0.301). It's unreliable rule.

```
plot(DisatRule[50:80] , method = "paracoord")
```



The width of the arrows represents support and the intensity of the color represent confidence.

If performance fully meets with department=IT/IS with Position=Personnel exist, it lead us to dissatisfied. From HR analysis point of view this may be explained that an employee who fits these characteristics is unhappy with his current position and been waiting for a promotion for a long time.

let's look at the most intense rule (high confident level);

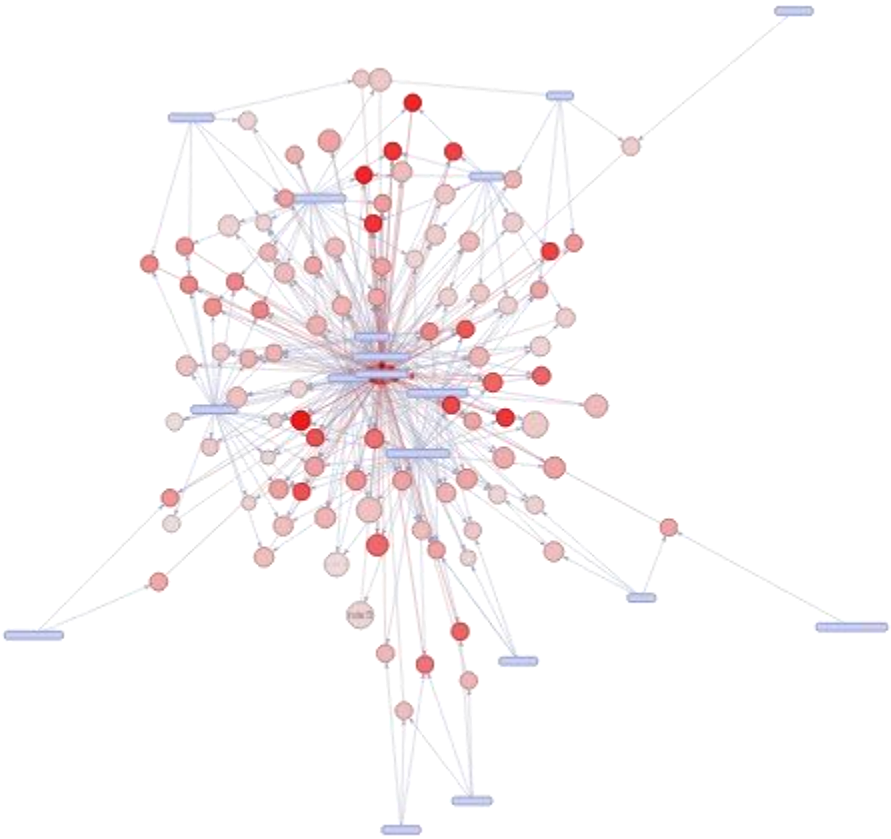
If absence= perfect(been absent a lot) with engagement=least engaged and with position=personnel leads to dissatisfied.

Satisfied rules

```
plot(SatisfiedRule, method = "graph", engine = "htmlwidget")
```

```
## Warning: Too many rules supplied. Only plotting the best 100 rules using lift
## (change control parameter max if needed)
```

Select by id



Rule 100 : {Salary=below average,Department=Production ,PerformanceScore=Fully Meets,Absences=good}=> {EmpSatisfaction=satisfied} have lift of 1.3 and confidence level of 80% => Reliable rule.

Multivariate Analysis

Normality test

To test out the normal distribution, we can use the mean, median, and mode for some of the variable:

```
mean(HRdata$Salary)
```

```
## [1] 69292.32
```

```
median(HRdata$Salary)
```

```
## [1] 62910
```

```
sd(HRdata$Salary)
```

```
## [1] 25406.09
```

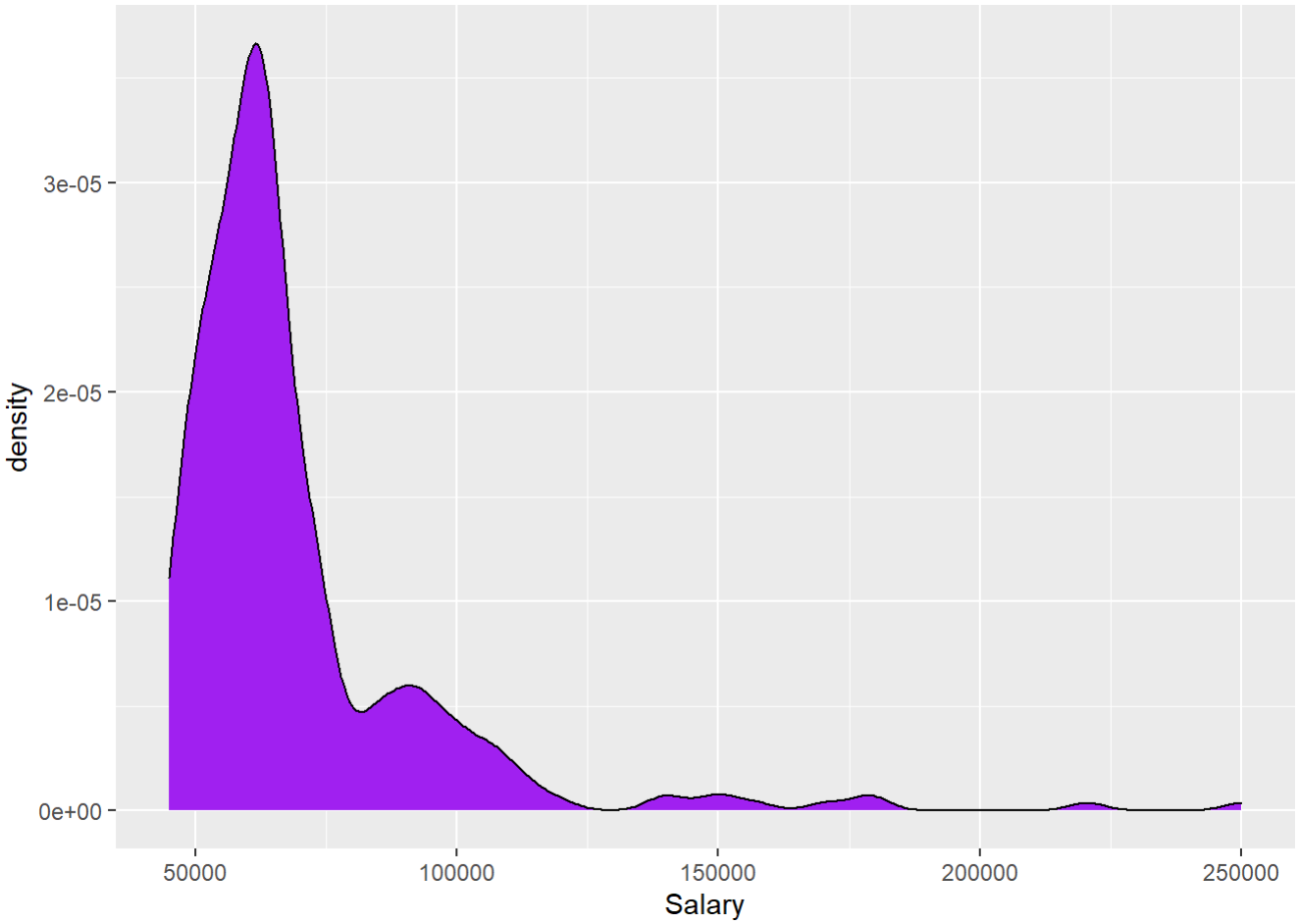
```
var(HRdata$Salary)
```

```
## [1] 645469550
```

```
skewness(HRdata$Salary)  #positively  skewed
```

```
## [1] 3.25132
```

