Business Analytics: HW 3

Harsh Yadav (hy1217), Nitish Dabas(nd1292), Kush Shah (ks4437)

December 14, 2017

Description:

The objective of this homework is to predict which valuable employees of a company will leave next. The variable of interest is *left*. The description of the other variables are listed below.

Data

satisfaction_level: Level of satisfaction (0-1)

last_evaluation: Time since last performance evaluation (in Years)

number_project: Number of projects completed while at work

average_montly_hours: Average monthly hours at workplace

time_spend_company: Number of years spent in the company

Work_accident: Whether the employee had a workplace accident

left: Whether the employee left the workplace or not (1 or 0) Factor

promotion_last_5years: Whether the employee was promoted in the last five years

sales: Department in which they work for

salary: Relative level of salary (high)

```
data <- read.csv("C:/Users/Harsh Yadav/Desktop/hw3_data.csv")
str(data)
```

```
## 'data.frame':    10000 obs. of  11 variables:
##  $ X                    : int  9571 12021 8870 13444 12079 14466 6107 10945 6932 7553 ...
##  $ satisfaction_level   : num  0.6 0.11 0.91 0.65 0.43 0.6 0.61 0.62 0.61 0.37 ...
##  $ last_evaluation      : num  0.59 0.83 0.5 0.85 0.56 0.92 0.52 0.51 0.67 0.71 ...
##  $ number_project       : int  5 6 4 4 2 2 4 4 3 2 ...
##  $ average_montly_hours : int  146 282 231 201 157 258 171 193 188 139 ...
##  $ time_spend_company   : int  4 4 3 10 3 5 2 3 5 4 ...
##  $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ left                 : int  0 1 0 0 1 1 0 0 0 0 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sales                : Factor w/ 10 levels "accounting","hr",..: 1 8 8 9 8 8 5 1 3 7 ...
##  $ salary               : Factor w/ 3 levels "high","low","medium": 2 2 1 2 2 2 2 1 2 1 ...
```

Here, we can see that there are 10000 observations in our dataset with 11 variables. We have to take predict $left(target variable) based on all the other predictors.

```
sum(is.na(data))
```

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

This observation shows that there are no NA values in our dataset therefore there is no need to change the dataset.

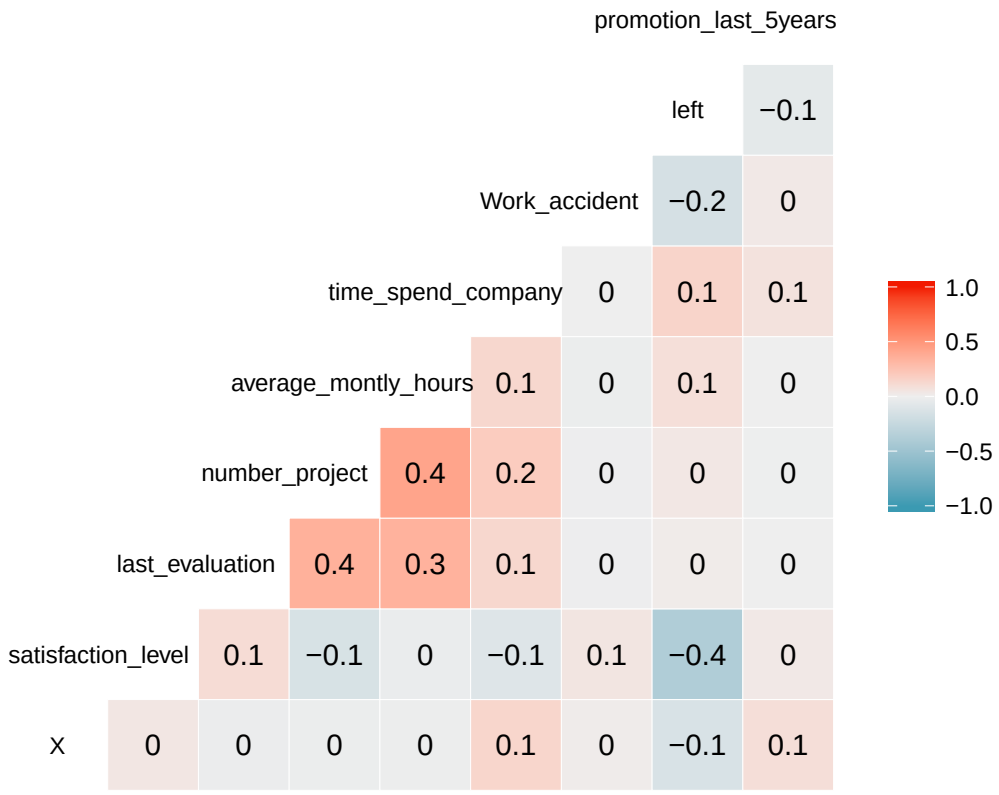
```
summary(data)
```

```
##        X         satisfaction_level last_evaluation  number_project
##  Min.   :    1   Min.   :0.0900     Min.   :0.3600   Min.   :2.000
##  1st Qu.: 3724   1st Qu.:0.4400     1st Qu.:0.5600   1st Qu.:3.000
##  Median : 7500   Median :0.6400     Median :0.7200   Median :4.000
##  Mean   : 7489   Mean   :0.6123     Mean   :0.7178   Mean   :3.819
##  3rd Qu.:11272   3rd Qu.:0.8100     3rd Qu.:0.8700   3rd Qu.:5.000
##  Max.   :14999   Max.   :1.0000     Max.   :1.0000   Max.   :7.000
##
##  average_montly_hours time_spend_company Work_accident         left
##  Min.   : 96.0        Min.   : 2.000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.:156.0        1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median :201.0        Median : 3.000     Median :0.0000   Median :0.0000
##  Mean   :201.6        Mean   : 3.515     Mean   :0.1449   Mean   :0.2377
##  3rd Qu.:246.0        3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :310.0        Max.   :10.000     Max.   :1.0000   Max.   :1.0000
##
##  promotion_last_5years         sales         salary
##  Min.   :0.0000        sales      :2748   high  : 812
##  1st Qu.:0.0000        technical  :1819   low   :4901
##  Median :0.0000        support    :1519   medium:4287
##  Mean   :0.0215        IT         : 791
##  3rd Qu.:0.0000        product_mng: 625
##  Max.   :1.0000        marketing  : 592
##                        (Other)    :1906
```

