Practicum I

Sylvester Brown

Spring 2020

Purpose of project:

Showcase skills in data manipulation and engineering, exploratory data analysis, visualizations and Machine leaning.

The problems to solve are:

what main characteristics contribute to the reason of why employees are leaving?

Which learning model appears to be better for predicting which employees will leave.

About the Dataset:

The dataset belongs to William Walter. Data can be found at "https://www.kaggle.com/colara/human-resource"

Other Resources used:

"Pandas for Everyone' by Daniel Chen

https://www.kaggle.com/colara/human-resources-analytics-a-descriptive-analysis

https://www.kaggle.com/daphnecor/predict-employee-turnover-rate-0

https://www.kaggle.com/henryshtang/hr-data-exploration

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

Libraries

```r
# load the data.table, dplyr, and ggplot2 libraries

library(data.table)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:data.table':
##
##     between, first, last
```

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

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

```r
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following objects are masked from 'package:data.table':
##
##     dcast, melt
```

```r
library(caret)
```

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

```r
library(Boruta)
```

```
## Loading required package: ranger
```

```r
library(gmodels)
library (Hmisc)
```

```
## Loading required package: survival
```

```
##
## Attaching package: 'survival'
```

```
## The following object is masked from 'package:caret':
##
##     cluster
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library (caTools)
library (ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
```

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

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

```
## Loading required package: rpart
```

Upload Data

```
# use read.csv to import data
hr <- read.csv("C:/Users/spbro/OneDrive/Desktop/Human Resources.csv")
```

Eploratory Data Analysis

```
# convert the data to a data table

hr <- as.data.table(hr)

# how many observations and columns are there?

dim(hr)
```

```
## [1] 14999    10
```

```
# Check to see if there are any missing values in our data and checking overall summary

str(hr)
```

```
## Classes 'data.table' and 'data.frame':   14999 obs. of  10 variables:
##  $ ï..satisfaction_level: num  0.38 0.8 0.11 0.72 0.37 0.41 0.1 0.92 0.89 0.42 ...
##  $ last_evaluation      : num  0.53 0.86 0.88 0.87 0.52 0.5 0.77 0.85 1 0.53 ...
##  $ number_project       : int  2 5 7 5 2 2 6 5 5 2 ...
##  $ average_montly_hours : int  157 262 272 223 159 153 247 259 224 142 ...
##  $ time_spend_company   : int  3 6 4 5 3 3 4 5 5 3 ...
##  $ Work_accident        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ left                 : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ promotion_last_5years: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ sales                : Factor w/ 10 levels "accounting","hr",..: 8 8 8 8 8 8 8 8 8 8 ...
##  $ salary               : Factor w/ 3 levels "high","low","medium": 2 3 3 2 2 2 2 2 2 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
summary(hr)
```

```
##  ï..satisfaction_level last_evaluation  number_project  average_montly_hours
##  Min.   :0.0900        Min.   :0.3600   Min.   :2.000   Min.   : 96.0
##  1st Qu.:0.4400        1st Qu.:0.5600   1st Qu.:3.000   1st Qu.:156.0
##  Median :0.6400        Median :0.7200   Median :4.000   Median :200.0
##  Mean   :0.6128        Mean   :0.7161   Mean   :3.803   Mean   :201.1
##  3rd Qu.:0.8200        3rd Qu.:0.8700   3rd Qu.:5.000   3rd Qu.:245.0
##  Max.   :1.0000        Max.   :1.0000   Max.   :7.000   Max.   :310.0
##
##  time_spend_company Work_accident         left        promotion_last_5years
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median : 3.000     Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381   Mean   :0.02127
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##
##          sales         salary
##  sales      :4140   high  :1237
##  technical  :2720   low   :7316
##  support    :2229   medium:6446
##  IT         :1227
##  product_mng: 902
##  marketing  : 858
##  (Other)    :2923
```