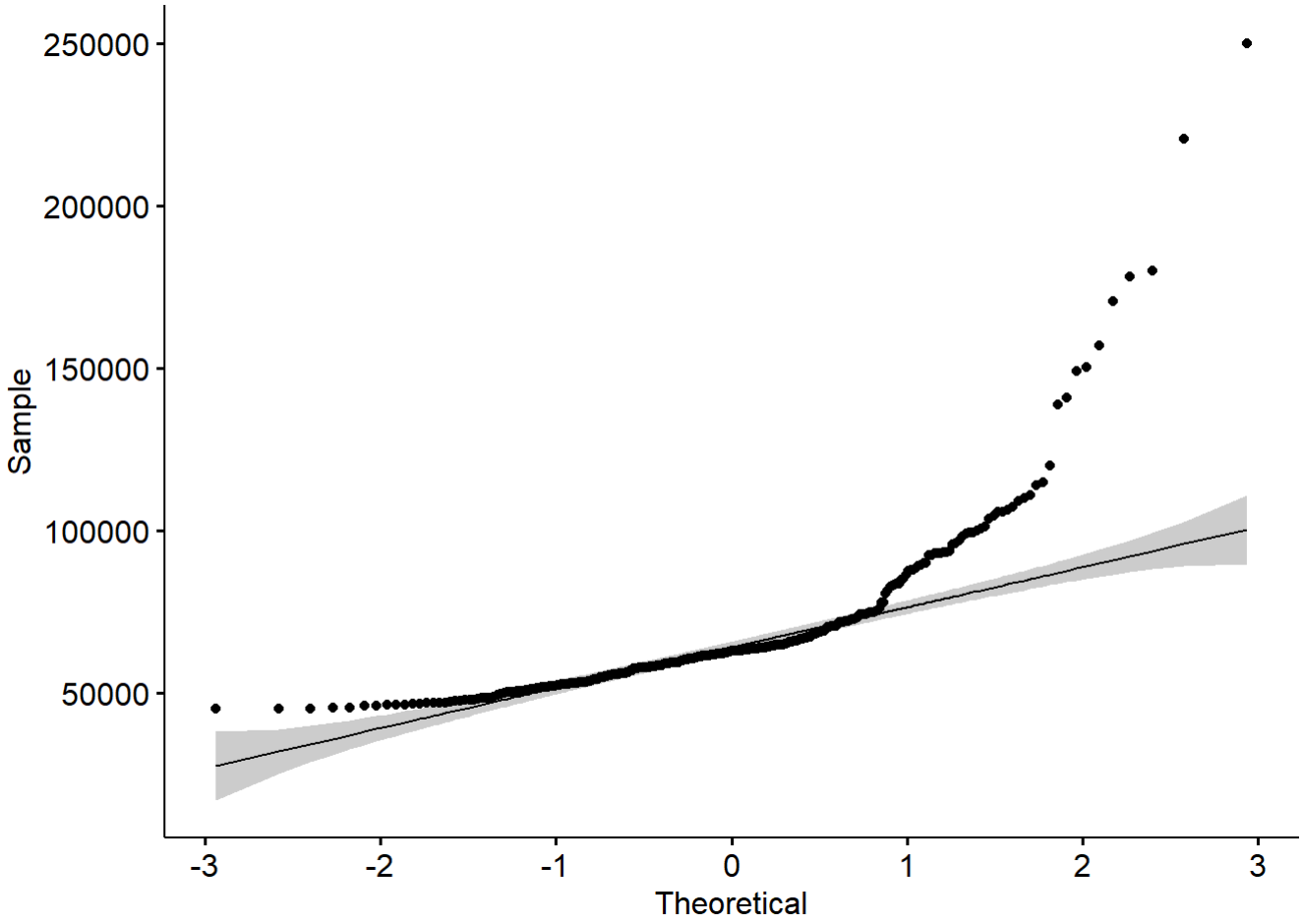
```
ggplot(data=HRdata, aes(Salary)) + geom_density(fill="purple")
```



From the preceding image, we can conclude that the salary variable is positively skewed because of the presence of some outlier values on the right-hand side of the distribution.

