HR Analytics Using Machine Learning

Wail Hassan

2023-09-23

Project Objectives

The project aim to uncover the factors that lead to employee attrition.build a model that can predict attrition based on certain features of the employee provided in the dataset

Instal The Required Packages and Load Libararies

```
install.packages('rmarkdown',repos = "http://cran.us.r-project.org" )
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'rmarkdown' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
install.packages('ggplot2', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
install.packages('car',repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'car' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
install.packages('carData',repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'carData' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
install.packages('cowplot', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'cowplot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
install.packages('ROCR', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'ROCR' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
install.packages('ROSE', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'ROSE' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
install.packages('rpart.plot', repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/Acc/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)

## package 'rpart.plot' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Acc\AppData\Local\Temp\RtmpIzV91E\downloaded_packages
```

```
library(rmarkdown)
library(plyr)
library(dplyr)
```

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

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

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

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

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(carData)
library(car)
```

```
##
## Attaching package: 'car'

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

```
library(cowplot)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(MASS)
```

```
##
## Attaching package: 'MASS'

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

```r
library(ROCR)
library(ROSE)
```

```
## Loaded ROSE 0.0-4
```

```r
library(rpart)
library(rpart.plot)
```

Collect Data (upload csv files)

```r
getwd()
```

```
## [1] "C:/Users/Acc/Desktop/Meri SKILL Internship/Projects/Project 3 - HR Analytics"
```

```r
setwd("C:\\Users\\Acc\\Desktop\\Meri SKILL Internship\\Projects\\Project 3 - HR Analytics")

general_data <- read.csv('HR-Employee-Attrition.csv', stringsAsFactors = TRUE, header = TRUE, sep = ',')

employee_survey_data <- read.csv('Employee-Survey_data.csv', stringsAsFactors = TRUE,header = TRUE,sep =

manager_survey_data <- read.csv('Manager_Survey-data.csv', stringsAsFactors = TRUE,header = TRUE, sep =

in_time <- read.csv('In-time.csv', stringsAsFactors = TRUE,header = TRUE, sep = ',')

out_time <- read.csv('Out-time.csv', stringsAsFactors = TRUE,header = TRUE, sep = ',')
```

Exploring data frames

```r
str(employee_survey_data)
```

```
## 'data.frame':    1470 obs. of  4 variables:
##  $ EmployeeID             : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ EnvironmentSatisfaction: int  2 3 4 4 1 4 3 4 4 3 ...
##  $ JobSatisfaction        : int  4 2 3 3 2 4 1 3 3 3 ...
##  $ WorkLifeBalance        : int  1 3 3 3 3 2 2 3 3 2 ...
```

```
str(manager_survey_data)
```

```
## 'data.frame':    1470 obs. of  3 variables:
##  $ EmployeeID       : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ JobInvolvement   : int  3 2 2 3 3 3 4 3 2 3 ...
##  $ PerformanceRating: int  3 4 3 3 3 3 4 4 4 3 ...
```

```
str(general_data)
```

```
## 'data.frame':    1470 obs. of  24 variables:
##  $ Age                    : int  41 49 37 33 27 32 59 30 38 36 ...
##  $ Attrition              : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ BusinessTravel         : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 3 2 3 2 3 3 2
##  $ Department             : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ DistanceFromHome       : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ Education              : int  2 1 2 4 1 2 3 1 3 3 ...
##  $ EducationField         : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ EmployeeCount          : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeID             : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender                 : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...
##  $ JobLevel               : int  2 2 1 1 1 1 1 1 3 2 ...
##  $ JobRole                : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1 .
##  $ MaritalStatus          : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...
##  $ MonthlyIncome          : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ NumCompaniesWorked     : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ Over18                 : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ PercentSalaryHike      : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ StandardHours          : int  80 80 80 80 80 80 80 80 80 80 ...
##  $ StockOptionLevel       : int  0 1 0 0 1 0 3 1 0 2 ...
##  $ TotalWorkingYears      : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear  : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ YearsAtCompany         : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsSinceLastPromotion: int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager   : int  5 7 0 0 2 6 0 0 8 7 ...
```

```
str(in_time)
```

```
## 'data.frame':    4410 obs. of  261 variables:
##  $ X01.01.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X02.01.15: Factor w/ 96 levels "","0","02.01.15 10:00",..: 80 18 20 8 31 80 23 94 2 92 ...
##  $ X05.01.15: Factor w/ 92 levels "","0","05.01.15 10:00",..: 11 24 83 89 82 17 63 81 4 24 ...
##  $ X06.01.15: Factor w/ 91 levels "","0","06.01.15 10:00",..: 86 2 17 14 77 11 80 86 82 6 ...
##  $ X07.01.15: Factor w/ 95 levels "","0","07.01.15 10:00",..: 70 81 83 73 85 21 82 88 5 9 ...
##  $ X08.01.15: Factor w/ 95 levels "","0","08.01.15 10:00",..: 87 12 6 5 22 35 95 94 10 88 ...
##  $ X09.01.15: Factor w/ 93 levels "","0","09.01.15 10:00",..: 12 77 8 11 3 22 16 87 93 70 ...
##  $ X12.01.15: Factor w/ 97 levels "","0","12.01.15 10:00",..: 80 3 6 16 32 86 61 96 17 97 ...
##  $ X13.01.15: Factor w/ 98 levels "","0","13.01.15 10:00",..: 16 42 24 92 98 93 3 81 2 11 ...
##  $ X14.01.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X15.01.15: Factor w/ 96 levels "","0","15.01.15 10:00",..: 4 74 92 3 9 36 94 26 89 2 ...
##  $ X16.01.15: Factor w/ 99 levels "","0","16.01.15 10:00",..: 22 97 8 98 6 56 92 73 4 6 ...
##  $ X19.01.15: Factor w/ 99 levels "","0","19.01.15 10:00",..: 2 26 87 83 2 8 78 14 2 92 ...
##  $ X20.01.15: Factor w/ 97 levels "","0","20.01.15 10:00",..: 88 67 95 32 13 93 95 36 85 96 ...
```

```
##  $ X21.01.15: Factor w/ 100 levels "","0","21.01.15 10:00",..: 90 87 32 7 31 27 96 99 63 79 ...
##  $ X22.01.15: Factor w/ 97 levels "","0","22.01.15 10:00",..: 85 6 97 89 13 88 88 90 9 13 ...
##  $ X23.01.15: Factor w/ 93 levels "","0","23.01.15 10:00",..: 57 54 19 90 68 90 83 6 9 90 ...
##  $ X26.01.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X27.01.15: Factor w/ 96 levels "","0","27.01.15 10:00",..: 87 10 10 85 93 2 23 19 4 91 ...
##  $ X28.01.15: Factor w/ 91 levels "","0","28.01.15 10:00",..: 88 11 8 3 77 14 89 72 3 64 ...
##  $ X29.01.15: Factor w/ 93 levels "","0","29.01.15 10:00",..: 87 86 9 83 14 78 69 9 87 57 ...
##  $ X30.01.15: Factor w/ 93 levels "","0","30.01.15 10:00",..: 12 83 17 90 27 3 19 93 73 92 ...
##  $ X02.02.15: Factor w/ 98 levels "","0","02.02.15 10:00",..: 77 10 93 17 82 70 88 71 93 82 ...
##  $ X03.02.15: Factor w/ 95 levels "","0","03.02.15 10:00",..: 26 91 63 4 91 2 83 4 21 23 ...
##  $ X04.02.15: Factor w/ 96 levels "","0","04.02.15 10:00",..: 85 28 7 34 6 95 86 95 2 86 ...
##  $ X05.02.15: Factor w/ 96 levels "","0","05.02.15 10:00",..: 89 8 11 5 5 13 3 83 96 15 ...
##  $ X06.02.15: Factor w/ 95 levels "","0","06.02.15 10:00",..: 89 51 26 60 91 88 91 30 18 13 ...
##  $ X09.02.15: Factor w/ 96 levels "","0","09.02.15 10:00",..: 4 2 11 93 18 10 8 5 13 69 ...
##  $ X10.02.15: Factor w/ 92 levels "","0","10.02.15 10:00",..: 17 13 29 2 76 74 86 9 10 75 ...
##  $ X11.02.15: Factor w/ 95 levels "","0","11.02.15 10:00",..: 21 76 5 3 6 4 90 15 65 32 ...
##  $ X12.02.15: Factor w/ 95 levels "","0","12.02.15 10:00",..: 13 13 91 4 5 6 69 11 42 86 ...
##  $ X13.02.15: Factor w/ 94 levels "","0","13.02.15 10:00",..: 50 70 35 17 79 12 14 86 61 10 ...
##  $ X16.02.15: Factor w/ 96 levels "","0","16.02.15 10:00",..: 17 7 21 87 92 8 14 82 8 89 ...
##  $ X17.02.15: Factor w/ 95 levels "","0","17.02.15 10:00",..: 6 81 7 88 12 24 7 89 3 7 ...
##  $ X18.02.15: Factor w/ 92 levels "","0","18.02.15 10:00",..: 26 87 19 72 70 70 88 84 2 81 ...
##  $ X19.02.15: Factor w/ 100 levels "","0","19.02.15 10:00",..: 2 14 88 84 79 11 85 8 93 25 ...
##  $ X20.02.15: Factor w/ 97 levels "","0","20.02.15 10:00",..: 92 9 23 4 5 86 78 4 89 10 ...
##  $ X23.02.15: Factor w/ 95 levels "","0","23.02.15 10:00",..: 82 77 17 69 28 90 84 84 90 7 ...
##  $ X24.02.15: Factor w/ 91 levels "","0","24.02.15 10:00",..: 14 76 26 30 83 25 74 87 16 2 ...
##  $ X25.02.15: Factor w/ 96 levels "","0","25.02.15 10:00",..: 89 95 96 82 92 67 95 79 20 94 ...
##  $ X26.02.15: Factor w/ 99 levels "","0","26.02.15 10:00",..: 96 99 4 88 98 30 2 90 14 11 ...
##  $ X27.02.15: Factor w/ 96 levels "","0","27.02.15 10:00",..: 95 34 94 80 10 93 26 18 22 80 ...
##  $ X02.03.15: Factor w/ 97 levels "","0","02.03.15 10:00",..: 22 93 5 2 97 72 74 94 86 32 ...
##  $ X03.03.15: Factor w/ 99 levels "","0","03.03.15 10:00",..: 11 34 10 18 82 84 15 90 14 97 ...
##  $ X04.03.15: Factor w/ 98 levels "","0","04.03.15 10:00",..: 8 95 18 98 3 14 4 84 18 29 ...
##  $ X05.03.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X06.03.15: Factor w/ 93 levels "","0","06.03.15 10:00",..: 89 4 82 2 87 84 8 18 15 80 ...
##  $ X09.03.15: Factor w/ 92 levels "","0","09.03.15 10:00",..: 31 5 20 8 89 6 84 6 24 24 ...
##  $ X10.03.15: Factor w/ 96 levels "","0","10.03.15 10:00",..: 84 69 83 66 5 6 3 14 70 91 ...
##  $ X11.03.15: Factor w/ 95 levels "","0","11.03.15 10:00",..: 84 93 29 93 84 81 19 8 10 76 ...
##  $ X12.03.15: Factor w/ 97 levels "","0","12.03.15 10:00",..: 13 97 94 79 72 82 91 16 78 96 ...
##  $ X13.03.15: Factor w/ 97 levels "","0","13.03.15 10:00",..: 10 97 74 90 15 92 33 17 78 81 ...
##  $ X16.03.15: Factor w/ 92 levels "","0","16.03.15 10:00",..: 2 89 10 89 92 86 6 4 73 16 ...
##  $ X17.03.15: Factor w/ 96 levels "","0","17.03.15 10:00",..: 8 73 3 89 19 9 15 88 12 73 ...
##  $ X18.03.15: Factor w/ 93 levels "","0","18.03.15 10:00",..: 31 3 23 28 83 88 70 13 20 66 ...
##  $ X19.03.15: Factor w/ 97 levels "","0","19.03.15 10:00",..: 4 84 92 29 9 74 23 90 19 72 ...
##  $ X20.03.15: Factor w/ 97 levels "","0","20.03.15 10:00",..: 40 9 2 25 84 83 33 18 13 81 ...
##  $ X23.03.15: Factor w/ 92 levels "","0","23.03.15 10:00",..: 14 5 7 24 15 60 87 31 86 77 ...
##  $ X24.03.15: Factor w/ 94 levels "","0","24.03.15 10:00",..: 36 2 21 29 72 67 81 76 2 13 ...
##  $ X25.03.15: Factor w/ 100 levels "","0","25.03.15 10:00",..: 82 6 84 3 19 20 83 14 14 10 ...
##  $ X26.03.15: Factor w/ 93 levels "","0","26.03.15 10:00",..: 2 86 71 24 9 35 84 59 79 88 ...
##  $ X27.03.15: Factor w/ 96 levels "","0","27.03.15 10:00",..: 84 5 2 2 84 94 76 20 93 12 ...
##  $ X30.03.15: Factor w/ 94 levels "","0","30.03.15 10:00",..: 14 7 89 14 94 81 86 2 16 23 ...
##  $ X31.03.15: Factor w/ 100 levels "","0","31.03.15 10:00",..: 11 33 3 100 31 12 71 74 10 24 ...
##  $ X01.04.15: Factor w/ 96 levels "","0","01.04.15 10:00",..: 15 83 89 85 83 11 66 81 8 12 ...
##  $ X02.04.15: Factor w/ 97 levels "","0","02.04.15 10:00",..: 83 91 27 75 92 84 89 12 40 93 ...
##  $ X03.04.15: Factor w/ 88 levels "","0","03.04.15 10:00",..: 3 33 4 84 80 3 18 19 4 23 ...
##  $ X06.04.15: Factor w/ 97 levels "","0","06.04.15 10:00",..: 78 21 71 97 8 92 10 23 6 88 ...
```