The above observation represents the summary of each of the feature of our dataset representing their min, max, quartile values.

```
library(GGally)
ggcorr(data, label_size = 4, size = 3,hjust = 0.80, label = TRUE)
```

```
## Warning in ggcorr(data, label_size = 4, size = 3, hjust = 0.8, label =
## TRUE): data in column(s) 'sales', 'salary' are not numeric and were ignored
```

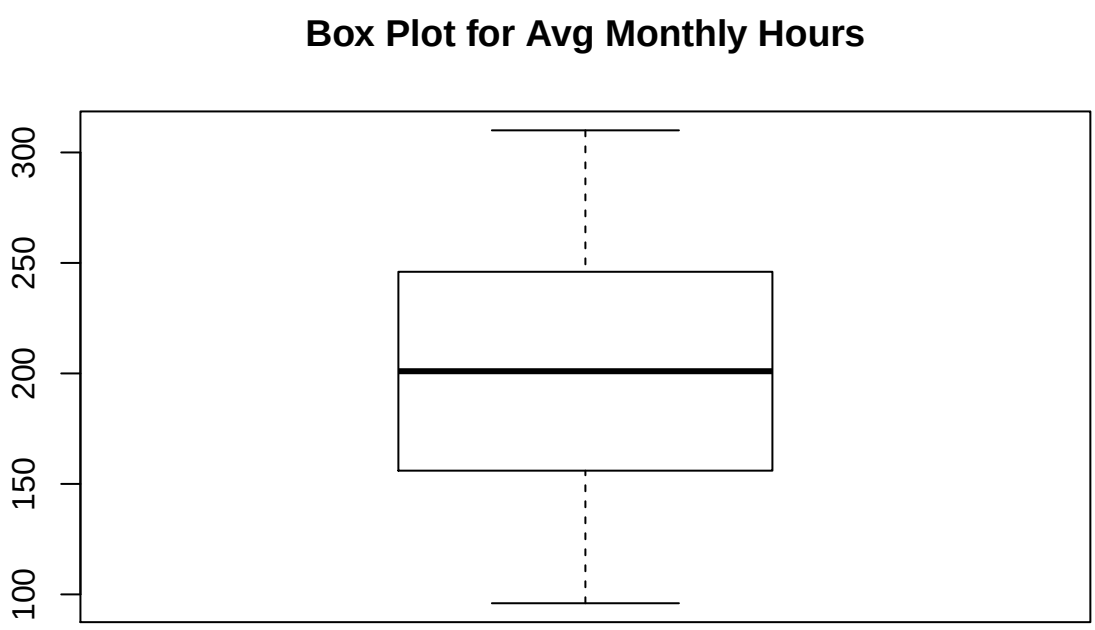


The correlation graph given above shows correlations between all the variables. We can see that there are positive as well as negative correlations between the variables but none of the varibales are highly correlated with each other. This matrix is useful for our EDA analysis since we can choose the parameters to plot the graph effectively.

```
library(ggplot2)
```

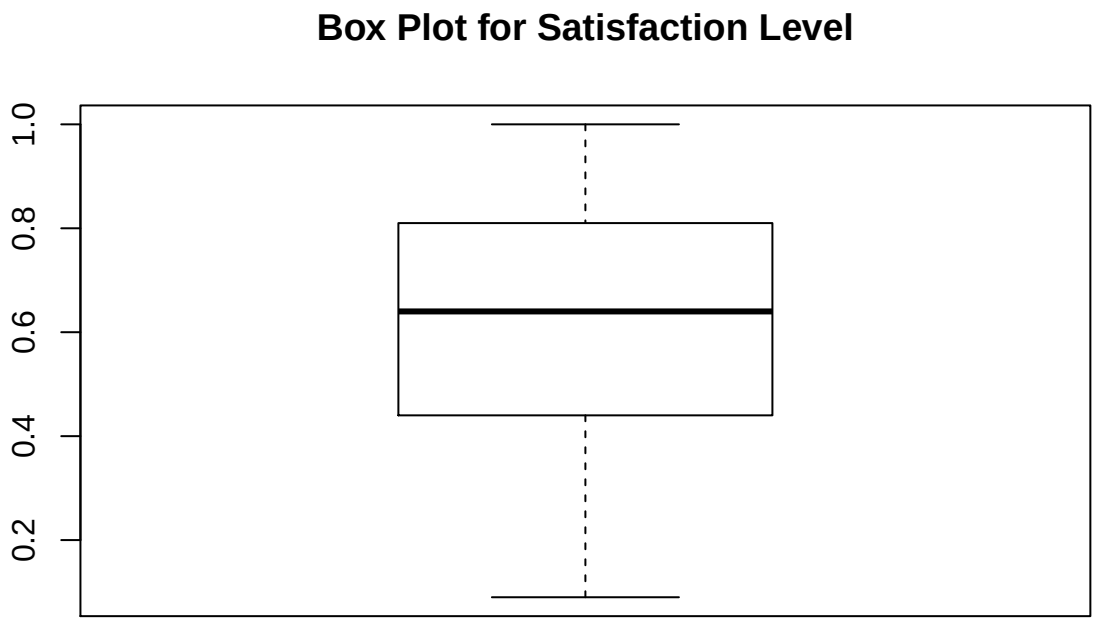
```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
#mean(data$average_montly_hours)
#range(data$average_montly_hours)
boxplot(data$average_montly_hours,main="Box Plot for Avg Monthly Hours")
```