To prove that we can graph QQ plot. In a QQ plot, each observation is plotted as a single dot. If the data are normal, the dots should form a straight line.

```
library("ggpubr")
ggqqplot(HRdata$Salary)
```



Not all the points fall approximately along this reference line, we can't assume normality.

Hypothesis test for a test of normality

Null hypothesis: The data is normally distributed. If p> 0.05, normality can be assumed

```
shapiro.test(HRdata$Salary)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  HRdata$Salary
## W = 0.68867, p-value < 2.2e-16
```

For the skewed data, p-value < 2.2e-16 suggesting strong evidence of non-normality and a nonparametric test should be used

Now let's try to understand a case where skewed data can be used to answer any hypothesis. Suppose the variable SALARY with a mean of 69292.32 and a standard deviation of 25406.09 What is the probability that a new employee have a salary of 110000?

```
pnorm(110000,mean(HRdata$Salary),sd(HRdata$Salary),lower.tail  =  F)
```

```
## [1] 0.05454681
```

Hence the required probability that a new employee would have a salary of 110000 is 5.45%,

Hypothesis testing

H0: both males and females have the same average salary

H11: average salary of males ! average salary of females

```
table(HRdata$Sex,HRdata$Salary  >65000)
```

```
##
##       FALSE  TRUE
##    F    115    56
##    M     74    58
```

there is 56 out 171 female who have a salary greater the 65000. and 58 out of 132 male who have a salary greater than 65000.

```
p1 <- 56/171
p0<- 0.11
n <- length(HRdata$Salary)
z <- (p1-p0)/sqrt(p0*(1-p0)/n)
z
```

```
## [1] 12.09929
```

Computing the critical value at 5% alpha level

```
alpha = .05
z1 = qnorm(1-alpha)
z1
```

```
## [1] 1.644854
```

```
ifelse(z > z1,"Reject the Null Hypothesis","Accept the Null Hypothesis")
```

```
## [1] "Reject the Null Hypothesis"
```

This proves what we've already noticed earlier on the graph.

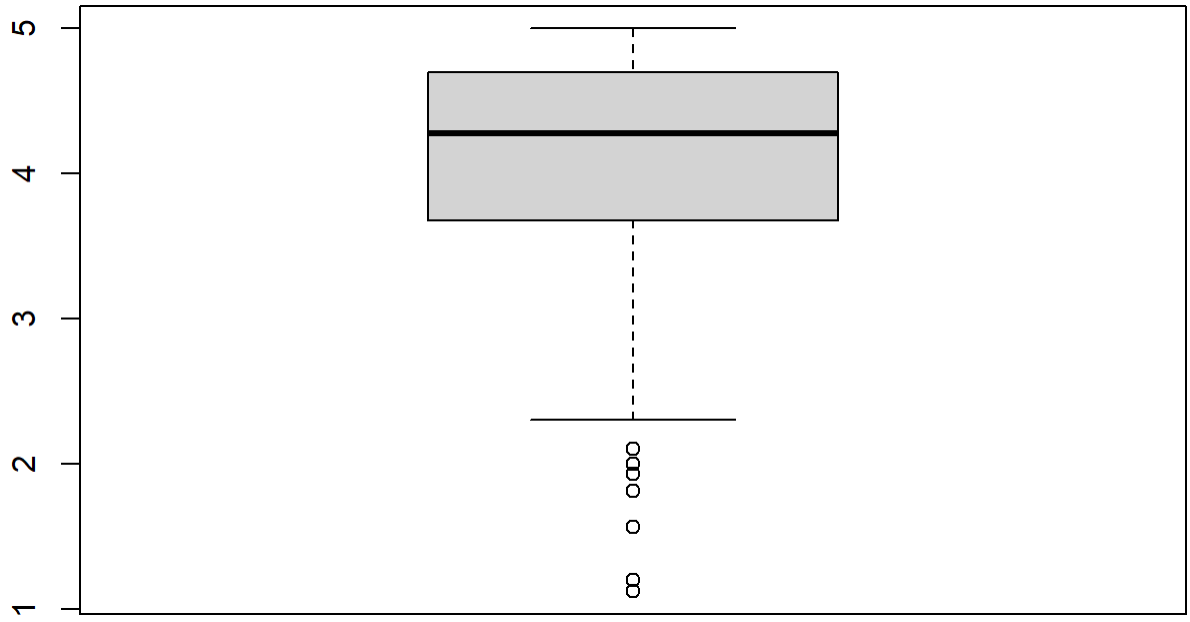
One-Sample t Test & Confidence Interval for engagement

In this part I'm going to apply hypothesis testing on the engagement variable. In this case, when we want to check if the sample mean represents the population mean.

```
class(HRdata$EngagementSurvey)
```

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

```
boxplot(HRdata$EngagementSurvey)
```



Examine the plot of the data to help me choose mu.

H0: mu =< 4.2

One sided 95% confidence level interval for mu.

```
t.test(HRdata$EngagementSurvey, mu=4.2, conf = 0.95)
```

```
##
##  One Sample t-test
##
## data:  HRdata$EngagementSurvey
## t = -2.0562, df = 302, p-value = 0.04062
## alternative hypothesis: true mean is not equal to 4.2
## 95 percent confidence interval:
##  4.016310 4.195967
## sample estimates:
## mean of x
##  4.106139
```

In above, p-value = 0.04062, this value is less than alpha value, and thus we have to reject the null hypothesis. Here the null hypothesis was that the average engagement of the employees is 4.2

95 percent confidence interval: 4.016310 4.195967

The 95% CI does not includes the 4.2.