```
##  $ X07.04.15: Factor w/ 94 levels "","0","07.04.15 10:00",..: 19 25 16 38 84 92 86 21 93 5 ...
##  $ X08.04.15: Factor w/ 97 levels "","0","08.04.15 10:00",..: 7 84 40 29 11 77 95 89 92 13 ...
##  $ X09.04.15: Factor w/ 98 levels "","0","09.04.15 10:00",..: 60 5 20 15 96 38 77 23 24 90 ...
##  $ X10.04.15: Factor w/ 97 levels "","0","10.04.15 10:00",..: 84 85 31 25 12 10 80 90 37 22 ...
##  $ X13.04.15: Factor w/ 96 levels "","0","13.04.15 10:00",..: 76 70 17 68 96 66 40 33 18 76 ...
##  $ X14.04.15: Factor w/ 95 levels "","0","14.04.15 10:00",..: 94 86 65 2 94 83 19 16 10 93 ...
##  $ X15.04.15: Factor w/ 102 levels "","0","15.04.15 10:00",..: 98 9 9 15 101 92 33 41 19 79 ...
##  $ X16.04.15: Factor w/ 95 levels "","0","16.04.15 10:00",..: 82 10 82 81 85 10 93 77 91 20 ...
##  $ X17.04.15: Factor w/ 98 levels "","0","17.04.15 10:00",..: 93 79 19 7 5 91 92 10 26 23 ...
##  $ X20.04.15: Factor w/ 96 levels "","0","20.04.15 10:00",..: 6 7 73 88 92 4 2 2 79 3 ...
##  $ X21.04.15: Factor w/ 101 levels "","0","21.04.15 10:00",..: 92 24 10 16 2 91 3 16 61 13 ...
##  $ X22.04.15: Factor w/ 92 levels "","0","22.04.15 10:00",..: 89 73 89 83 12 8 51 8 91 69 ...
##  $ X23.04.15: Factor w/ 97 levels "","0","23.04.15 10:00",..: 15 9 2 75 3 3 6 90 11 94 ...
##  $ X24.04.15: Factor w/ 97 levels "","0","24.04.15 10:00",..: 8 13 13 96 21 75 9 78 92 14 ...
##  $ X27.04.15: Factor w/ 101 levels "","0","27.04.15 10:00",..: 15 75 71 95 21 92 85 4 17 27 ...
##  $ X28.04.15: Factor w/ 91 levels "","0","28.04.15 10:00",..: 89 84 4 56 73 89 86 91 85 89 ...
##  $ X29.04.15: Factor w/ 99 levels "","0","29.04.15 10:00",..: 86 2 94 95 24 15 87 98 10 88 ...
##  $ X30.04.15: Factor w/ 93 levels "","0","30.04.15 10:00",..: 82 17 74 5 23 88 65 26 4 2 ...
##  $ X01.05.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X04.05.15: Factor w/ 97 levels "","0","04.05.15 10:00",..: 78 91 93 31 79 78 20 11 36 23 ...
##  $ X05.05.15: Factor w/ 93 levels "","0","05.05.15 10:00",..: 88 28 87 3 12 15 6 11 15 18 ...
##  $ X06.05.15: Factor w/ 95 levels "","0","06.05.15 10:00",..: 85 82 23 21 31 62 83 25 65 89 ...
##  $ X07.05.15: Factor w/ 94 levels "","0","07.05.15 10:00",..: 9 84 84 6 8 78 79 76 92 85 ...
##  $ X08.05.15: Factor w/ 94 levels "","0","08.05.15 10:00",..: 90 91 24 3 23 16 5 85 94 2 ...
##  $ X11.05.15: Factor w/ 106 levels "","0","11.05.15 10:00",..: 2 105 94 26 90 2 2 100 37 94 ...
##  $ X12.05.15: Factor w/ 96 levels "","0","12.05.15 10:00",..: 87 81 76 73 83 3 4 10 96 86 ...
##  $ X13.05.15: Factor w/ 99 levels "","0","13.05.15 10:00",..: 27 9 90 16 12 6 34 65 31 97 ...
##  $ X14.05.15: Factor w/ 97 levels "","0","14.05.15 10:00",..: 88 2 83 83 10 43 9 76 16 4 ...
##  $ X15.05.15: Factor w/ 96 levels "","0","15.05.15 10:00",..: 92 80 85 64 64 32 22 2 81 72 ...
##  $ X18.05.15: Factor w/ 95 levels "","0","18.05.15 10:00",..: 2 2 6 65 11 83 94 95 91 81 ...
##  $ X19.05.15: Factor w/ 96 levels "","0","19.05.15 10:00",..: 86 85 77 89 78 72 14 66 73 79 ...
##   [list output truncated]
```

```
str(out_time)
```

```
## 'data.frame':    4410 obs. of  261 variables:
##  $ X01.01.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X02.01.15: Factor w/ 341 levels "","0","02.01.15 15:02",..: 96 182 99 125 190 296 110 102 2 117 .
##  $ X05.01.15: Factor w/ 339 levels "","0","05.01.15 14:38",..: 118 146 104 112 147 322 100 90 98 125
##  $ X06.01.15: Factor w/ 333 levels "","0","06.01.15 15:04",..: 120 2 79 108 127 322 105 74 111 134 .
##  $ X07.01.15: Factor w/ 332 levels "","0","07.01.15 14:53",..: 75 110 74 73 138 301 52 83 137 116 ..
##  $ X08.01.15: Factor w/ 343 levels "","0","08.01.15 14:51",..: 112 138 128 97 163 342 105 62 134 86
##  $ X09.01.15: Factor w/ 333 levels "","0","09.01.15 14:42",..: 134 88 93 115 140 331 119 85 115 73 .
##  $ X12.01.15: Factor w/ 336 levels "","0","12.01.15 14:56",..: 98 136 128 113 211 303 44 81 114 154
##  $ X13.01.15: Factor w/ 340 levels "","0","13.01.15 14:58",..: 163 161 122 112 175 282 96 67 2 122 .
##  $ X14.01.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X15.01.15: Factor w/ 326 levels "","0","15.01.15 14:45",..: 118 110 117 89 177 318 116 105 123 2
##  $ X16.01.15: Factor w/ 331 levels "","0","16.01.15 15:08",..: 132 137 115 89 184 270 116 51 88 104
##  $ X19.01.15: Factor w/ 333 levels "","0","19.01.15 14:52",..: 2 157 91 53 2 330 63 85 2 107 ...
##  $ X20.01.15: Factor w/ 339 levels "","0","20.01.15 14:57",..: 76 104 91 138 158 291 80 112 111 90 .
##  $ X21.01.15: Factor w/ 339 levels "","0","21.01.15 15:12",..: 94 140 116 85 199 334 88 48 83 82 ...
##  $ X22.01.15: Factor w/ 332 levels "","0","22.01.15 14:39",..: 88 124 125 89 184 317 88 78 125 143 .
##  $ X23.01.15: Factor w/ 333 levels "","0","23.01.15 15:09",..: 98 104 109 103 91 307 89 93 153 110 .
##  $ X26.01.15: int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ X27.01.15: Factor w/ 337 levels "","0","27.01.15 14:45",..: 125 132 119 100 149 2 125 128 141 104
##  $ X28.01.15: Factor w/ 337 levels "","0","28.01.15 15:07",..: 101 117 62 131 138 321 109 25 96 79 .
##  $ X29.01.15: Factor w/ 348 levels "","0","29.01.15 14:58",..: 122 111 124 88 180 319 89 82 115 64 .
##  $ X30.01.15: Factor w/ 333 levels "","0","30.01.15 15:00",..: 100 93 116 109 186 296 143 69 92 137
##  $ X02.02.15: Factor w/ 341 levels "","0","02.02.15 14:59",..: 72 145 99 118 153 272 81 42 101 95 ..
##  $ X03.02.15: Factor w/ 339 levels "","0","03.02.15 14:48",..: 135 133 38 127 157 2 111 87 124 129 .
##  $ X04.02.15: Factor w/ 343 levels "","0","04.02.15 15:03",..: 125 165 126 131 164 315 106 82 2 76 .
##  $ X05.02.15: Factor w/ 341 levels "","0","05.02.15 14:49",..: 129 146 72 114 170 332 74 57 114 106
##  $ X06.02.15: Factor w/ 344 levels "","0","06.02.15 15:00",..: 122 107 119 74 139 310 85 106 110 103
##  $ X09.02.15: Factor w/ 336 levels "","0","09.02.15 14:57",..: 136 2 87 115 185 317 70 102 131 81 ..
##  $ X10.02.15: Factor w/ 337 levels "","0","10.02.15 15:12",..: 130 172 115 2 124 319 109 84 104 75 .
##  $ X11.02.15: Factor w/ 328 levels "","0","11.02.15 15:13",..: 125 113 84 99 156 322 56 94 94 130 ..
##  $ X12.02.15: Factor w/ 330 levels "","0","12.02.15 15:09",..: 134 166 60 120 140 290 54 124 167 115
##  $ X13.02.15: Factor w/ 344 levels "","0","13.02.15 15:01",..: 70 124 146 127 193 327 134 71 95 107
##  $ X16.02.15: Factor w/ 340 levels "","0","16.02.15 15:10",..: 147 143 117 108 148 323 99 63 145 98
##  $ X17.02.15: Factor w/ 342 levels "","0","17.02.15 15:00",..: 129 110 87 100 164 341 69 94 139 120
##  $ X18.02.15: Factor w/ 336 levels "","0","18.02.15 14:44",..: 174 149 120 72 144 289 68 83 2 119 ..
##  $ X19.02.15: Factor w/ 343 levels "","0","19.02.15 15:02",..: 2 167 85 116 130 316 120 76 140 145 .
##  $ X20.02.15: Factor w/ 339 levels "","0","20.02.15 15:12",..: 145 132 116 85 177 298 58 104 75 116
##  $ X23.02.15: Factor w/ 347 levels "","0","23.02.15 14:56",..: 115 159 161 86 175 306 67 60 127 109
##  $ X24.02.15: Factor w/ 341 levels "","0","24.02.15 15:08",..: 124 132 108 130 157 337 63 78 100 2 .
##  $ X25.02.15: Factor w/ 337 levels "","0","25.02.15 15:08",..: 128 144 97 44 158 294 102 80 141 71 .
##  $ X26.02.15: Factor w/ 342 levels "","0","26.02.15 15:06",..: 112 153 138 113 144 339 2 45 146 124
##  $ X27.02.15: Factor w/ 335 levels "","0","27.02.15 14:52",..: 123 199 90 101 167 301 104 125 127 96
##  $ X02.03.15: Factor w/ 344 levels "","0","02.03.15 14:51",..: 134 141 137 2 206 271 80 97 119 109 .
##  $ X03.03.15: Factor w/ 338 levels "","0","03.03.15 14:46",..: 98 176 115 143 143 325 97 44 134 91 .
##  $ X04.03.15: Factor w/ 338 levels "","0","04.03.15 15:00",..: 154 156 130 70 168 329 83 85 127 133
##  $ X05.03.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X06.03.15: Factor w/ 341 levels "","0","06.03.15 14:49",..: 146 123 77 2 166 293 106 105 104 87 .
##  $ X09.03.15: Factor w/ 335 levels "","0","09.03.15 14:39",..: 153 169 108 115 177 310 80 118 163 130
##  $ X10.03.15: Factor w/ 338 levels "","0","10.03.15 15:13",..: 79 111 75 90 163 293 132 58 105 88 ..
##  $ X11.03.15: Factor w/ 335 levels "","0","11.03.15 14:56",..: 102 160 135 131 160 302 113 110 123 93
##  $ X12.03.15: Factor w/ 341 levels "","0","12.03.15 15:05",..: 115 130 113 118 139 299 69 87 107 104
##  $ X13.03.15: Factor w/ 337 levels "","0","13.03.15 14:53",..: 126 149 66 83 176 277 127 66 119 122
##  $ X16.03.15: Factor w/ 339 levels "","0","16.03.15 15:05",..: 2 109 94 80 147 325 60 93 85 131 ...
##  $ X17.03.15: Factor w/ 338 levels "","0","17.03.15 14:58",..: 102 100 109 106 175 318 114 98 138 74
##  $ X18.03.15: Factor w/ 340 levels "","0","18.03.15 14:41",..: 143 119 142 134 158 299 60 119 130 110
##  $ X19.03.15: Factor w/ 339 levels "","0","19.03.15 14:53",..: 153 129 68 128 152 276 121 39 118 80
##  $ X20.03.15: Factor w/ 334 levels "","0","20.03.15 14:37",..: 149 133 2 137 134 306 122 91 79 93 ..
##  $ X23.03.15: Factor w/ 336 levels "","0","23.03.15 14:55",..: 150 160 92 120 155 283 105 134 110 111
##  $ X24.03.15: Factor w/ 332 levels "","0","24.03.15 15:02",..: 150 2 114 124 120 289 88 35 2 140 ...
##  $ X25.03.15: Factor w/ 338 levels "","0","25.03.15 15:05",..: 107 121 81 119 166 329 55 83 149 106
##  $ X26.03.15: Factor w/ 338 levels "","0","26.03.15 14:57",..: 2 150 104 172 155 335 76 57 81 108 ..
##  $ X27.03.15: Factor w/ 333 levels "","0","27.03.15 14:59",..: 120 149 2 2 141 311 78 123 112 120 ..
##  $ X30.03.15: Factor w/ 336 levels "","0","30.03.15 14:37",..: 133 128 77 113 141 309 85 2 140 141 .
##  $ X31.03.15: Factor w/ 350 levels "","0","31.03.15 14:51",..: 165 193 134 140 181 349 77 68 125 166
##  $ X01.04.15: Factor w/ 341 levels "","0","01.04.15 14:47",..: 134 132 89 121 150 324 56 52 121 144
##  $ X02.04.15: Factor w/ 333 levels "","0","02.04.15 14:58",..: 126 141 116 74 162 262 111 91 123 90
##  $ X03.04.15: Factor w/ 341 levels "","0","03.04.15 14:58",..: 112 168 87 100 142 326 97 108 96 133
##  $ X06.04.15: Factor w/ 329 levels "","0","06.04.15 14:46",..: 81 139 66 122 183 309 132 75 128 89 .
##  $ X07.04.15: Factor w/ 339 levels "","0","07.04.15 14:50",..: 153 148 125 137 149 305 82 124 102 107
##  $ X08.04.15: Factor w/ 328 levels "","0","08.04.15 15:04",..: 129 108 135 138 142 282 65 107 75 99
##  $ X09.04.15: Factor w/ 348 levels "","0","09.04.15 14:43",..: 114 112 150 120 171 348 94 120 111 96
##  $ X10.04.15: Factor w/ 344 levels "","0","10.04.15 14:51",..: 93 143 113 121 166 343 86 90 143 159
```