It can be inferred from the box plot that an employee works for Avg Monthly hours in the range of 100-310(approx.) and on an average , the avg monthly hours for each employee in this company is around 200. It can also be noted that there are no outliers in the observations for this parameter. We have also calculated the range and mean by using the functions to check our observations based on box plot.

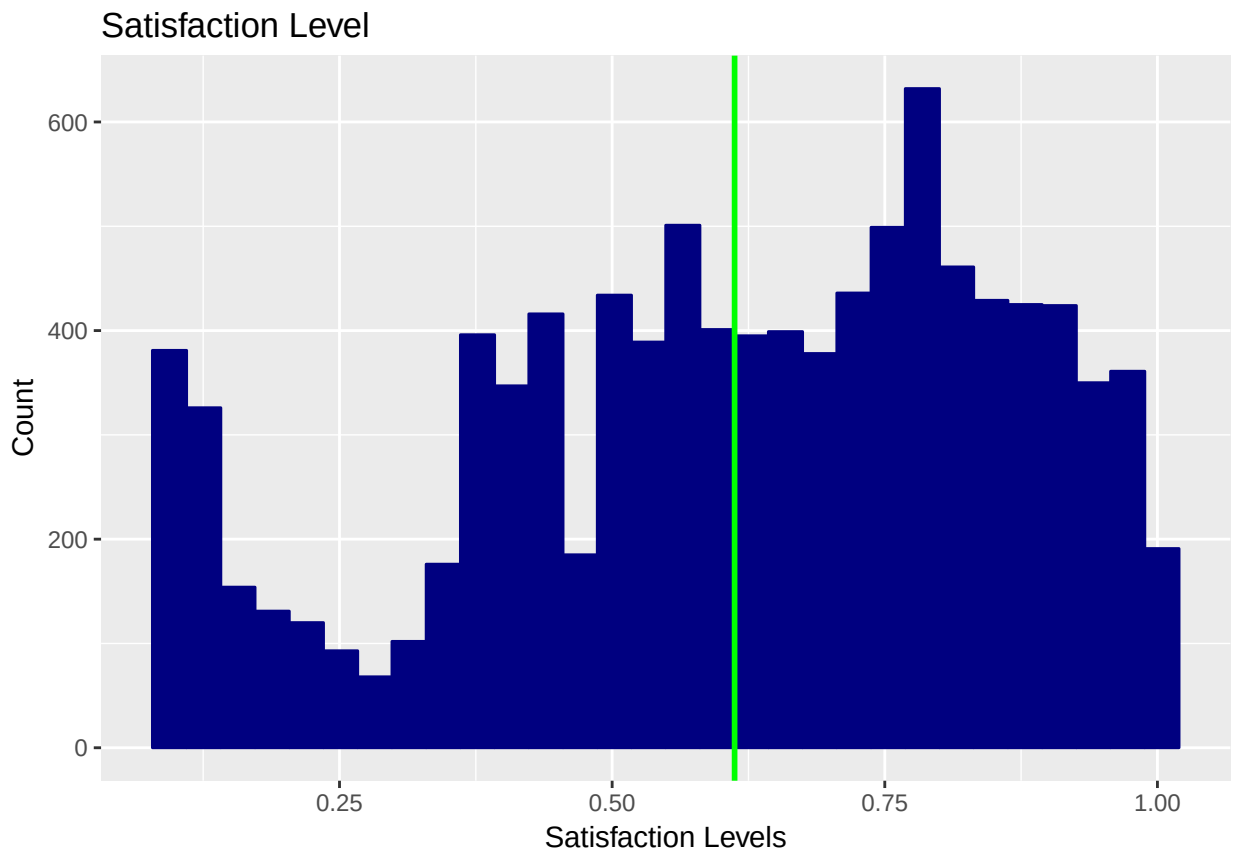
```
library(ggplot2)
#mean(data$satisfaction_level)
#range(data$satisfaction_level)
boxplot(data$satisfaction_level,main="Box Plot for Satisfaction Level")
```



Based on the above box plot, the least satisfaction level recorded by an employee is close to 0.1(approx.) and the avg satisfaction levels for the employees is 0.61. The maximum satisfaction level can be 1. Also, 50% of the employees have satisfaction level between 0.4 to 0.8 (25th to 75th percentile).