Two sample t-test

This method in appropriate for examine the difference in mean of 2 populations. It can also examine the relationship between a numeric outcome and a categorical variable with 2 levels. we will be exploring the relationship between performance score and

engagement results.

Now we want to know is there statistical significant difference between two groups in term of the average. So here we going to select only 2 levels of the Performance score variable

H0: mean of engagement of an exceed performance = engagement of fully meets performance (there is no difference between engagement of an employee who's performance fully meets and an employee who's performance exceeds)

If p-value is very small we reject the null hypothesis and accept H1(that there is diff).

So let's test this out.

```
tdata<- HRdata %>%
  select(PerformanceScore, EngagementSurvey) %>%
  filter(PerformanceScore== "Exceeds" |
           PerformanceScore== "Fully Meets")
```

```
t.test(data= tdata, EngagementSurvey~PerformanceScore, mu =0 , alt= "two.sided", conf= 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  EngagementSurvey by PerformanceScore
## t = 3.1986, df = 64.19, p-value = 0.002146
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.09167297 0.39662270
## sample estimates:
##     mean in group Exceeds mean in group Fully Meets
##                  4.481944                  4.237797
```

Since p-value is less than 0.05(p-value = 0.001932) it means we reject the null hypothesis: the average engagement rate of employees that exceed in they performance is different than of fully meets. From the this result the difference between engagement is about 0.25 points. CI: 0.09167297 and 0.39662270 ; zero did not appear in at least 95% of the experiments, and thus we conclude that our decision to reject the null hypothesis is correct.

```
var(tdata$EngagementSurvey[tdata$PerformanceScore=="Exceeds"])
```

```
## [1] 0.1533361
```

```
var(tdata$EngagementSurvey[tdata$PerformanceScore=="Fully  Meets"])
```

```
## [1] 0.369739
```

As you can see the Var of fully meets group is almost double of exceeds group.

Prediction who's going to terminate

I'm going to use *LogisticRegression* to see who's going to terminate(leave work). So why logistic regression?

Logistic regression answers the question "will it happen or not" while Linear Regression answers "how much". Logistic Regression is used when the response variable has 2 outcomes ('yes' or 'no', '0' or '1').

Split the data

First we will split our data into a training (75%) and testing (25%) data sets so we can assess how well our model performs on an out-of-sample data set.

```
smp_siz = floor(0.75*nrow(HRdata)) ## creates a value for dividing the data into train and test. 7
5% of the number of rows in the data set.
set.seed(123) #to have same random numbers generated
dt<- sample(seq_len(nrow(HRdata)), size =smp_siz)
train <- HRdata[dt, ] #creates the training data set with row numbers stored in sample
test <- HRdata[-dt, ]
```

```
prop.table(table(train$Termd))
```

```
##
##         0         1
## 0.6696035 0.3303965
```

```
prop.table(table(test$Termd))
```

```
##
##         0         1
## 0.6184211 0.3815789
```

We can see that we have almost equal percentage of distribution for Active(0) and Terminated(1) employees for both train and test data sets.

Train the model using the training data and glm() function.

The glm() function fits generalized linear models, a class of models that includes logistic regression. The syntax of the glm() function is similar to that of lm(), except that we must pass the argument 'family = binomial 'in order to tell R to run a logistic regression rather than some other type of generalized linear model.

Covert variables to Factor variables

```
HRdata$Termd <- as.factor(HRdata$Termd)
HRdata$PerformanceScore <- as.factor(HRdata$PerformanceScore)
```