```
##  $ X13.04.15: Factor w/ 335 levels "","0","13.04.15 15:00",..: 101 146 131 63 152 319 155 116 129 77
##  $ X14.04.15: Factor w/ 340 levels "","0","14.04.15 15:04",..: 111 164 87 2 193 309 82 103 131 123 .
##  $ X15.04.15: Factor w/ 337 levels "","0","15.04.15 14:54",..: 124 136 124 83 163 325 148 97 132 57
##  $ X16.04.15: Factor w/ 339 levels "","0","16.04.15 14:53",..: 132 122 90 91 160 306 111 55 100 115
##  $ X17.04.15: Factor w/ 330 levels "","0","17.04.15 15:04",..: 108 121 149 100 152 315 114 75 147 13
##  $ X20.04.15: Factor w/ 336 levels "","0","20.04.15 14:46",..: 145 151 103 110 174 324 2 2 107 100 .
##  $ X21.04.15: Factor w/ 342 levels "","0","21.04.15 15:01",..: 89 206 115 118 2 316 109 72 71 137 ..
##  $ X22.04.15: Factor w/ 338 levels "","0","22.04.15 15:05",..: 127 140 123 106 138 323 39 79 138 66
##  $ X23.04.15: Factor w/ 336 levels "","0","23.04.15 15:04",..: 132 164 2 115 150 303 132 61 130 106
##  $ X24.04.15: Factor w/ 339 levels "","0","24.04.15 15:11",..: 141 144 107 126 170 293 104 63 112 11
##  $ X27.04.15: Factor w/ 331 levels "","0","27.04.15 15:00",..: 126 112 69 116 215 301 72 102 120 121
##  $ X28.04.15: Factor w/ 338 levels "","0","28.04.15 15:04",..: 152 131 124 61 112 301 123 74 108 99
##  $ X29.04.15: Factor w/ 342 levels "","0","29.04.15 14:46",..: 125 2 90 102 158 282 54 45 117 92 ...
##  $ X30.04.15: Factor w/ 341 levels "","0","30.04.15 14:43",..: 119 179 84 110 168 316 42 131 144 2 .
##  $ X01.05.15: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ X04.05.15: Factor w/ 338 levels "","0","04.05.15 14:58",..: 108 144 85 162 139 251 124 77 142 112
##  $ X05.05.15: Factor w/ 342 levels "","0","05.05.15 15:01",..: 112 167 52 83 188 341 81 93 138 140 .
##  $ X06.05.15: Factor w/ 331 levels "","0","06.05.15 15:14",..: 113 134 124 165 214 268 87 102 60 115
##  $ X07.05.15: Factor w/ 334 levels "","0","07.05.15 14:56",..: 110 126 81 113 127 278 59 85 122 106
##  $ X08.05.15: Factor w/ 337 levels "","0","08.05.15 15:01",..: 131 107 113 143 160 286 80 82 101 2 .
##  $ X11.05.15: Factor w/ 347 levels "","0","11.05.15 14:23",..: 2 132 91 133 132 2 2 56 167 82 ...
##  $ X12.05.15: Factor w/ 337 levels "","0","12.05.15 14:47",..: 116 139 85 93 184 333 71 80 99 109 ..
##  $ X13.05.15: Factor w/ 341 levels "","0","13.05.15 14:41",..: 155 136 81 124 139 314 107 73 137 85
##  $ X14.05.15: Factor w/ 348 levels "","0","14.05.15 14:51",..: 153 2 110 110 174 346 128 62 172 107
##  $ X15.05.15: Factor w/ 338 levels "","0","15.05.15 14:55",..: 90 112 71 82 121 324 99 2 120 80 ...
##  $ X18.05.15: Factor w/ 340 levels "","0","18.05.15 14:43",..: 2 2 121 66 143 293 125 95 59 86 ...
##  $ X19.05.15: Factor w/ 342 levels "","0","19.05.15 14:55",..: 111 155 107 125 162 306 118 61 130 10
##   [list output truncated]
```

Clean Up and Prepare for the Analysis

```
sapply(employee_survey_data, function(x) sum(is.na(x)))/nrow(employee_survey_data)*100 #checking for mi
```

```
##              EmployeeID EnvironmentSatisfaction         JobSatisfaction
##                       0                       0                       0
##         WorkLifeBalance
##                       0
```

```
sapply(manager_survey_data, function(x) sum(is.na(x)))/nrow(manager_survey_data)*100 #checking for miss
```

```
##        EmployeeID    JobInvolvement PerformanceRating
##                 0                 0                 0
```

```
sapply(general_data, function(x) sum(is.na(x)))/nrow(general_data)*100 #checking for missing values
```

```
##                    Age              Attrition          BusinessTravel
##                      0                      0                       0
##             Department       DistanceFromHome              Education
##                      0                      0                       0
##          EducationField          EmployeeCount             EmployeeID
```

```
##                      0                      0                      0
##                 Gender               JobLevel                JobRole
##                      0                      0                      0
##          MaritalStatus          MonthlyIncome     NumCompaniesWorked
##                      0                      0                      0
##                 Over18      PercentSalaryHike          StandardHours
##                      0                      0                      0
##       StockOptionLevel      TotalWorkingYears  TrainingTimesLastYear
##                      0                      0                      0
##         YearsAtCompany YearsSinceLastPromotion   YearsWithCurrManager
##                      0                      0                      0
```

```
# Before merging we take a look if each observation is from a different employee

setdiff(employee_survey_data$EmployeeID, manager_survey_data$EmployeeID)
```

```
## integer(0)
```

```
setdiff(manager_survey_data$EmployeeID, general_data$EmployeeID)
```

```
## integer(0)
```

```
# Since all of them are complete we can merge them

general_data_merged <- inner_join(general_data,manager_survey_data, by = 'EmployeeID') %>%
  inner_join(., employee_survey_data, by = 'EmployeeID')
```

```
# Droping values that have the same value for all observations

same_values <- nearZeroVar(general_data_merged, names = TRUE)
general_data_merged <- general_data_merged %>%
  dplyr::select(-c(c('EmployeeID', same_values)))
```

```
## Warning: Using an external vector in selections was deprecated in tidyselect 1.1.0.
## i Please use ‘all_of()‘ or ‘any_of()‘ instead.
##   # Was:
##   data %>% select(same_values)
##
##   # Now:
##   data %>% select(all_of(same_values))
##
## See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Now we check the structure of the new dataframe and the missing values. And evaluate if they are sign

str(general_data_merged)
```

```
## 'data.frame':    1470 obs. of  25 variables:
##  $ Age                     : int  41 49 37 33 27 32 59 30 38 36 ...
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 3 2 3 2 3 3 2
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ DistanceFromHome        : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ Education               : int  2 1 2 4 1 2 3 1 3 3 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...
##  $ JobLevel                : int  2 2 1 1 1 1 1 1 3 2 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1 .
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...
##  $ MonthlyIncome           : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ NumCompaniesWorked      : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ PercentSalaryHike       : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ StockOptionLevel        : int  0 1 0 0 1 0 3 1 0 2 ...
##  $ TotalWorkingYears       : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear   : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ YearsAtCompany          : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsSinceLastPromotion: int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager    : int  5 7 0 0 2 6 0 0 8 7 ...
##  $ JobInvolvement          : int  3 2 2 3 3 3 4 3 2 3 ...
##  $ PerformanceRating       : int  3 4 3 3 3 3 4 4 4 3 ...
##  $ EnvironmentSatisfaction: int  2 3 4 4 1 4 3 4 4 3 ...
##  $ JobSatisfaction         : int  4 2 3 3 2 4 1 3 3 3 ...
##  $ WorkLifeBalance         : int  1 3 3 3 3 2 2 3 3 2 ...
```

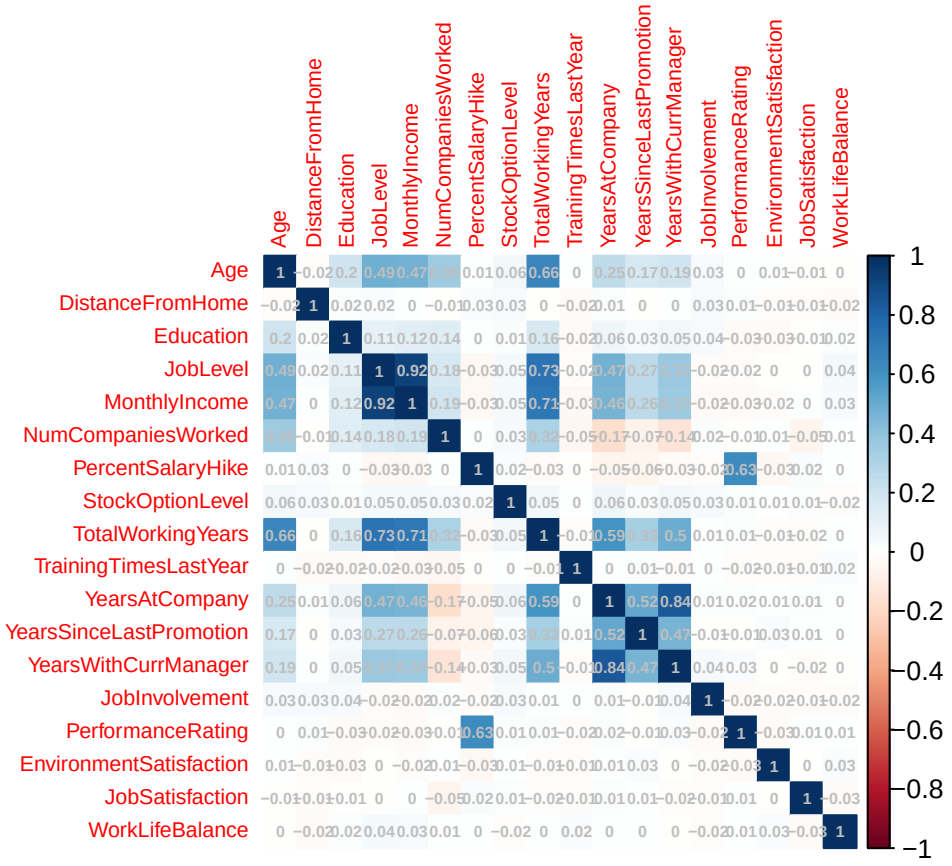
```
sapply(general_data_merged, function(x) sum(is.na(x)))/nrow(general_data_merged)*100
```

```
##                      Age                Attrition           BusinessTravel
##                        0                        0                        0
##               Department         DistanceFromHome                Education
##                        0                        0                        0
##          EducationField                   Gender                 JobLevel
##                        0                        0                        0
##                  JobRole            MaritalStatus            MonthlyIncome
##                        0                        0                        0
##      NumCompaniesWorked        PercentSalaryHike         StockOptionLevel
##                        0                        0                        0
##       TotalWorkingYears    TrainingTimesLastYear           YearsAtCompany
##                        0                        0                        0
## YearsSinceLastPromotion     YearsWithCurrManager           JobInvolvement
##                        0                        0                        0
##       PerformanceRating EnvironmentSatisfaction          JobSatisfaction
##                        0                        0                        0
##         WorkLifeBalance
##                        0
```

Processing and Analyzing

Plot The Correlation Between The Data Frames

```
# First we keep the numeric variables

nums <- unlist(lapply(general_data_merged, is.numeric))
general_data_merged_with_ordinal_values <- general_data_merged[, nums]

# With this data we create a correlation matrix in order to see the relantionships between the features

cor_matrix <- cor(general_data_merged_with_ordinal_values, method = 'spearman')
corrplot(corr = cor_matrix, method = 'color', addCoef.col = 'gray',tl.cex = 0.7, number.cex = 0.5)
```