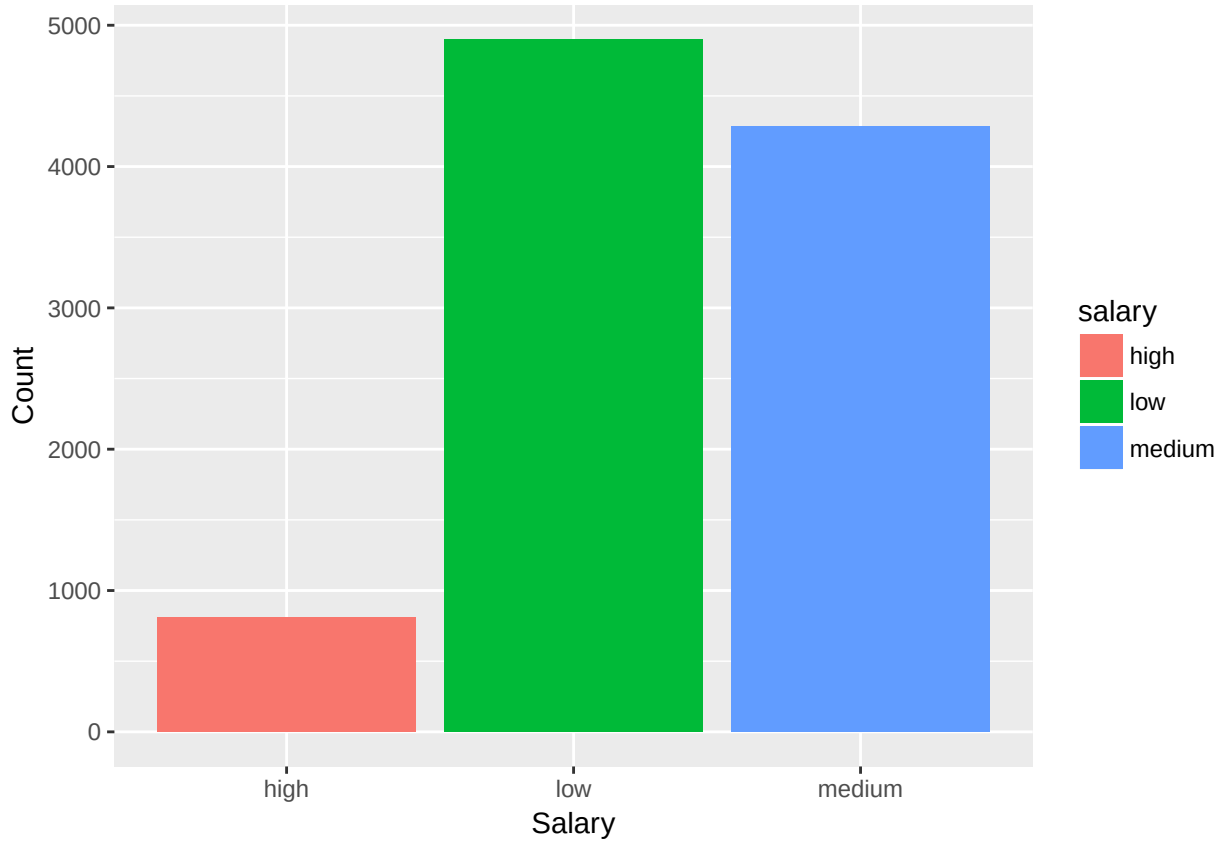
```
library(ggplot2)
#mean(data$satisfaction_level)
ggplot(data, aes(x=satisfaction_level)) +
    geom_histogram(colour="navy", fill="navy") +
    ggtitle("Satisfaction Level") +
    labs(x="Satisfaction Levels",y="Count") +
    geom_vline(aes(xintercept=mean(satisfaction_level, na.rm=T)),
               color="green", linetype="solid", size=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



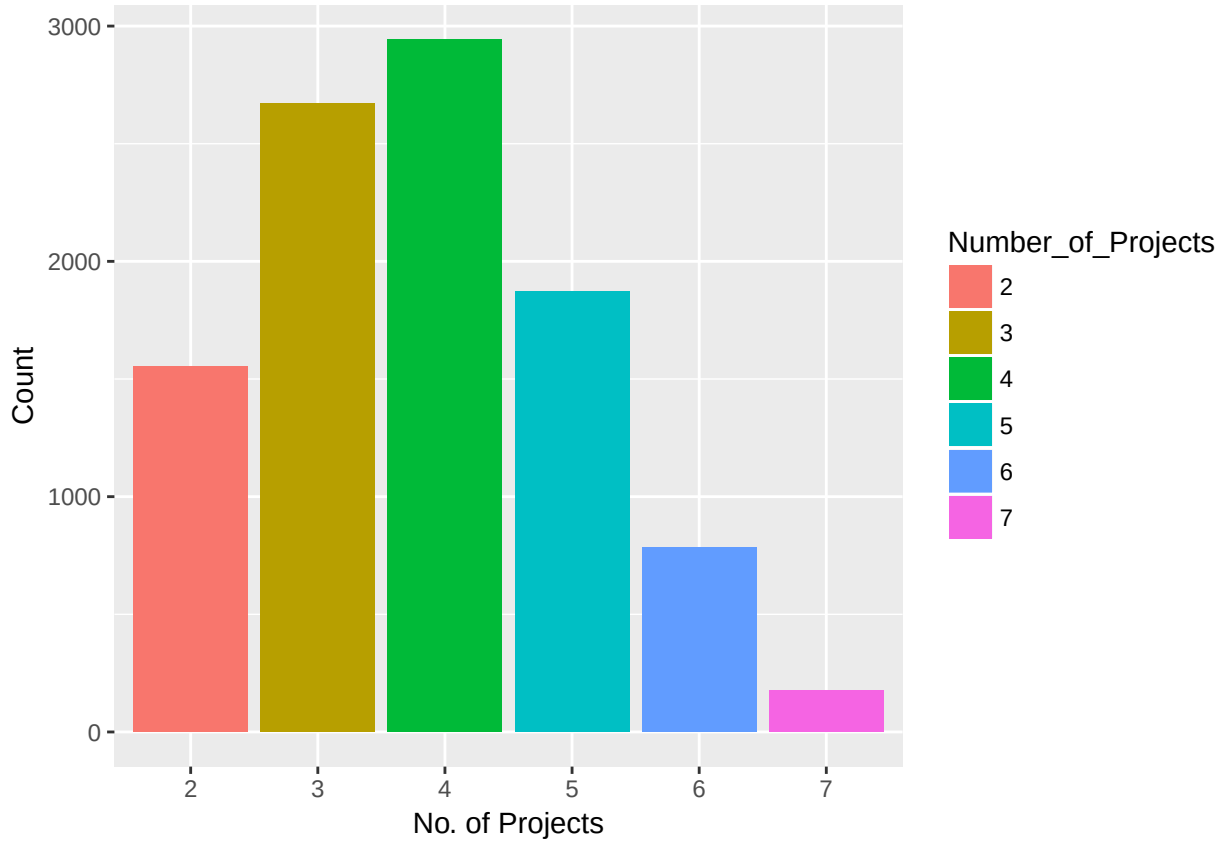
The above histogram for satisfaction level validates the results we derived by the box plot discussed above.