Define full and null models and do step procedure

```
model.null = glm(Termd ~ 1,family = "binomial", train)

model.full = glm(Termd ~  MaritalStatusID  + DeptID  + PerfScoreID      + Salary    + MaritalDesc    +
EngagementSurvey  + EmpSatisfaction +  Absences + SpecialProjectsCount  +  DaysLateLast30  +  Perf
ormanceScore + CitizenDesc +  HispanicLatino  + RaceDesc, family ="binomial", train)

step(model.null,
     scope = list(upper=model.full),
     direction="both",
     data=train)
```

```
## Start:  AIC=290.04
## Termd ~ 1
##
##                          Df  Deviance    AIC
## + DaysLateLast30          1   283.40 287.40
## + SpecialProjectsCount    1   283.46 287.46
## + Salary                  1   284.66 288.66
## + DeptID                  1   284.97 288.97
## + MaritalDesc             4   279.40 289.40
## + PerfScoreID             1   285.81 289.81
## <none>                        288.04 290.04
## + Absences                1   286.30 290.30
## + MaritalStatusID         1   287.16 291.16
## + CitizenDesc             2   285.74 291.74
## + PerformanceScore        3   284.02 292.02
## + EmpSatisfaction         1   288.03 292.03
## + EngagementSurvey        1   288.04 292.04
## + HispanicLatino          2   287.15 293.15
## + RaceDesc                4   283.78 293.78
##
## Step:  AIC=287.4
## Termd ~ DaysLateLast30
##
##                          Df  Deviance    AIC
## + SpecialProjectsCount    1   279.49 285.49
## + EngagementSurvey        1   279.68 285.68
## + MaritalDesc             4   274.01 286.01
## + Salary                  1   280.65 286.65
## + DeptID                  1   280.88 286.88
## <none>                        283.40 287.40
## + Absences                1   281.69 287.69
## + PerformanceScore        3   277.86 287.86
## + MaritalStatusID         1   282.06 288.06
## + CitizenDesc             2   280.96 288.96
## + EmpSatisfaction         1   283.15 289.15
## + PerfScoreID             1   283.32 289.32
## - DaysLateLast30          1   288.04 290.04
## + HispanicLatino          2   282.63 290.63
## + RaceDesc                4   279.56 291.56
##
## Step:  AIC=285.49
## Termd ~ DaysLateLast30 + SpecialProjectsCount
##
##                          Df  Deviance    AIC
## + MaritalDesc             4   270.06 284.06
## + EngagementSurvey        1   276.32 284.32
## <none>                        279.49 285.49
## + Absences                1   277.83 285.83
## + PerformanceScore        3   273.96 285.96
## + MaritalStatusID         1   278.42 286.42
## + CitizenDesc             2   276.47 286.47
## + Salary                  1   279.06 287.06
## + EmpSatisfaction         1   279.21 287.21
## - SpecialProjectsCount    1   283.40 287.40
## + PerfScoreID             1   279.41 287.41
## - DaysLateLast30          1   283.46 287.46
```

```
## + DeptID                 1    279.49 287.49
## + HispanicLatino         2    278.82 288.82
## + RaceDesc               4    275.39 289.39
##
## Step:  AIC=284.06
## Termd ~ DaysLateLast30 + SpecialProjectsCount + MaritalDesc
##
##                         Df  Deviance    AIC
## + EngagementSurvey       1    266.93 282.93
## <none>                        270.06 284.06
## + PerformanceScore       3    264.08 284.08
## + CitizenDesc            2    266.64 284.64
## + Absences               1    268.78 284.78
## - MaritalDesc            4    279.49 285.49
## + Salary                 1    269.78 285.78
## + EmpSatisfaction        1    269.84 285.84
## - SpecialProjectsCount   1    274.01 286.01
## + DeptID                 1    270.01 286.01
## + PerfScoreID            1    270.03 286.03
## - DaysLateLast30         1    274.66 286.66
## + HispanicLatino         2    269.31 287.31
## + RaceDesc               4    266.34 288.34
##
## Step:  AIC=282.93
## Termd ~ DaysLateLast30 + SpecialProjectsCount + MaritalDesc +
##     EngagementSurvey
##
##                         Df Deviance    AIC
## <none>                       266.93 282.93
## + Absences               1    265.57 283.57
## + CitizenDesc            2    263.62 283.62
## + PerformanceScore       3    261.87 283.87
## - EngagementSurvey       1    270.06 284.06
## - MaritalDesc            4    276.32 284.32
## - SpecialProjectsCount   1    270.33 284.33
## + Salary                 1    266.56 284.56
## + DeptID                 1    266.63 284.63
## + EmpSatisfaction        1    266.77 284.77
## + PerfScoreID            1    266.82 284.82
## + HispanicLatino         2    265.99 285.99
## + RaceDesc               4    262.96 286.96
## - DaysLateLast30         1    274.66 288.66
```

```
##
## Call:  glm(formula = Termd ~ DaysLateLast30 + SpecialProjectsCount +
##     MaritalDesc + EngagementSurvey, family = "binomial", data = train)
##
## Coefficients:
##           (Intercept)         DaysLateLast30   SpecialProjectsCount
##               -1.5630                 0.3742                -0.1279
##    MaritalDescMarried   MaritalDescSeparated      MaritalDescSingle
##               -0.8215                -2.4281                -1.1695
##    MaritalDescWidowed       EngagementSurvey
##               -0.3164                 0.4190
##
## Degrees of Freedom: 226 Total (i.e. Null);  219 Residual
```

```
## Null Deviance:        288
## Residual Deviance: 266.9      AIC: 282.9
```

Training model

Final Logistic Regression Model is:

```
model <- glm(formula = Termd ~ DaysLateLast30 + SpecialProjectsCount +
    MaritalDesc + EngagementSurvey, family = "binomial", data = train)
```

By using function summary() we obtain the results of our model:

```
summary(model)
```

```
##
## Call:
## glm(formula = Termd ~ DaysLateLast30 + SpecialProjectsCount +
##     MaritalDesc + EngagementSurvey, family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7512  -0.9092  -0.6677   1.1793   2.1308
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.56299    1.12227  -1.393   0.1637
## DaysLateLast30         0.37418    0.13826   2.706   0.0068 **
## SpecialProjectsCount  -0.12791    0.07243  -1.766   0.0774 .
## MaritalDescMarried    -0.82153    0.47621  -1.725   0.0845 .
## MaritalDescSeparated  -2.42813    1.13615  -2.137   0.0326 *
## MaritalDescSingle     -1.16948    0.47768  -2.448   0.0144 *
## MaritalDescWidowed    -0.31640    1.48437  -0.213   0.8312
## EngagementSurvey       0.41904    0.24107   1.738   0.0822 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 288.04  on 226  degrees of freedom
## Residual deviance: 266.93  on 219  degrees of freedom
## AIC: 282.93
##
## Number of Fisher Scoring iterations: 4
```

The AIC or Akaike Information Criteria in this case is 288.74 . The AIC is the measure of fit which penalizes model for the number of model coefficients.

-The p-value in the last column is more than 0.05 for the variables “EngagementSurvey” and “MaritalDescWidowed”, we consider them to be statistically insignificant . As for DaysLateLast30 , MaritalDescSingle and MaritalDescSeparated has the lowest p-value impacts the “Termd” value in this regression model.

-In the case of DaysLateLast30, we see that the estimate is positive, meaning that if employees who been late to work frequently during the last 30 days are significantly more likely to leave.

-Our MaritalDescSeparated and MaritalDescSingle coefficients are significant and negative. This means single and separate employees are less likely to leave than married employees .