```
# We create the second dataframe with the labels of the ordinal features

general_data_merged_with_categories <- general_data_merged

general_data_merged_with_categories$Education[which(general_data_merged_with_categories$Education == 1)]
general_data_merged_with_categories$Education[which(general_data_merged_with_categories$Education == 2)]
general_data_merged_with_categories$Education[which(general_data_merged_with_categories$Education == 3)]
general_data_merged_with_categories$Education[which(general_data_merged_with_categories$Education == 4)]
general_data_merged_with_categories$Education[which(general_data_merged_with_categories$Education == 5)]
```

```
general_data_merged_with_categories$EnvironmentSatisfaction[which(general_data_merged_with_categories$En
general_data_merged_with_categories$EnvironmentSatisfaction[which(general_data_merged_with_categories$En
general_data_merged_with_categories$EnvironmentSatisfaction[which(general_data_merged_with_categories$En
general_data_merged_with_categories$EnvironmentSatisfaction[which(general_data_merged_with_categories$En

general_data_merged_with_categories$JobInvolvement[which(general_data_merged_with_categories$JobInvolver
general_data_merged_with_categories$JobInvolvement[which(general_data_merged_with_categories$JobInvolver
general_data_merged_with_categories$JobInvolvement[which(general_data_merged_with_categories$JobInvolver
general_data_merged_with_categories$JobInvolvement[which(general_data_merged_with_categories$JobInvolver

general_data_merged_with_categories$JobSatisfaction[which(general_data_merged_with_categories$JobSatisfa
general_data_merged_with_categories$JobSatisfaction[which(general_data_merged_with_categories$JobSatisfa
general_data_merged_with_categories$JobSatisfaction[which(general_data_merged_with_categories$JobSatisfa
general_data_merged_with_categories$JobSatisfaction[which(general_data_merged_with_categories$JobSatisfa

general_data_merged_with_categories$PerformanceRating[which(general_data_merged_with_categories$Performa
general_data_merged_with_categories$PerformanceRating[which(general_data_merged_with_categories$Performa
general_data_merged_with_categories$PerformanceRating[which(general_data_merged_with_categories$Performa
general_data_merged_with_categories$PerformanceRating[which(general_data_merged_with_categories$Performa

general_data_merged_with_categories$WorkLifeBalance[which(general_data_merged_with_categories$WorkLifeBa
general_data_merged_with_categories$WorkLifeBalance[which(general_data_merged_with_categories$WorkLifeBa
general_data_merged_with_categories$WorkLifeBalance[which(general_data_merged_with_categories$WorkLifeBa
general_data_merged_with_categories$WorkLifeBalance[which(general_data_merged_with_categories$WorkLifeBa
```

Pivot Tables

```
general_data_merged_with_categories %>%
  group_by(Gender, Education) %>%
  summarize(Average_Income = mean(MonthlyIncome)) %>%
  arrange(-Average_Income)
```

```
## `summarise()` has grouped output by 'Gender'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 10 x 3
## # Groups:   Gender [2]
##    Gender Education     Average_Income
##    <fct>  <chr>                  <dbl>
##  1 Female Doctor                 9241.
##  2 Male   Doctor                 7463.
##  3 Female Master                 6878.
##  4 Female Bachelor               6811.
##  5 Male   Master                 6804.
##  6 Male   Bachelor               6313.
##  7 Male   College                6267.
##  8 Female College                6169.
##  9 Female Below College          5781.
## 10 Male   Below College          5564.
```

```
general_data_merged_with_categories %>%
  group_by(MaritalStatus, WorkLifeBalance) %>%
  summarize( Percent_employees = round(n()/nrow(general_data_merged_with_categories)*100,0)) %>%
  arrange(-Percent_employees)
```

```
## `summarise()` has grouped output by 'MaritalStatus'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 12 x 3
## # Groups:   MaritalStatus [3]
##    MaritalStatus WorkLifeBalance Percent_employees
##    <fct>         <chr>                       <dbl>
##  1 Married       Better                         28
##  2 Single        Better                         20
##  3 Divorced      Better                         13
##  4 Married       Good                           10
##  5 Single        Good                            7
##  6 Divorced      Good                            6
##  7 Married       Best                            5
##  8 Married       Bad                             3
##  9 Single        Best                            3
## 10 Divorced      Best                            2
## 11 Single        Bad                             2
## 12 Divorced      Bad                             1
```

Findings:

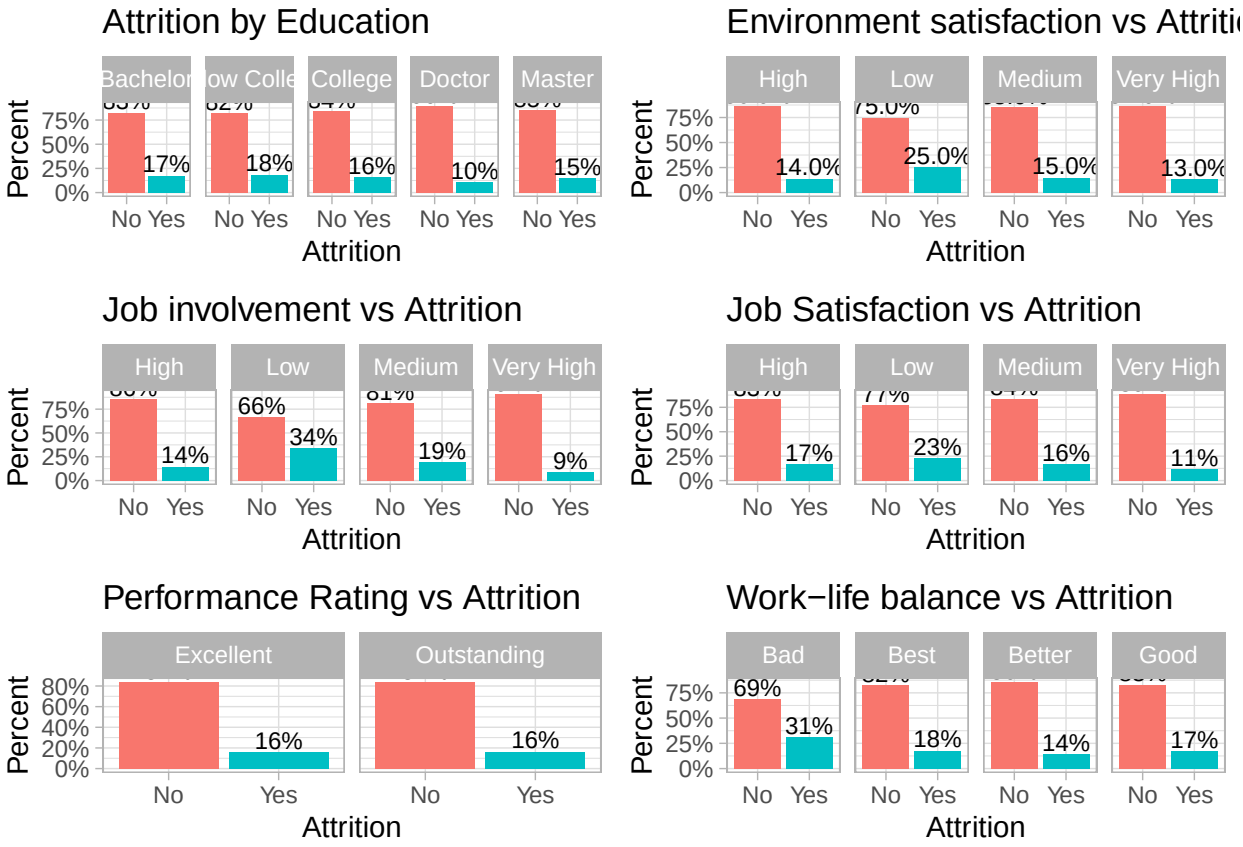
*It seems that female workers have a high income in comparison with men, except for the women who have a masters degree since they earn less on average than the men.

*More than a half of the employees say they have a 'Better' work-life balance. ## More plots to see the rate of attrition by other variables such as Education, Environment Satisfaction, Job Involvement, Job satisfaction, Performance rating, work balance

```
g1 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = Education))
  geom_bar(aes(y = after_stat(prop), fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~Education) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.1
  labs(y = 'Percent', x = 'Attrition', title = 'Attrition by Education', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

g2 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = EnvironmentS
  geom_bar(aes(y = after_stat(prop), fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~EnvironmentSatisfaction) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.1
  labs(y = 'Percent', x = 'Attrition', title = 'Environment satisfaction vs Attrition', fill = 'Attritio
  theme_light() +
  theme(legend.position = 'none')
```

```r
g3 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = JobInvolveme
  geom_bar(aes(y = after_stat(prop), fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~JobInvolvement) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.1
  labs(y = 'Percent', x = 'Attrition', title = 'Job involvement vs Attrition', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

g4 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = JobSatisfact
  geom_bar(aes(y = after_stat(prop), fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~JobSatisfaction) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.1
  labs(y = 'Percent', x = 'Attrition', title = 'Job Satisfaction vs Attrition', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

g5 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = Performancel
  geom_bar(aes(y = after_stat(prop), fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~PerformanceRating) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.1
  labs(y = 'Percent', x = 'Attrition', title = 'Performance Rating vs Attrition', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

g6 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = WorkLifeBala
  geom_bar(aes(y = after_stat(prop), fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~WorkLifeBalance) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.1
  labs(y = 'Percent', x = 'Attrition', title = 'Work-life balance vs Attrition', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

plot_grid(g1,g2,g3,g4,g5,g6, nrow = 3)
```

```
## Warning: The dot-dot notation (‘..x..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(x)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



Findings:

*We can see that attrition comes most from people with college education, perhaps they're could be interns.

*People that have a low environment satisfaction have a high rate of attrition.

*People with low job involvement have a high rate of attrition.

*People with low job satisfaction have a high rate of attrition.

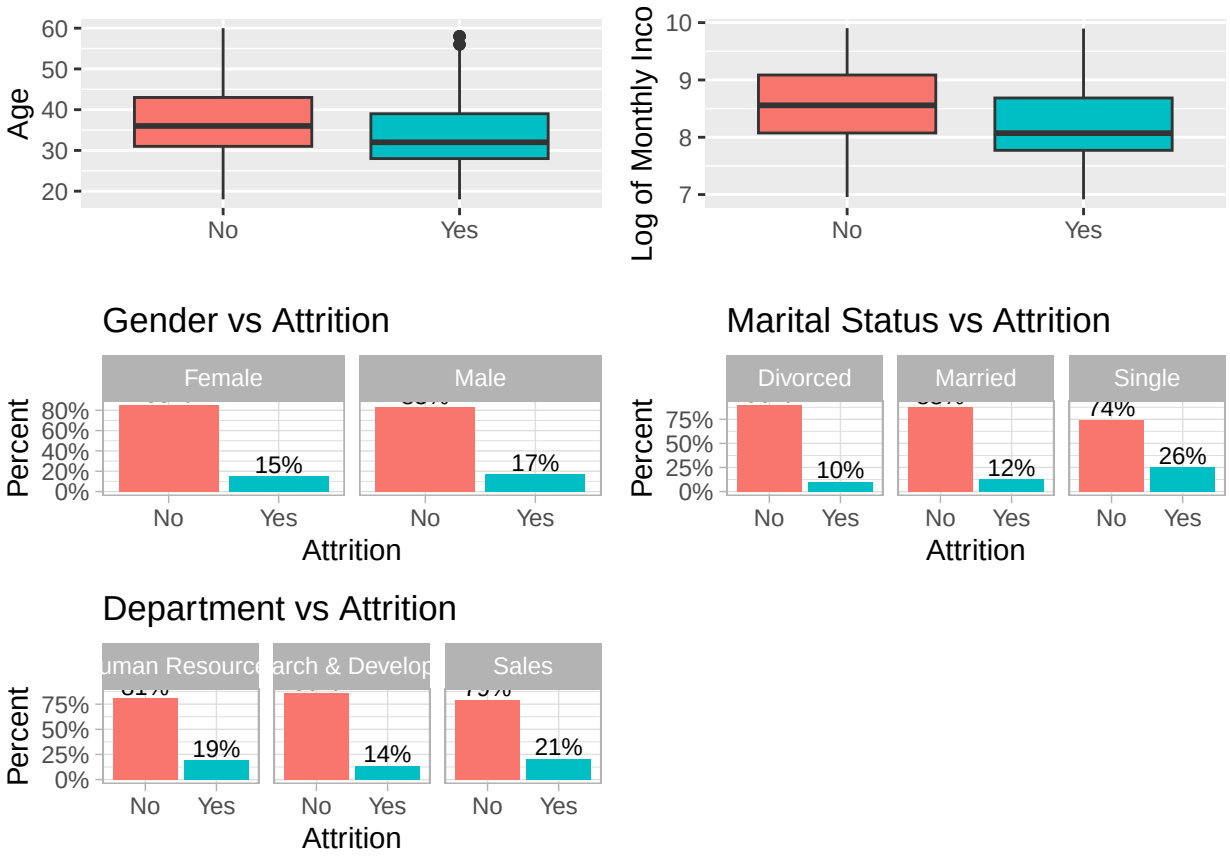
*People with low work-life balance have a high rate of attrition.

*Performance rating does not have a significance difference in the level of attrition.