```
ggplot(data,aes(x=salary, fill=salary)) + geom_bar() + labs(x="Salary",y="Count")
```



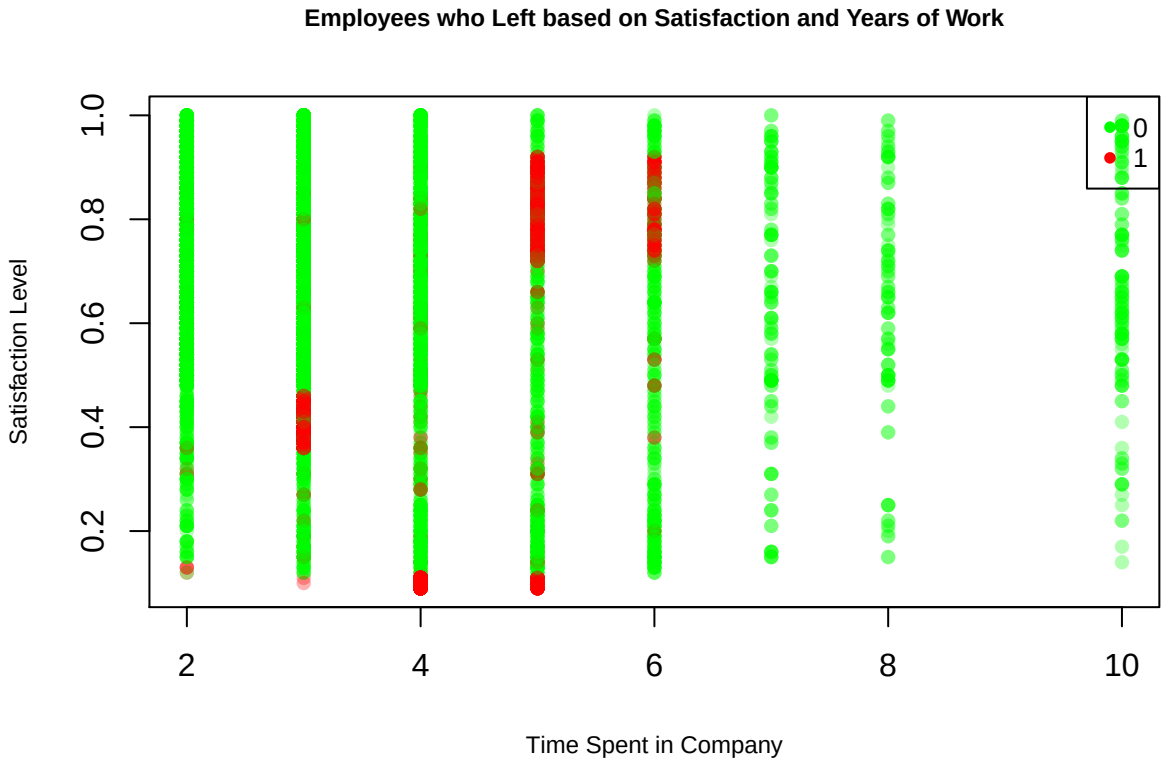
This bar plots shows the number of employees in each salary group. We can see that most number of employees are in the low salary slab whereas there are least number of employees in the high salary slab. This can be related to a common scenario since there are less number of people for eg. managers in a company who have higher salaries.