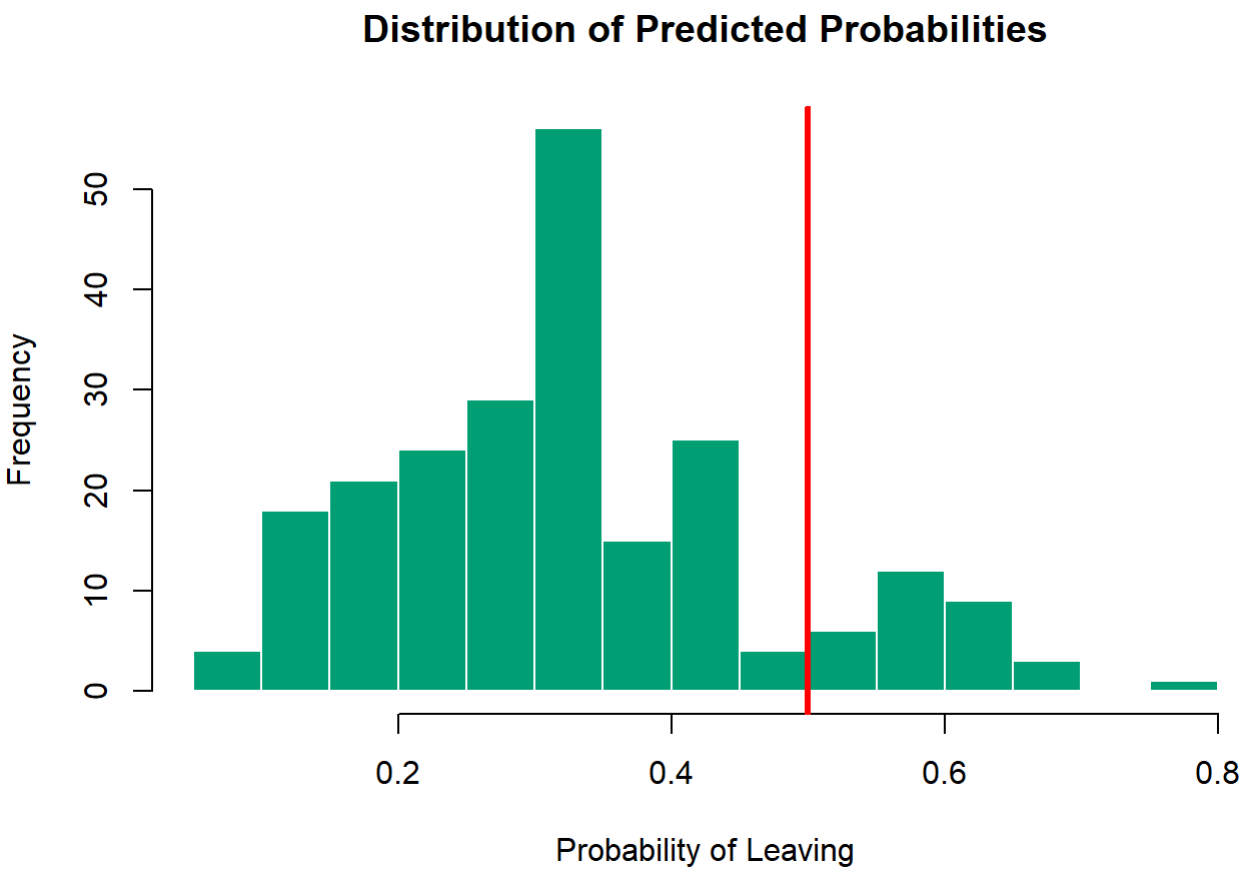
Use the Model to Make Predictions

Once we've fit the logistic regression model, we can then use it to calculate probability of termination(leaving) for each employee in test data set.

Let's see the distribution.

```
# A color palette
cbPalette <- c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79
A7")

hist(model$fitted.values, main = "Distribution of Predicted Probabilities",
     xlab = "Probability of Leaving", col = cbPalette[4], border = F, breaks = 25)
abline(v = .5, col = "red", lwd = 3)
```




```
prop.table(table(model$fitted.values>=  .5))
```

```
##
##      FALSE       TRUE
## 0.8634361 0.1365639
```

The histogram for the training data show us that we have 13.65% of our employees with a probability of leaving at 50% or higher.

Now let's see how the model will do with the test data.