More Plots

```
g7 <-  ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), y = Age, fill = At
  geom_boxplot() +
  xlab('')+
  theme(legend.position = 'none')

g8 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), y = log(MonthlyInco
  geom_boxplot()+
  xlab('')+
  ylab('Log of Monthly Income')+
  theme(legend.position = 'none')
```

```
g9 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = Gender)) +
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~Gender) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.
  labs(y = 'Percent', x = 'Attrition', title = 'Gender vs Attrition', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

g10 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = MaritalSta
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~MaritalStatus) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.
  labs(y = 'Percent', x = 'Attrition', title = 'Marital Status vs Attrition', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

g11 <- ggplot(data = general_data_merged_with_categories, aes(x = factor(Attrition), group = Department
  geom_bar(aes(y = ..prop.., fill = factor(..x..)), stat = 'count', position = 'stack') +
  facet_grid(~Department) +
  scale_y_continuous(labels = scales::percent) +
  geom_text(aes(label = scales::percent(round((..prop..),2)), y = ..prop..), stat = 'count', vjust = -.
  labs(y = 'Percent', x = 'Attrition', title = 'Department vs Attrition', fill = 'Attrition')+
  theme_light() +
  theme(legend.position = 'none')

plot_grid(g7,g8,g9,g10,g11, nrow = 3)
```



Findings:

*Younger people have higher level of attrition than older people.

*It seems that income and gender are not relevant in the level of attrition.

*People that are single have a higher rate of attrition than the ones that are married or divorced.

*Human Resources have a higher rate of attrition than Research & Development and Sales departments.

Model building

We want to understand the most important factors that lead to employee attrition. For this we use logistic regression to uncover which factors are the most relevant. For this moment we do not want to predict. In the following part Model building: Part 2 we split the data into training and test and predict the outcomes based on the best classification algorithm either logistic regression, decision trees or random forest.

```
model_1 <- glm(formula = Attrition ~ Age+BusinessTravel+Department+DistanceFromHome+Education+
               Gender+JobLevel+MaritalStatus+MonthlyIncome+NumCompaniesWorked+YearsAtCompany+
               JobInvolvement+PerformanceRating+EnvironmentSatisfaction+JobSatisfaction+WorkLifeBalance
               , data = general_data_merged, family = 'binomial')
```

```
summary(model_1)
```

```
##
```

```
## Call:
## glm(formula = Attrition ~ Age + BusinessTravel + Department +
##     DistanceFromHome + Education + Gender + JobLevel + MaritalStatus +
##     MonthlyIncome + NumCompaniesWorked + YearsAtCompany + JobInvolvement +
##     PerformanceRating + EnvironmentSatisfaction + JobSatisfaction +
##     WorkLifeBalance, family = "binomial", data = general_data_merged)
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      2.166e+00  1.065e+00   2.033 0.042058 *
## Age                             -3.299e-02  1.096e-02  -3.010 0.002614 **
## BusinessTravelTravel_Frequently  1.700e+00  3.647e-01   4.660 3.16e-06 ***
## BusinessTravelTravel_Rarely      9.165e-01  3.413e-01   2.685 0.007250 **
## DepartmentResearch & Development -5.616e-01  3.680e-01  -1.526 0.127000
## DepartmentSales                  1.666e-01  3.801e-01   0.438 0.661068
## DistanceFromHome                 3.545e-02  9.336e-03   3.797 0.000146 ***
## Education                       -3.914e-03  7.835e-02  -0.050 0.960152
## GenderMale                       2.669e-01  1.625e-01   1.643 0.100462
## JobLevel                        -5.147e-01  2.492e-01  -2.065 0.038910 *
## MaritalStatusMarried             2.909e-01  2.329e-01   1.249 0.211752
## MaritalStatusSingle              1.147e+00  2.314e-01   4.955 7.22e-07 ***
## MonthlyIncome                    1.024e-06  6.005e-05   0.017 0.986390
## NumCompaniesWorked               1.337e-01  3.276e-02   4.081 4.48e-05 ***
## YearsAtCompany                  -1.455e-02  1.994e-02  -0.730 0.465521
## JobInvolvement                  -5.277e-01  1.078e-01  -4.897 9.73e-07 ***
## PerformanceRating                1.081e-02  2.177e-01   0.050 0.960407
## EnvironmentSatisfaction         -2.867e-01  7.080e-02  -4.049 5.14e-05 ***
## JobSatisfaction                 -3.164e-01  7.041e-02  -4.494 6.98e-06 ***
## WorkLifeBalance                 -3.100e-01  1.083e-01  -2.863 0.004196 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1298.6  on 1469  degrees of freedom
## Residual deviance: 1062.8  on 1450  degrees of freedom
## AIC: 1102.8
##
## Number of Fisher Scoring iterations: 6
```

```
vif(mod = model_1)
```

```
##                            GVIF Df GVIF^(1/(2*Df))
## Age                    1.421084  1        1.192092
## BusinessTravel         1.046590  2        1.011449
## Department             1.147680  2        1.035035
## DistanceFromHome       1.039483  1        1.019550
## Education              1.068287  1        1.033580
## Gender                 1.014208  1        1.007079
## JobLevel               6.917314  1        2.630079
## MaritalStatus          1.051013  2        1.012516
## MonthlyIncome          6.312394  1        2.512448
## NumCompaniesWorked     1.230579  1        1.109315
## YearsAtCompany         1.400105  1        1.183260
```

```
## JobInvolvement          1.016202  1        1.008068
## PerformanceRating       1.010329  1        1.005151
## EnvironmentSatisfaction 1.015001  1        1.007472
## JobSatisfaction         1.023492  1        1.011678
## WorkLifeBalance         1.023230  1        1.011548
```

```
model_2 <- stepAIC(model_1, direction = 'both', trace = FALSE)
```

```
summary(model_2)
```

```
##
## Call:
## glm(formula = Attrition ~ Age + BusinessTravel + Department +
##     DistanceFromHome + Gender + JobLevel + MaritalStatus + NumCompaniesWorked +
##     JobInvolvement + EnvironmentSatisfaction + JobSatisfaction +
##     WorkLifeBalance, family = "binomial", data = general_data_merged)
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      2.194893   0.791643   2.773 0.005561 **
## Age                             -0.034277   0.010725  -3.196 0.001394 **
## BusinessTravelTravel_Frequently  1.701305   0.364491   4.668 3.05e-06 ***
## BusinessTravelTravel_Rarely      0.922773   0.340951   2.706 0.006800 **
## DepartmentResearch & Development -0.559582   0.367416  -1.523 0.127754
## DepartmentSales                  0.180636   0.379049   0.477 0.633681
## DistanceFromHome                 0.035682   0.009294   3.839 0.000123 ***
## GenderMale                       0.271467   0.162268   1.673 0.094336 .
## JobLevel                        -0.551661   0.109887  -5.020 5.16e-07 ***
## MaritalStatusMarried             0.289230   0.232805   1.242 0.214100
## MaritalStatusSingle              1.148037   0.231322   4.963 6.94e-07 ***
## NumCompaniesWorked               0.139149   0.031678   4.393 1.12e-05 ***
## JobInvolvement                  -0.528639   0.107563  -4.915 8.89e-07 ***
## EnvironmentSatisfaction         -0.288678   0.070727  -4.082 4.47e-05 ***
## JobSatisfaction                 -0.315708   0.070330  -4.489 7.16e-06 ***
## WorkLifeBalance                 -0.310503   0.108157  -2.871 0.004093 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1298.6  on 1469  degrees of freedom
## Residual deviance: 1063.3  on 1454  degrees of freedom
## AIC: 1095.3
##
## Number of Fisher Scoring iterations: 5
```

```
vif(mod = model_2)
```

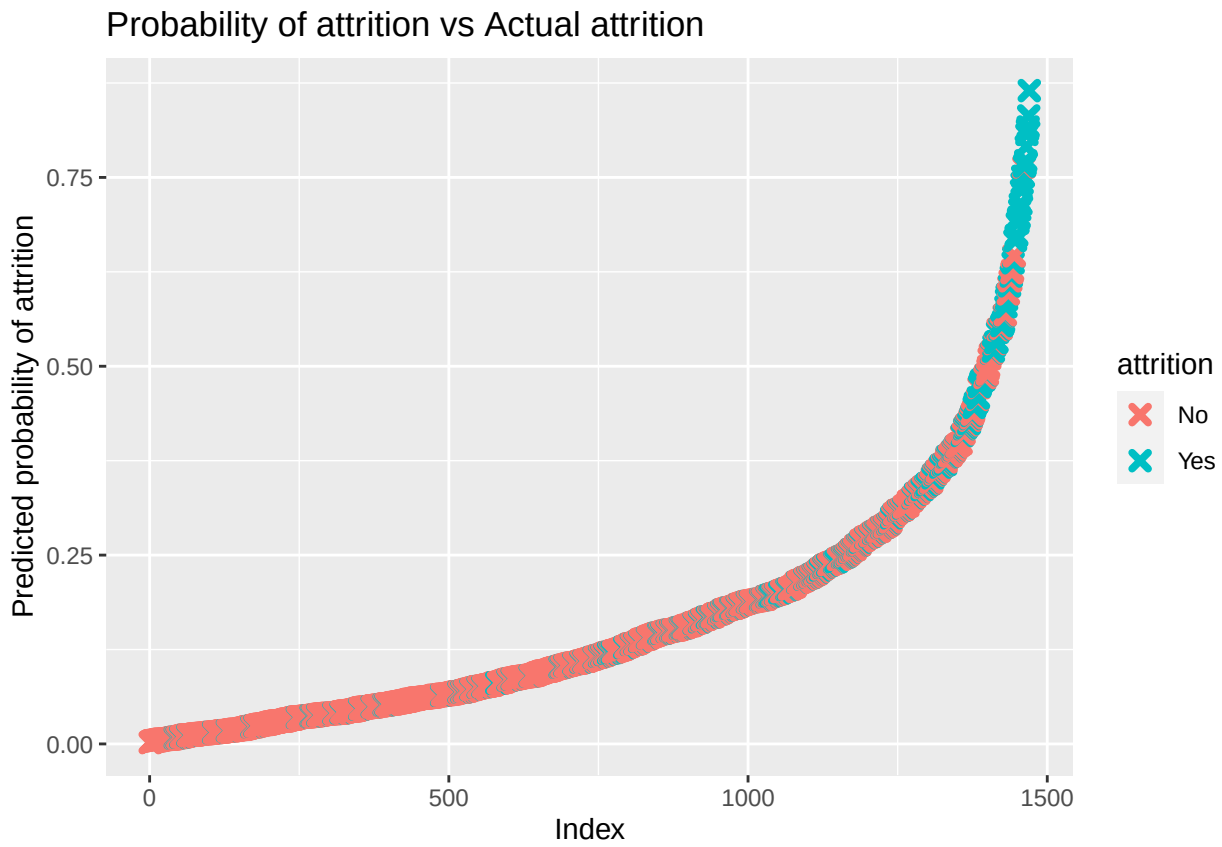
```
##                           GVIF Df GVIF^(1/(2*Df))
## Age                   1.352700  1        1.163056
## BusinessTravel        1.043577  2        1.010721
## Department            1.120682  2        1.028894
```

```
## DistanceFromHome         1.032172  1        1.015959
## Gender                   1.012281  1        1.006122
## JobLevel                 1.344199  1        1.159396
## MaritalStatus            1.047996  2        1.011789
## NumCompaniesWorked       1.156300  1        1.075314
## JobInvolvement           1.014537  1        1.007242
## EnvironmentSatisfaction 1.012771  1        1.006365
## JobSatisfaction          1.022473  1        1.011174
## WorkLifeBalance          1.022393  1        1.011135
```

Assesment

To evaluate our model we create a graph comparing the predicted probabilities against actual attrition.

```
predicted_data <- data.frame(prob_of_attrition = model_2$fitted.values, attrition = general_data_merged$

predicted_data <- predicted_data %>%
  arrange(prob_of_attrition) %>%
  mutate(rank = 1:nrow(predicted_data))

ggplot(data = predicted_data, aes(x = rank, y = prob_of_attrition))+
  geom_point(aes(color = attrition), alpha = 1, shape = 4, stroke = 2)+
  labs(y = 'Predicted probability of attrition', x = 'Index', title = 'Probability of attrition vs Actua
```



The Three Models Application

we want to make predictions on which employee will leave based on several characteristcs gathered from the company. The steps we take are the following:

1-We split the data into training and test datasets.

2-We build three models: Logistic regression, decision trees and random forest and compare their results.

3-Select the best model.