```
Number_of_Projects<-factor(data$number_project)
ggplot(data,aes(x=Number_of_Projects, fill=Number_of_Projects)) + geom_bar()+
  labs(x="No. of Projects",y="Count")
```



It can be inferred from the above bar plot that most number of people are working on 4 projects at a time whereas there are only a few people who are working on 6 or 7 projects.

```
left1<-factor(data$left)
mycolors = c('green','red')
plot(data$time_spend_company, data$satisfaction_level, pch = 16,
     col = alpha(mycolors[left1], 0.3),
     main="Employees who Left based on Satisfaction and Years of Work",
     xlab="Time Spent in Company", ylab="Satisfaction Level",
     cex.main=0.75,
     cex.lab=0.75)
legend("topright", legend=levels(left1), pch=16, cex=0.8, col=unique(mycolors))
```



The above scatter plot gives a few important observations as mentioned below:

1. Employees who have worked for more than 6 years are less likely to leave the company (0 in this case) even if there satisfaction level is low , since there are no red points in the plot after 6 years.

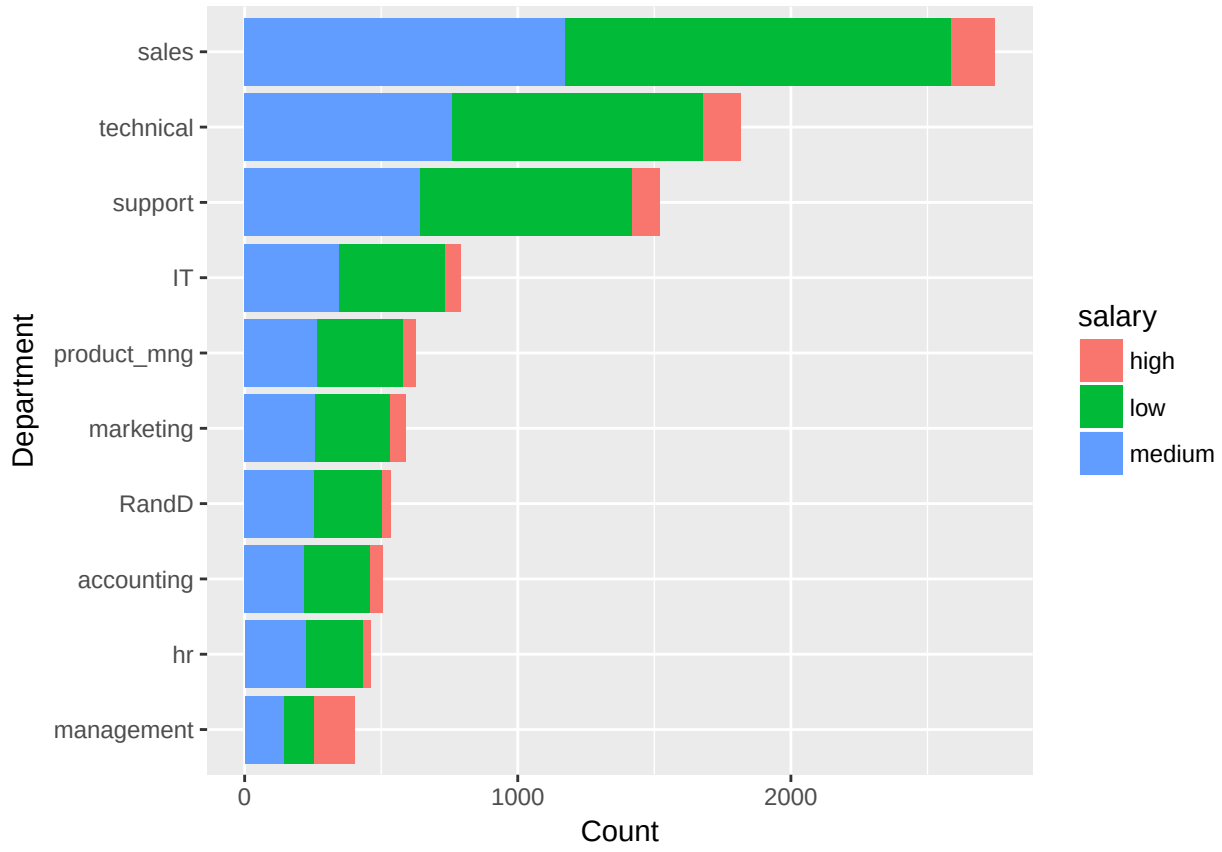
2. Employees who have worked for 2 years are less likely to leave the company irrespective of their satisfaction level since they are at the start of their careers.

3. Employees who have worked for 5,6 years are most likely to leave the company even if they are highly satisfied with what they do. This decision can be based on the future aspirations of an individual since 4-5 years of experience gives you a good market value to switch companies.