Predicting with test Data

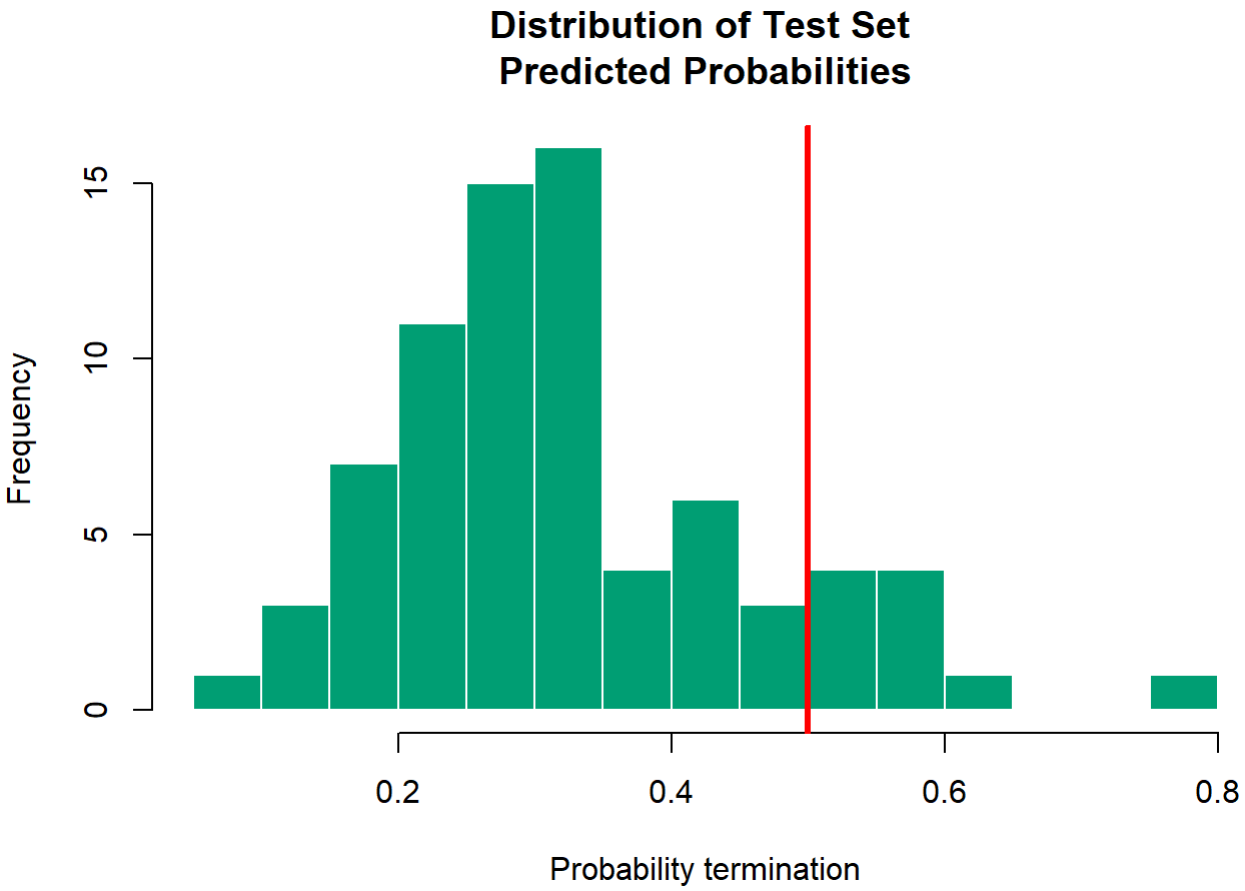
Measuring Accuracy

To see how accurate our model is, we need use it to predict the outcomes for our test data set.

Using predict() function and type = “response” to get the predicted probabilities for each person.

```
testpredict<- predict(model, test, type = "response")

hist(testpredict, main = "Distribution of Test Set \nPredicted Probabilities",
     xlab = "Probability termination", col = cbPalette[4], border = F, breaks = 15)
abline(v = .5, col = "red", lwd = 3)
```




```
prop.table(table(testpredict >= .5))
```

```
##
##     FALSE      TRUE
## 0.8684211 0.1315789
```

It’s similar to our training set predictions, we see that 13.15% have a predicted probability greater than 0.5.

We can also use it to make predictions about whether or not an employee will get terminated based on their engagement, MaritalDesc,…:

```
newPredict <- data.frame(DaysLateLast30 = 0 , SpecialProjectsCount = 5   , MaritalDesc="Married"   ,
EngagementSurvey=4.96)
testpredict1<- predict(model, newPredict, type = "response")
testpredict1
```

```
##         1
## 0.2797673
```

The probability of an employee who's married, never been late in the last 30 days, done 5 projects and has an engagement rate of 4.96 (he's well engaged in the CO.), has a probability of being terminated(leaving) of 0.279.

The Confusion Matrix

To check accuracy, we'll build a "confusion matrix".

```
#validate the model confusion matrix
prop.table(table(test$Termd))
```

```
##
##         0         1
## 0.6184211 0.3815789
```

```
accuracy <- table(testpredict > .5, test$Termd) # confusion matrix

addmargins(table(testpredict > .5, test$Termd))
```

```
##
##           0  1 Sum
##   FALSE  42 24  66
##   TRUE    5  5  10
##   Sum    47 29  76
```

The values on the right represent the predicted outcome (where False = 0 and True = 1). The values for the columns represent the real, observed outcomes.

Accuracy

Let's measure the prediction percentage using confusion matrix.

```
(accuracy[[1,1]] + accuracy [[2,2]]) / sum(accuracy)
```

```
## [1] 0.6184211
```

The accuracy here indicates that 61.8% of the observations in our data are active employees(not terminated)

Cutoff

```
addmargins(accuracy)
```

```
##
##           0  1 Sum
##   FALSE  42 24  66
##   TRUE    5  5  10
##   Sum    47 29  76
```

With this cutoff, our model catches 5 of the 29 termd (leave) events. This gives us a True Positive Rate of 17% . That is, we are correctly labeling 5 of the 29 "termination" cases. Correspondingly, that same .5 cutoff catches 42 of the 47 stay(not terminated) events, yielding a False Negative Rate of 89% percent.

ROC AUC Curve

'Receiver Operating Characteristic' or ROC curve gives an idea on the performance of a model by measuring the trade off between true-positive rate and false-positive rate. Higher the area under the curve(AUC), better is the model.

The model is evaluated using the Confusion matrix, AUC(Area under the curve), and ROC(Receiver operating characteristics) curve.

First I'm going to install some necessary packages for this part.

```
#install.packages("ROCR")
#install.packages("caTools")

# Loading packages
library(ROCR) # For ROC curve to evaluate model
library(caTools) # For Logistic regression ,calculation of AUC
```

Steps for ROCR

```
pr <- prediction(testpredict, test$Termd)
prf <- performance(pr,measure = "tpr", x.measure = "fpr")
```

The cut point is "optimal" in the sense it weighs both sensitivity and specificity equally. To determine this cutoff, I'm going to us the code below. The code takes in both the performance object and prediction object and gives the optimal cutoff value of predictions: Let's have at a detailed ROC curve:

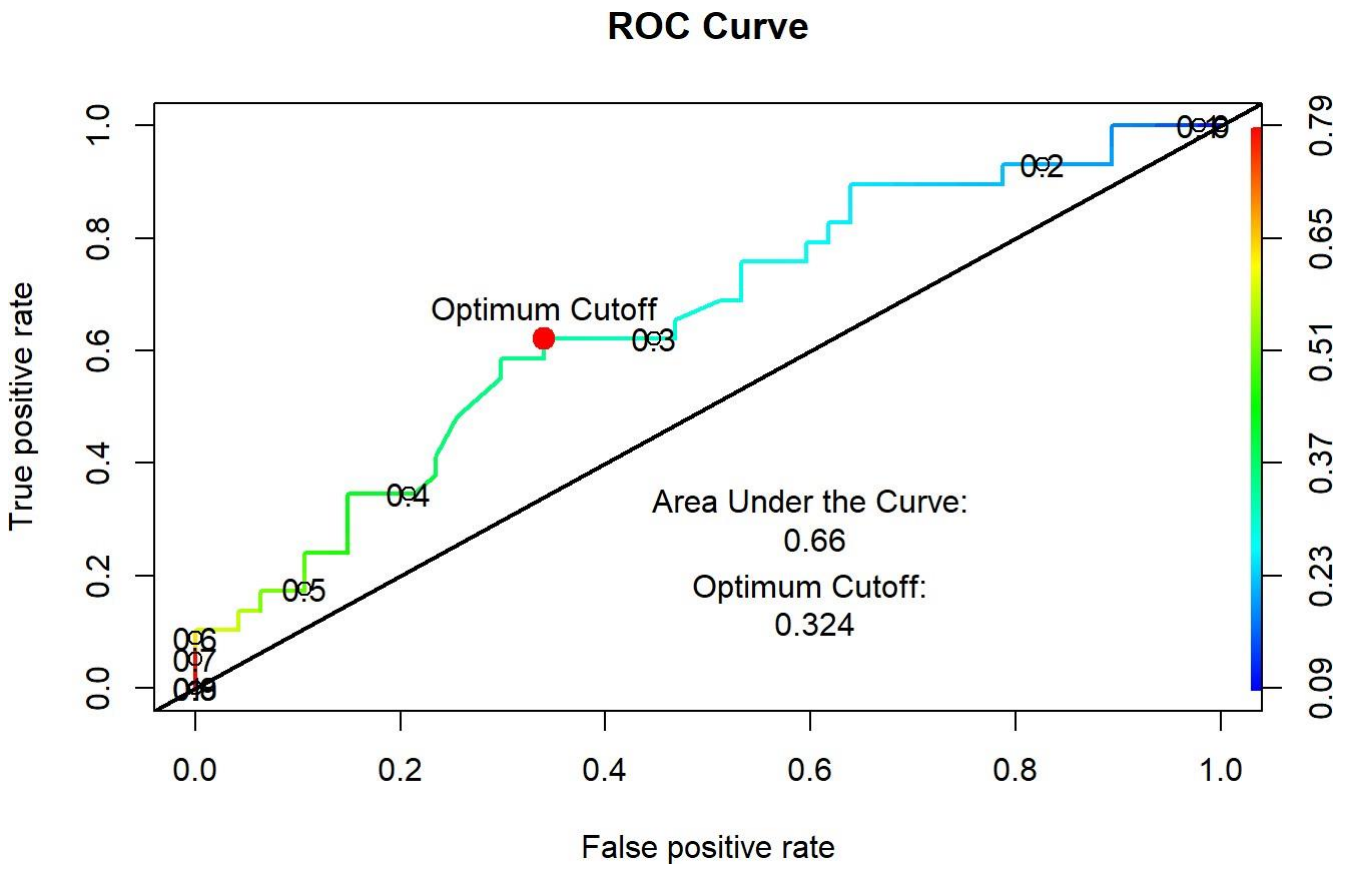
```
# Function to get the best cutoff point
opt.cut <- function(prf, pr){
    cut.ind <- mapply(FUN=function(x, y, p){
        d <- (x - 0)^2 + (y-1)^2
        ind <- which(d == min(d))
        c(sensitivity = y[[ind]], specificity = 1-x[[ind]],
            cutoff = p[[ind]])
    }, prf@x.values, prf@y.values, pr@cutoffs)
}

# the plot
plot(prf, colorize = TRUE, print.cutoffs.at = seq(0,1,.1),
         main = "ROC Curve", lwd = 2)
    abline(coef = c(0,1), col = "black", lwd = 2)

# get the optimum cutoff
opt <- opt.cut(prf, pr)
    points(x = 1-opt[2], y = opt[1], pch = 19, col = "red", cex = 1.5)
    text(x = 1-opt[2],  y = opt[1] + .05, labels = "Optimum Cutoff")

# Area Under the Curve
text(x = .6, y = .3, label = paste("Area Under the Curve:\n",
                                   round(as.numeric(performance(pr, "auc")@y.values), 2)))

    text(x = .6, y = .15, label = paste("Optimum Cutoff:\n",  round(opt[3],3)))
```



True Positive Rate(TPR): True Positive/positive on the y axis and False Positive Rate(FPR): False Positive /Negative on the x axis.

Interpreting the ROC Curve

We can see that the AUC is 0.6555, which is high. This indicates that our model does a good but not excellent job of predicting whether or not an employee is going to terminate(leave). The more AUC is, the better the model performs. We can see values for the whole continuum of predicted probabilities. The color-coded line tells you what the cutoff values are at each point along the curve. This graph definitely proves our model is good because it curves substantially above the diagonal line.

The best possible cutoff value is 0.324

To get the best trade off between False Positives and True Positives, we would categorize everyone below 0.324 as a “0” (stayer) and everyone at or above it as 1 (leaver).

Resources

https://www.r-bloggers.com/2014/12/a-small-introduction-to-the-rocr-package/

http://gim.unmc.edu/dxtests/roc3.htm

https://rcompanion.org/rcompanion/e_07.html

Books:

R and Data Mining: Examples and Case Studies

R Data Mining Blueprints

R Data Mining Blueprints: Learn about data mining with real-world datasets