Apply Logistic Regression

```
set.seed(123)
```

```
training.sample <- general_data_merged$Attrition %>%
  createDataPartition(p = 0.8, list = FALSE)
```

```
train.data <- general_data_merged[training.sample,]
test.data <- general_data_merged[-training.sample,]
```

```
model_3 <- glm(formula = Attrition ~ Age+BusinessTravel+Department+DistanceFromHome+Education+
               Gender+JobLevel+MaritalStatus+log(MonthlyIncome)+NumCompaniesWorked+YearsAtCompany+
               JobInvolvement+PerformanceRating+EnvironmentSatisfaction+JobSatisfaction+WorkLifeBalance
               , data = train.data, family = 'binomial')
```

```
model_3_StepWise <- stepAIC(model_3, direction = 'both', trace = FALSE)
```

```
summary(model_3_StepWise)
```

```
##
## Call:
## glm(formula = Attrition ~ Age + BusinessTravel + Department +
##     DistanceFromHome + Gender + MaritalStatus + log(MonthlyIncome) +
##     NumCompaniesWorked + JobInvolvement + EnvironmentSatisfaction +
##     JobSatisfaction + WorkLifeBalance, family = "binomial", data = train.data)
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        8.72288    1.64520   5.302 1.15e-07 ***
## Age                               -0.02960    0.01211  -2.444 0.014522 *
## BusinessTravelTravel_Frequently    2.07137    0.44492   4.656 3.23e-06 ***
## BusinessTravelTravel_Rarely        1.20820    0.41727   2.895 0.003786 **
## DepartmentResearch & Development -0.14368    0.43980  -0.327 0.743906
## DepartmentSales                    0.69701    0.45296   1.539 0.123856
## DistanceFromHome                   0.03701    0.01073   3.449 0.000562 ***
## GenderMale                         0.32787    0.18378   1.784 0.074421 .
## MaritalStatusMarried               0.27720    0.26411   1.050 0.293925
## MaritalStatusSingle                1.14215    0.26227   4.355 1.33e-05 ***
## log(MonthlyIncome)                -0.98310    0.17995  -5.463 4.68e-08 ***
## NumCompaniesWorked                 0.13509    0.03659   3.692 0.000222 ***
```

```
## JobInvolvement                  -0.56030    0.12290  -4.559 5.14e-06 ***
## EnvironmentSatisfaction         -0.30127    0.08032  -3.751 0.000176 ***
## JobSatisfaction                 -0.33189    0.08013  -4.142 3.45e-05 ***
## WorkLifeBalance                 -0.31528    0.12227  -2.579 0.009919 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1040.54  on 1176  degrees of freedom
## Residual deviance:  835.36  on 1161  degrees of freedom
## AIC: 867.36
##
## Number of Fisher Scoring iterations: 5
```

```
p.training <- predict(model_3_StepWise, train.data, type = 'response') # vector of probabilites from th
```

```
p.training.attrition <- as.factor(ifelse(p.training > 0.5, 'Yes', 'No'))
```

```
confusionMatrix(p.training.attrition, train.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  970 142
##        Yes  17  48
##
##                Accuracy : 0.8649
##                  95% CI : (0.844, 0.8839)
##     No Information Rate : 0.8386
##     P-Value [Acc > NIR] : 0.006882
##
##                   Kappa : 0.3206
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9828
##             Specificity : 0.2526
##          Pos Pred Value : 0.8723
##          Neg Pred Value : 0.7385
##              Prevalence : 0.8386
##          Detection Rate : 0.8241
##    Detection Prevalence : 0.9448
##       Balanced Accuracy : 0.6177
##
##        'Positive' Class : No
##
```

```
p.test <- predict(model_3_StepWise, test.data, type = 'response')
```

```r
p.test.attrition <- as.factor(ifelse(p.test > 0.5, 'Yes','No'))
```

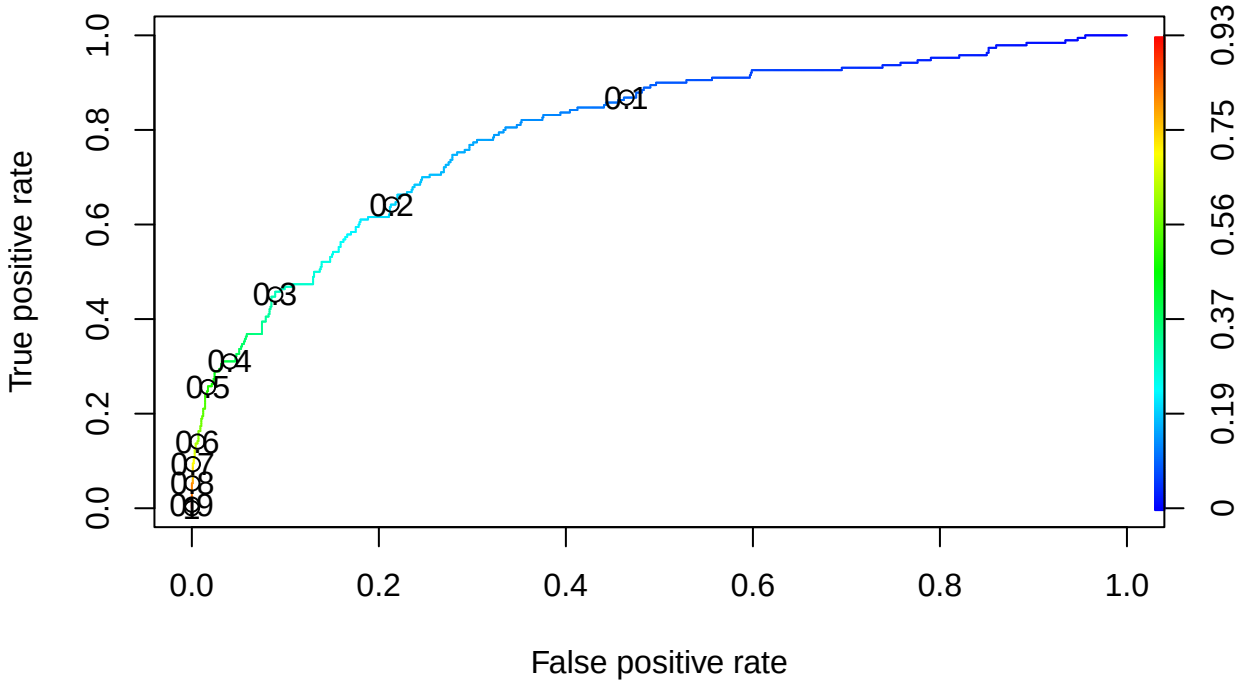
```r
confusionMatrix(p.test.attrition, test.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  240  37
##        Yes   6  10
##
##                Accuracy : 0.8532
##                  95% CI : (0.8075, 0.8917)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.293
##
##                   Kappa : 0.2569
##
##  Mcnemar's Test P-Value : 4.763e-06
##
##             Sensitivity : 0.9756
##             Specificity : 0.2128
##          Pos Pred Value : 0.8664
##          Neg Pred Value : 0.6250
##              Prevalence : 0.8396
##          Detection Rate : 0.8191
##    Detection Prevalence : 0.9454
##       Balanced Accuracy : 0.5942
##
##        'Positive' Class : No
##
```

```r
ROCR_prediction <- prediction(p.training, train.data$Attrition)
```

```r
ROCR_performance <- performance(ROCR_prediction, 'tpr', 'fpr')
```

```r
plot(ROCR_performance, colorize = TRUE, print.cutoffs.at = seq(0.1, by = 0.1))
```




```
p.training <- predict(model_3_StepWise, train.data, type = 'response')
p.training.attrition <- as.factor(ifelse(p.training > 0.2, 'Yes', 'No'))

confusionMatrix(p.training.attrition, train.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  777  68
##        Yes 210 122
##
##                Accuracy : 0.7638
##                  95% CI : (0.7385, 0.7878)
##     No Information Rate : 0.8386
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3298
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7872
##             Specificity : 0.6421
##          Pos Pred Value : 0.9195
##          Neg Pred Value : 0.3675
```

```
##              Prevalence : 0.8386
##          Detection Rate : 0.6602
##    Detection Prevalence : 0.7179
##       Balanced Accuracy : 0.7147
##
##        'Positive' Class : No
##
```

```
p.test <- predict(model_3_StepWise, test.data, type = 'response')
p.test.attrition <-  as.factor(ifelse(p.test> 0.2, 'Yes', 'No'))

confusionMatrix(p.test.attrition, test.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  190  16
##        Yes  56  31
##
##                Accuracy : 0.7543
##                  95% CI : (0.7008, 0.8025)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.9999
##
##                   Kappa : 0.3213
##
##  Mcnemar's Test P-Value : 4.303e-06
##
##             Sensitivity : 0.7724
##             Specificity : 0.6596
##          Pos Pred Value : 0.9223
##          Neg Pred Value : 0.3563
##              Prevalence : 0.8396
##          Detection Rate : 0.6485
##    Detection Prevalence : 0.7031
##       Balanced Accuracy : 0.7160
##
##        'Positive' Class : No
##
```

```
over <- ovun.sample(formula = Attrition ~., data = train.data, method = 'over')$data

table(over$Attrition)
```

```
##
##  No Yes
## 987 996
```

```
balanced_model <- glm(formula = Attrition ~ Age+BusinessTravel+Department+DistanceFromHome+Education+
               Gender+JobLevel+MaritalStatus+log(MonthlyIncome)+NumCompaniesWorked+YearsAtCompany+
               JobInvolvement+PerformanceRating+EnvironmentSatisfaction+JobSatisfaction+WorkLifeBalance
               , data = over, family = 'binomial')
```

```r
summary(balanced_model)
```

```
##
## Call:
## glm(formula = Attrition ~ Age + BusinessTravel + Department +
##     DistanceFromHome + Education + Gender + JobLevel + MaritalStatus +
##     log(MonthlyIncome) + NumCompaniesWorked + YearsAtCompany +
##     JobInvolvement + PerformanceRating + EnvironmentSatisfaction +
##     JobSatisfaction + WorkLifeBalance, family = "binomial", data = over)
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     11.996972   1.711832   7.008 2.41e-12 ***
## Age                             -0.024057   0.007203  -3.340 0.000839 ***
## BusinessTravelTravel_Frequently  2.147821   0.259577   8.274  < 2e-16 ***
## BusinessTravelTravel_Rarely      1.410695   0.239662   5.886 3.95e-09 ***
## DepartmentResearch & Development 0.221738   0.275496   0.805 0.420897
## DepartmentSales                  1.045995   0.285304   3.666 0.000246 ***
## DistanceFromHome                 0.040539   0.006759   5.998 2.00e-09 ***
## Education                       -0.105267   0.054418  -1.934 0.053063 .
## GenderMale                       0.371777   0.109528   3.394 0.000688 ***
## JobLevel                         0.302026   0.130024   2.323 0.020187 *
## MaritalStatusMarried             0.654664   0.153402   4.268 1.98e-05 ***
## MaritalStatusSingle              1.428064   0.157148   9.087  < 2e-16 ***
## log(MonthlyIncome)              -1.323606   0.208382  -6.352 2.13e-10 ***
## NumCompaniesWorked               0.133900   0.023198   5.772 7.83e-09 ***
## YearsAtCompany                   0.019578   0.011082   1.767 0.077294 .
## JobInvolvement                  -0.453334   0.076509  -5.925 3.12e-09 ***
## PerformanceRating               -0.217862   0.147143  -1.481 0.138711
## EnvironmentSatisfaction         -0.230982   0.047168  -4.897 9.73e-07 ***
## JobSatisfaction                 -0.342418   0.048569  -7.050 1.79e-12 ***
## WorkLifeBalance                 -0.291568   0.072520  -4.021 5.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2749.0  on 1982  degrees of freedom
## Residual deviance: 2156.7  on 1963  degrees of freedom
## AIC: 2196.7
##
## Number of Fisher Scoring iterations: 4
```

```r
balanced_model_step_wise <- stepAIC(balanced_model, direction = 'both', trace = FALSE)
```

```r
summary(balanced_model_step_wise)
```

```
##
## Call:
## glm(formula = Attrition ~ Age + BusinessTravel + Department +
##     DistanceFromHome + Education + Gender + JobLevel + MaritalStatus +
##     log(MonthlyIncome) + NumCompaniesWorked + YearsAtCompany +
```

```
##     JobInvolvement + PerformanceRating + EnvironmentSatisfaction +
##     JobSatisfaction + WorkLifeBalance, family = "binomial", data = over)
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     11.996972   1.711832   7.008 2.41e-12 ***
## Age                             -0.024057   0.007203  -3.340 0.000839 ***
## BusinessTravelTravel_Frequently  2.147821   0.259577   8.274  < 2e-16 ***
## BusinessTravelTravel_Rarely      1.410695   0.239662   5.886 3.95e-09 ***
## DepartmentResearch & Development 0.221738   0.275496   0.805 0.420897
## DepartmentSales                  1.045995   0.285304   3.666 0.000246 ***
## DistanceFromHome                 0.040539   0.006759   5.998 2.00e-09 ***
## Education                       -0.105267   0.054418  -1.934 0.053063 .
## GenderMale                       0.371777   0.109528   3.394 0.000688 ***
## JobLevel                         0.302026   0.130024   2.323 0.020187 *
## MaritalStatusMarried             0.654664   0.153402   4.268 1.98e-05 ***
## MaritalStatusSingle              1.428064   0.157148   9.087  < 2e-16 ***
## log(MonthlyIncome)              -1.323606   0.208382  -6.352 2.13e-10 ***
## NumCompaniesWorked               0.133900   0.023198   5.772 7.83e-09 ***
## YearsAtCompany                   0.019578   0.011082   1.767 0.077294 .
## JobInvolvement                  -0.453334   0.076509  -5.925 3.12e-09 ***
## PerformanceRating               -0.217862   0.147143  -1.481 0.138711
## EnvironmentSatisfaction         -0.230982   0.047168  -4.897 9.73e-07 ***
## JobSatisfaction                 -0.342418   0.048569  -7.050 1.79e-12 ***
## WorkLifeBalance                 -0.291568   0.072520  -4.021 5.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2749.0  on 1982  degrees of freedom
## Residual deviance: 2156.7  on 1963  degrees of freedom
## AIC: 2196.7
##
## Number of Fisher Scoring iterations: 4
```

```r
p.training <- predict(balanced_model_step_wise, train.data, type = 'response')
p.training.attrition <- as.factor(ifelse(p.training > 0.5, 'Yes', 'No'))

confusionMatrix(p.training.attrition, train.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  704  50
##        Yes 283 140
##
##                Accuracy : 0.7171
##                  95% CI : (0.6904, 0.7427)
##     No Information Rate : 0.8386
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3011
```

```
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7133
##             Specificity : 0.7368
##          Pos Pred Value : 0.9337
##          Neg Pred Value : 0.3310
##              Prevalence : 0.8386
##          Detection Rate : 0.5981
##    Detection Prevalence : 0.6406
##       Balanced Accuracy : 0.7251
##
##        'Positive' Class : No
##
```

```
p.test <- predict(balanced_model_step_wise, test.data, type = 'response')
p.test.attrition <- as.factor(ifelse(p.test > 0.5, 'Yes', 'No'))

confusionMatrix(p.test.attrition, test.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  167  16
##        Yes  79  31
##
##                Accuracy : 0.6758
##                  95% CI : (0.6189, 0.7291)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2195
##
##  Mcnemar's Test P-Value : 2.004e-10
##
##             Sensitivity : 0.6789
##             Specificity : 0.6596
##          Pos Pred Value : 0.9126
##          Neg Pred Value : 0.2818
##              Prevalence : 0.8396
##          Detection Rate : 0.5700
##    Detection Prevalence : 0.6246
##       Balanced Accuracy : 0.6692
##
##        'Positive' Class : No
##
```