```
library(forcats)
```

```
## Warning: package 'forcats' was built under R version 3.4.2
```

```
p<-ggplot(data,aes(x=fct_rev(fct_infreq(sales)), fill=salary)) +
  geom_bar()+
  labs(y="Count",x="Department")
p + coord_flip()
```



The above bar plot makes a few interesting observations:

1. The most number of people in any department are in the low salary slab whereas least number of people are on high salary payscale. We can relate this to the bar plot we saw earlier.

2. Most number of employees in the company are in the sales department followed by technical department.

3. The management department has least number of people since we only need a single manager to manage number of employees.

Predictive Analysis

Random Forest Algorithm

```
data1 <- read.csv("C:/Users/Harsh Yadav/Desktop/hw3_data.csv")
```

Here we have again loaded our dataset into variable named data1 for our predictive modelling using the Random forest algorithm.

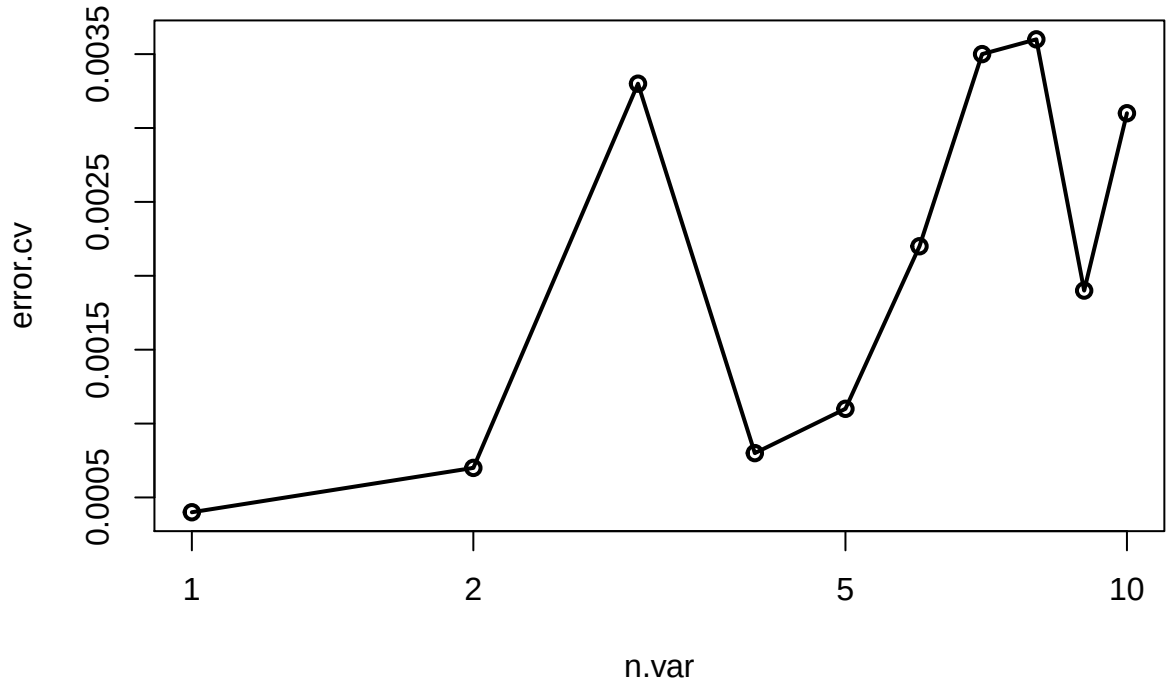
```
require(randomForest)
```

```
## Loading required package: randomForest
## Warning: package 'randomForest' was built under R version 3.4.2
## randomForest 4.6-12
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
data1$left=as.factor(data1$left)
df <- subset(data1, select = -c(left))
result <- rfcv(df, data1$left, cv.fold=10, scale = "log", step=0.9)
```

After importing the randomForest library, we have created another dataframe named df that has all the features from our origional dataset named data1. In this new dataframe df we have selected all the features except our target variable 'left'. Then we have performed 10 fold cross validation on the dataframe df mentioning the target variable as 'left'.

```
with(result, plot(n.var, error.cv, log="x", type="o", lwd=2))
```




```
result$error.cv
```

```
##     10      9      8      7      6      5      4      3      2      1
## 0.0031 0.0019 0.0036 0.0035 0.0022 0.0011 0.0008 0.0033 0.0007 0.0004
```

Here we can observe that the cross validation error is least when we consider just one variable, but that is not good for training our model, as in this case we are just making prediction based on one variable and the other variables are ignored, which may also be important. So, we definitely need to consider other variables for our analysis. Further, we can see that for all the cases, that we have considered, the cross validation error is very less. So, if we consider 6 variables then we have the same cross validation as for considering 9 variables which is around 0.0024. So, we can perform our analysis considering 6 or 9 variables. We can see which features to select by plotting the feature importance below.