Apply Decisison Tree

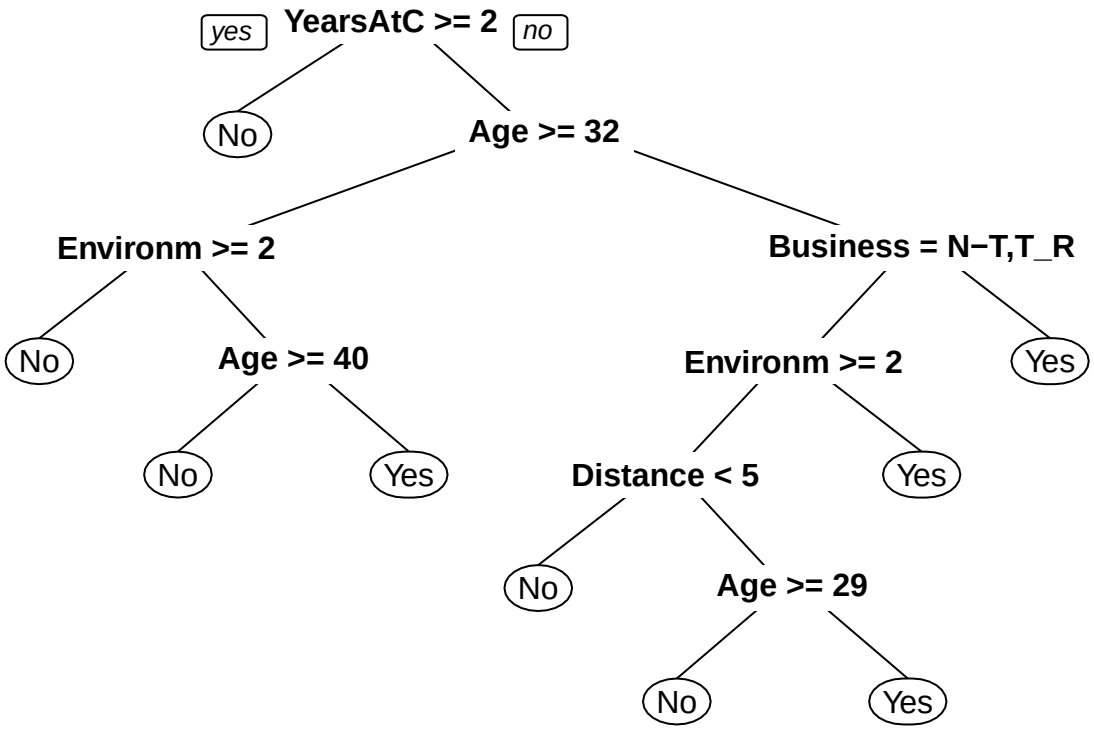
```r
set.seed(123)
```

```r
training.sample <- general_data_merged$Attrition %>%
  createDataPartition(p = 0.8, list = FALSE)
```

```r
train.data <- general_data_merged[training.sample,]
test.data <- general_data_merged[-training.sample,]
```

```r
model.tree <- rpart::rpart(formula = Attrition ~ Age+BusinessTravel+Department+DistanceFromHome+Educati
                           Gender+JobLevel+MaritalStatus+MonthlyIncome+NumCompaniesWorked+YearsAtCompa
                           JobInvolvement+PerformanceRating+EnvironmentSatisfaction+JobSatisfaction+W
                           , data = train.data, method = 'class')
```

```r
rpart.plot::prp(model.tree)
```



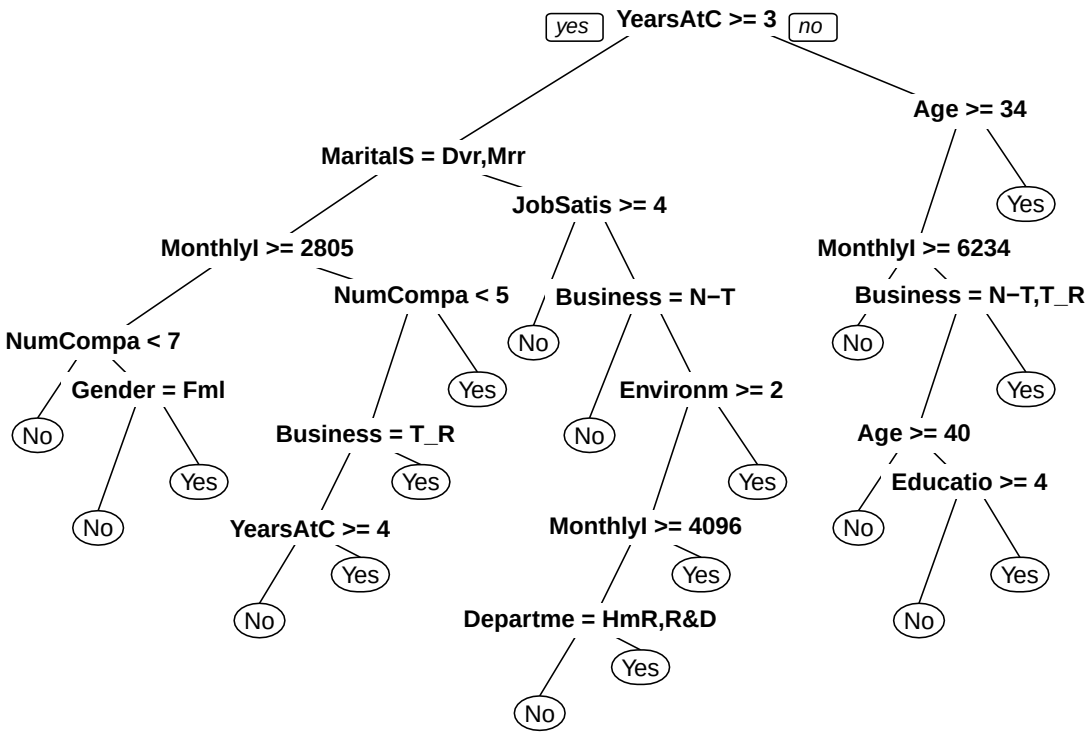

```r
predict_tree <- predict(model.tree, test.data, type = 'class')

confusionMatrix(predict_tree, test.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
```

```
##        No  236  41
##        Yes  10   6
##
##               Accuracy : 0.8259
##                 95% CI : (0.7776, 0.8676)
##    No Information Rate : 0.8396
##    P-Value [Acc > NIR] : 0.7658
##
##                  Kappa : 0.1187
##
##  Mcnemar's Test P-Value : 2.659e-05
##
##             Sensitivity : 0.9593
##             Specificity : 0.1277
##          Pos Pred Value : 0.8520
##          Neg Pred Value : 0.3750
##              Prevalence : 0.8396
##          Detection Rate : 0.8055
##    Detection Prevalence : 0.9454
##       Balanced Accuracy : 0.5435
##
##        'Positive' Class : No
##
```

```
over <- ovun.sample(formula = Attrition ~., data = train.data, method = 'over')$data

model.tree <- rpart::rpart(formula = Attrition ~ Age+BusinessTravel+Department+DistanceFromHome+Educati
                           Gender+JobLevel+MaritalStatus+MonthlyIncome+NumCompaniesWorked+YearsAtCompa
                           JobInvolvement+PerformanceRating+EnvironmentSatisfaction+JobSatisfaction+W
                           , data = over, method = 'class')
```

```
rpart.plot::prp(model.tree)
```




```
predict_tree <- predict(model.tree, test.data, type = 'class')

confusionMatrix(predict_tree, test.data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  183  24
##        Yes  63  23
##
##                Accuracy : 0.7031
##                  95% CI : (0.6472, 0.7548)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1746
##
##  Mcnemar's Test P-Value : 4.621e-05
##
##             Sensitivity : 0.7439
##             Specificity : 0.4894
##          Pos Pred Value : 0.8841
##          Neg Pred Value : 0.2674
##              Prevalence : 0.8396
```

```
##          Detection Rate : 0.6246
##    Detection Prevalence : 0.7065
##       Balanced Accuracy : 0.6166
##
##        'Positive' Class : No
##
```

Apply Random Forest

```
set.seed(123)
```

```
partition <- createDataPartition(general_data_merged$Attrition, p = 0.8, list = FALSE)

training <- general_data_merged[partition,]
test <- general_data_merged[-partition,]
```

```
model_rf_1 <- randomForest::randomForest(Attrition ~ Age+BusinessTravel+Department+DistanceFromHome+Edu
                              Gender+JobLevel+MaritalStatus+MonthlyIncome+NumCompaniesWorked+YearsAtCompa
                              JobInvolvement+PerformanceRating+EnvironmentSatisfaction+JobSatisfaction+W
                              , data = training, trControl = trainControl(method = 'cv', 10))
```

```
model_rf_1
```

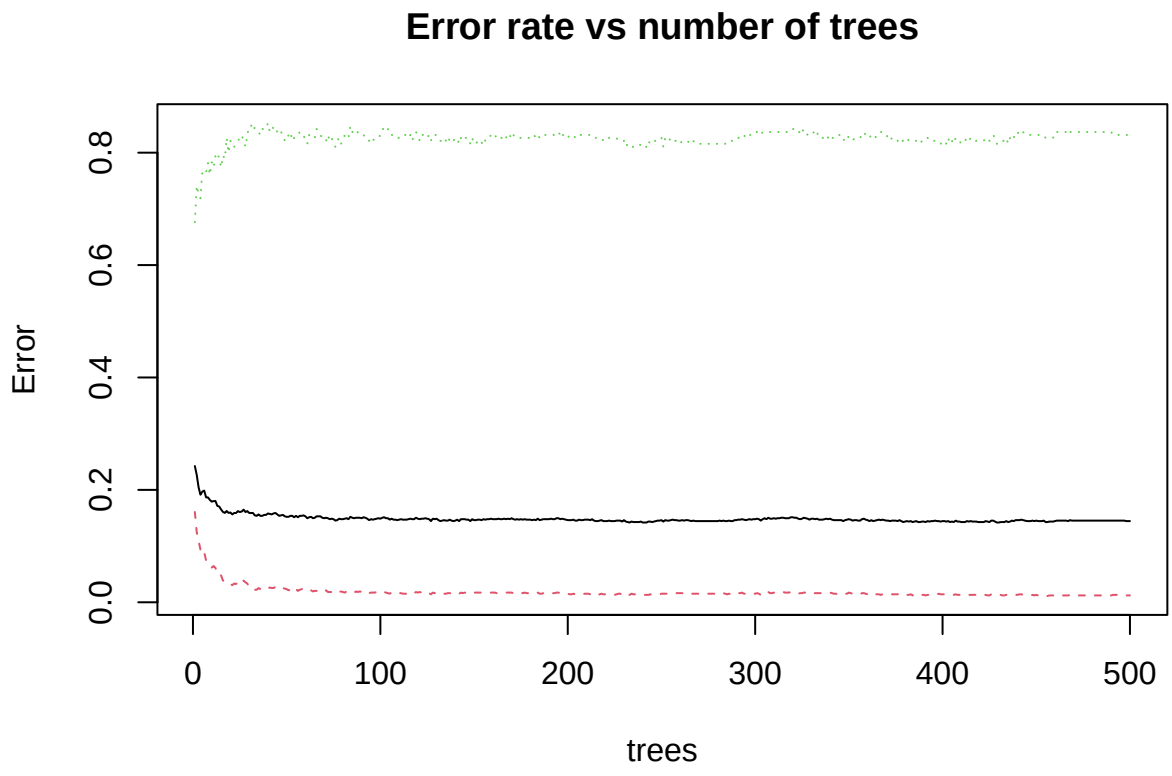
```
##
## Call:
##  randomForest(formula = Attrition ~ Age + BusinessTravel + Department +      DistanceFromHome + Educa
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 14.44%
## Confusion matrix:
##      No Yes class.error
## No  975  12  0.01215805
## Yes 158  32  0.83157895
```

```
p <- predict(model_rf_1, test, type = 'response')

confusionMatrix(p, test$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  241  42
##        Yes   5   5
##
##                Accuracy : 0.8396
##                  95% CI : (0.7925, 0.8797)
##     No Information Rate : 0.8396
```

```
##     P-Value [Acc > NIR] : 0.5388
##
##                   Kappa : 0.1263
##
##  Mcnemar's Test P-Value : 1.512e-07
##
##             Sensitivity : 0.9797
##             Specificity : 0.1064
##          Pos Pred Value : 0.8516
##          Neg Pred Value : 0.5000
##              Prevalence : 0.8396
##          Detection Rate : 0.8225
##    Detection Prevalence : 0.9659
##       Balanced Accuracy : 0.5430
##
##        'Positive' Class : No
##
```

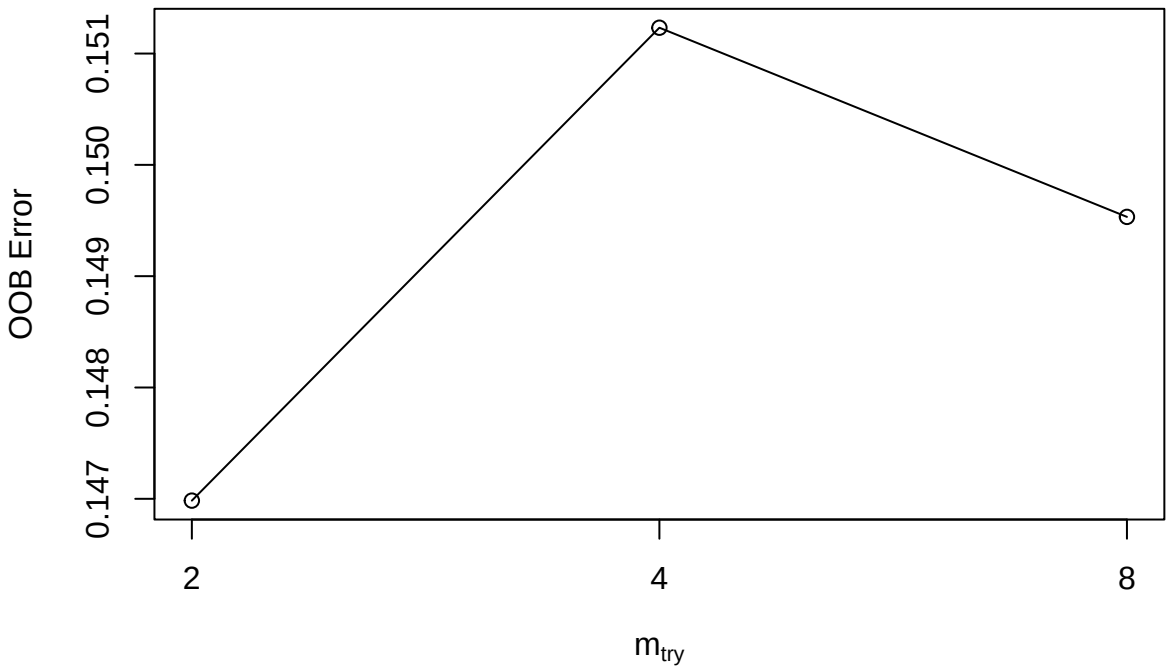
```r
plot(model_rf_1, main = 'Error rate vs number of trees')
```




```r
t <- randomForest::tuneRF(training[,-2], training[,2], stepFactor = 0.5, plot = TRUE, ntreeTry = 100, t
```

```
## mtry = 4  OOB error = 15.12%
## Searching left ...
## mtry = 8     OOB error = 14.95%
```

```
## 0.01123596 0.05
## Searching right ...
## mtry = 2     OOB error = 14.7%
## 0.02808989 0.05
```



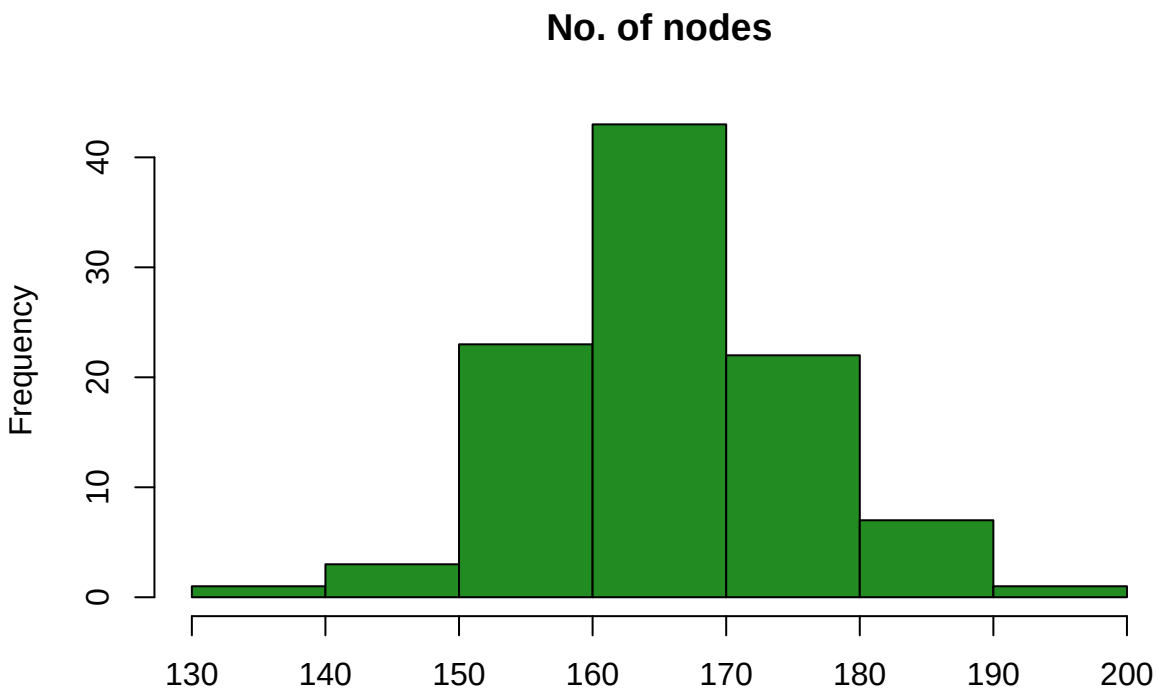

```
model_rf_2 <- randomForest::randomForest(Attrition ~ Age+BusinessTravel+Department+DistanceFromHome+Edu
                                         Gender+JobLevel+MaritalStatus+MonthlyIncome+NumCompaniesWork
                                         JobInvolvement+PerformanceRating+EnvironmentSatisfaction+Job
                                         , data = training, ntree = 100, trControl = trainControl(met

model_rf_2
```

```
##
## Call:
##  randomForest(formula = Attrition ~ Age + BusinessTravel + Department +      DistanceFromHome + Educa
##                Type of random forest: classification
##                      Number of trees: 100
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 14.36%
## Confusion matrix:
##      No Yes class.error
## No  974  13  0.01317123
## Yes 156  34  0.82105263
```

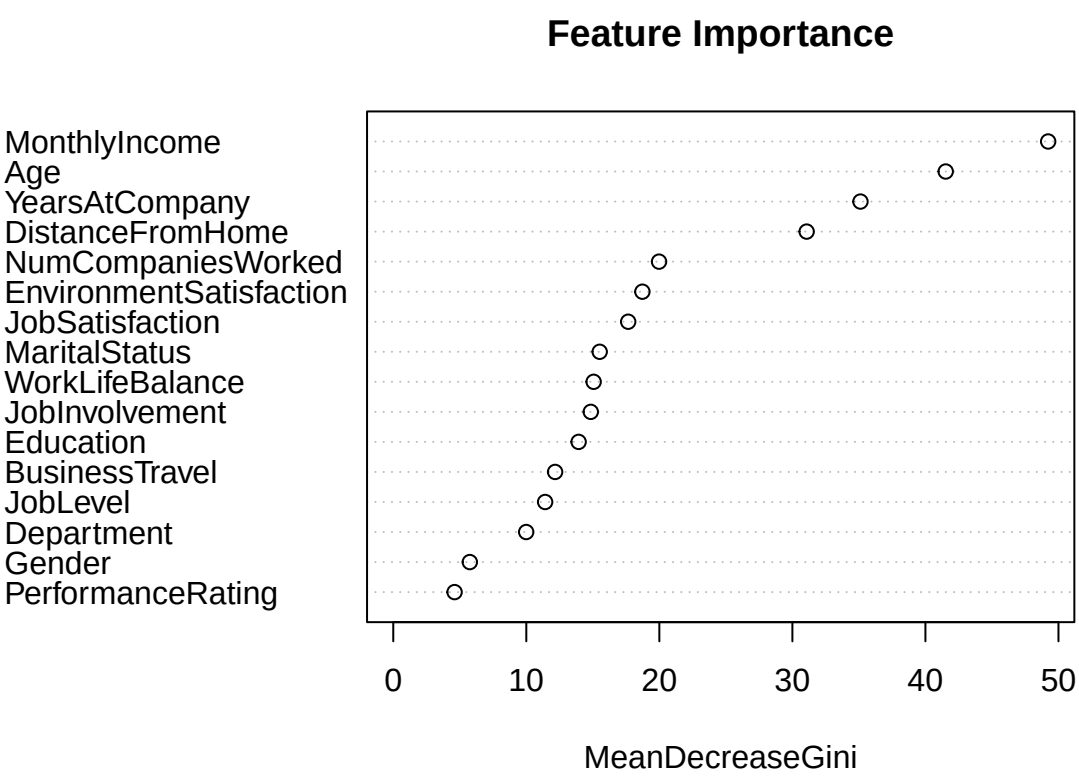
```r
p <- predict(model_rf_2, test, type = 'response')

confusionMatrix(p, test$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  240  42
##        Yes   6   5
##
##                Accuracy : 0.8362
##                  95% CI : (0.7887, 0.8767)
##     No Information Rate : 0.8396
##     P-Value [Acc > NIR] : 0.6009
##
##                   Kappa : 0.1188
##
##  Mcnemar's Test P-Value : 4.376e-07
##
##             Sensitivity : 0.9756
##             Specificity : 0.1064
##          Pos Pred Value : 0.8511
##          Neg Pred Value : 0.4545
##              Prevalence : 0.8396
##          Detection Rate : 0.8191
##    Detection Prevalence : 0.9625
##       Balanced Accuracy : 0.5410
##
##        'Positive' Class : No
##
```

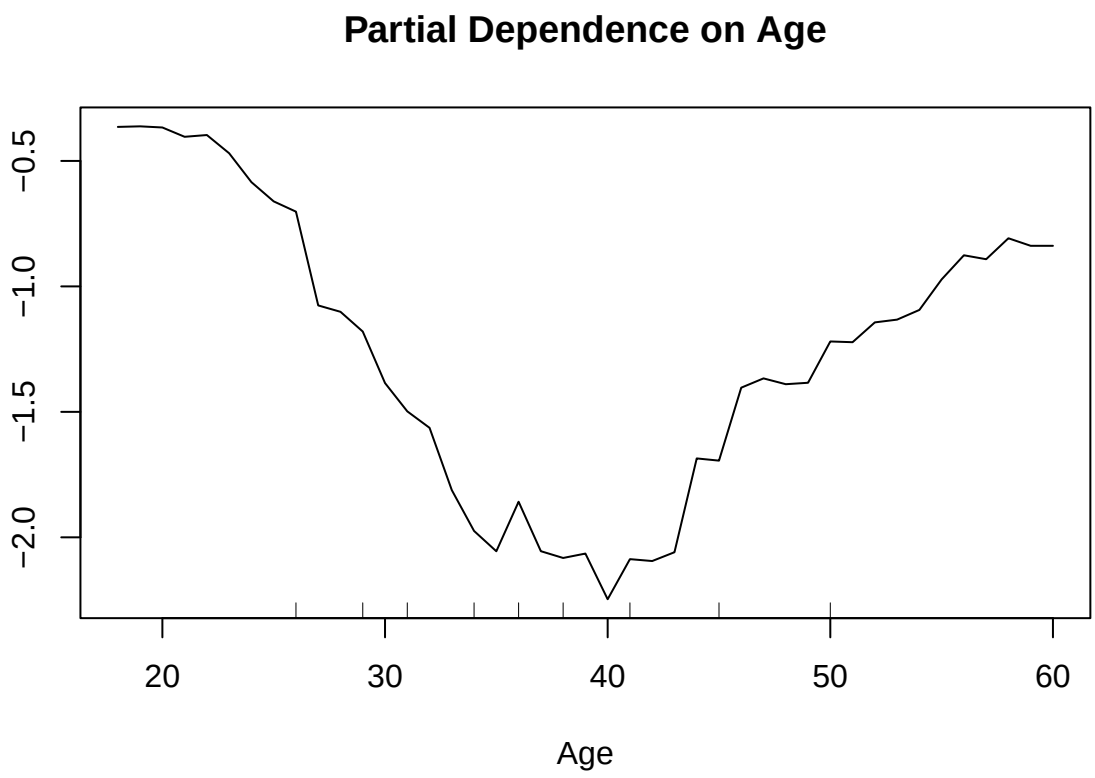
```r
hist(randomForest::treesize(model_rf_2), main = 'No. of nodes', xlab = '', col = 'forestgreen')
```




```
randomForest::varImpPlot(model_rf_2, main = 'Feature Importance')
```




```
randomForest::partialPlot(model_rf_2, training, Age, 'Yes')
```


Partial Dependence on Age
−0.5
−1.0
−1.5
−2.0
20
30
40
50
60
Age