```
set.seed(415)
```

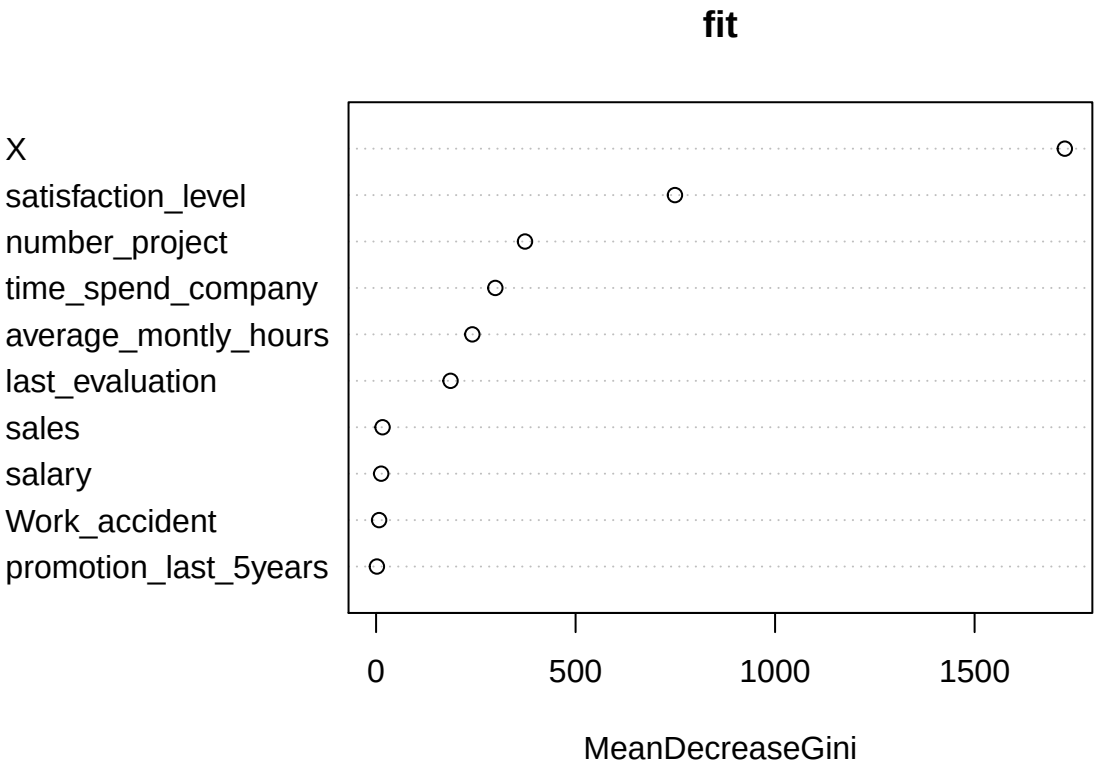
```
data2 <- sample(2,nrow(data1),replace=TRUE,prob=c(0.7,0.3))
trainData <- data1[data2==1,]
testData <- data1[data2==2,]
```

Here, we have set the seed to 415 and divided our dataframe into training and training data in the ratio of 70:30 respectively.

```
library(randomForest)
fit <- randomForest(left ~X+satisfaction_level+last_evaluation+
                      number_project+average_montly_hours+time_spend_company+
                      Work_accident+promotion_last_5years+sales+salary,
                      data=data1, ntree=100)
```

Here, we trained our random forest model using all the predictor variables present in our dataset. We have set the target variable as left and have taken the complete data for our analysis. We have set the number of trees to 100 for our model.

```
varImpPlot(fit)
```



In the feature importance plot we can see that in determining the Gini value, variables like promotion_last_5years, Work_accident, salary and sales are not that important as their values are almost near 0 and they have almost negligible contribution in making predictions.

So, if we ignore these 4 variables and train our model on just 6 variables then we will get almost same predicted results as for the case when we consider all 10 variables. So, now we will just consider 6 variables for our analysis using random forest algorithm.

```
library(randomForest)
fit1 <- randomForest(left ~X+satisfaction_level+last_evaluation+
                       number_project+average_montly_hours+
                       time_spend_company, data=trainData, ntree=100)
```

Here,we have trained our random forest predictive model using the 6 most important features, that we discussed above and have created a model named fit1. We have trained the model using the trainData model.

```
Pred<-predict(fit1,newdata=testData)
conf <- table(Pred, testData$left)
```

Here we have calculated the predicted results of the 'left' target variable and the results re stored in Pred. then we have created a table named conf, which we will use for evaluating the confusion matrix.

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.4.2
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.4.2
```

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

```
confusionMatrix(conf)
```

```
## Confusion Matrix and Statistics
##
##
## Pred    0    1
##    0 2203   12
##    1    0  718
##
##                Accuracy : 0.9959
##                  95% CI : (0.9929, 0.9979)
##     No Information Rate : 0.7511
```

```
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.989
##  Mcnemar's Test P-Value : 0.001496
##
##             Sensitivity : 1.0000
##             Specificity : 0.9836
##          Pos Pred Value : 0.9946
##          Neg Pred Value : 1.0000
##              Prevalence : 0.7511
##          Detection Rate : 0.7511
##    Detection Prevalence : 0.7552
##       Balanced Accuracy : 0.9918
##
##        'Positive' Class : 0
##
```

We have received an accuracy of 99.59 percent, so we can infer that the model has been trained very accurately and will be nicely able to predict the values of our target variable named 'left'. We can see from the confusion matrix that only 10 values of our target have been wrongly classified.

We have received the sensitivity of 1, specificity of 0.98 which show that we have a lot of true positive results from our modelling and our model is very accurate.

We can observe that maximum results belong to the 1st class, that predicted almost negligible fire. Further, the sensitivity, specificity and other parameters for all the classes have been shown. We have taken 95% confidence interval for our analysis and most of our results are within it.

The no information error rate is the error rate when the input and output are independent. So, for our case the value is very low, which indicates good modelling of our data.

The p-value tells us the probability of null hypothesis. A small p-value indicates strong evidence against the null hypothesis, so we reject the null hypothesis.

The Kappa value is a metric that compares an Observed Accuracy with an Expected Accuracy (random chance). The kappa statistic is used not only to evaluate a single classifier, but also to evaluate classifiers amongst themselves. In essence, the kappa statistic is a measure of how closely the instances classified by the machine learning classifier matched the data labeled as ground truth, controlling for the accuracy of a random classifier as measured by the expected accuracy.