Rename Variables

A closer look at the column names shows that some of the colums are not descriptive enough to help the analyst know what the column contains. For this reason the "Sales" column will need to be changed to "departments" and "average_montly_hours" will be changed to "average_monthly_hours". "Work_accidents" change to "work_accidents", "time_spend_company" to "time_spent_at_company", "number_project" to "number_of_projects"

```
#Renaming dataset

hr<-rename(hr, c("satisfaction_level"="ï..satisfaction_level"))
hr<-rename(hr, c("department"="sales"))
hr<-rename(hr, c("average_monthly_hours"="average_montly_hours"))
hr<-rename(hr, c("work_accidents"="Work_accident"))
hr<-rename(hr, c("time_spent_at_company"="time_spend_company"))
hr<-rename(hr, c("number_of_projects"="number_project"))

hr$salary <- as.numeric(1:3)[match(hr$salary, c('low', 'medium', 'high'))]

head(hr) # Display first 5 rows
```

```
##    satisfaction_level last_evaluation number_of_projects average_monthly_hours
## 1:               0.38            0.53                  2                   157
## 2:               0.80            0.86                  5                   262
## 3:               0.11            0.88                  7                   272
## 4:               0.72            0.87                  5                   223
## 5:               0.37            0.52                  2                   159
## 6:               0.41            0.50                  2                   153
##    time_spent_at_company work_accidents left promotion_last_5years department
## 1:                     3              0    1                      0      sales
## 2:                     6              0    1                      0      sales
## 3:                     4              0    1                      0      sales
## 4:                     5              0    1                      0      sales
## 5:                     3              0    1                      0      sales
## 6:                     3              0    1                      0      sales
##    salary
## 1:      1
## 2:      2
## 3:      2
## 4:      1
## 5:      1
## 6:      1
```

```r
turnover<-as.factor(hr$left)
summary(turnover)
```

```
##     0     1
## 11428  3571
```

```r
perc_turnover_rate<-sum(hr$left/length(hr$left))*100
#percentage of turnover
print(perc_turnover_rate)
```

```
## [1] 23.80825
```

```r
# Overview of summary (Turnover V.S. Non-turnover)
cor_vars<-hr[,c("satisfaction_level","last_evaluation","number_of_projects","average_monthly_hours","ti

aggregate(cor_vars[,c("satisfaction_level","last_evaluation","number_of_projects","average_monthly_hours
```

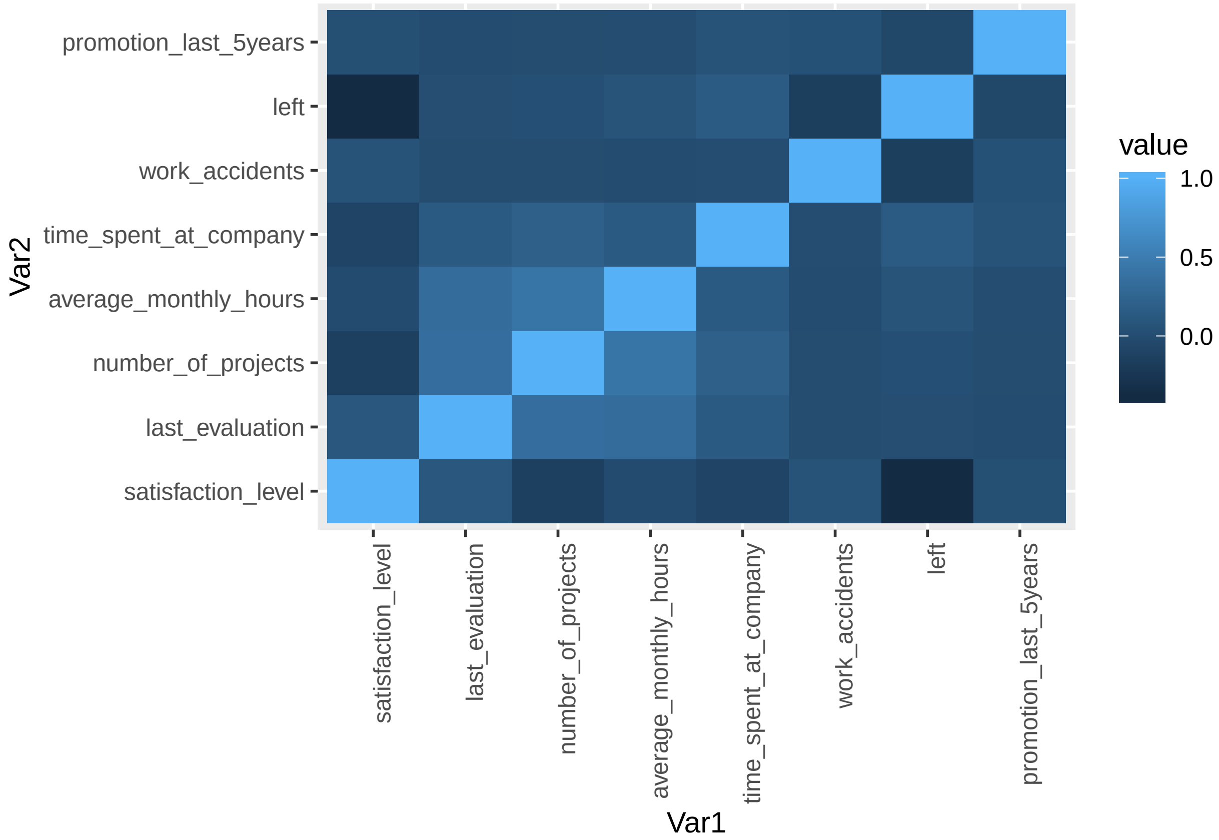
```
##   Category satisfaction_level last_evaluation number_of_projects
## 1        0          0.6668096       0.7154734           3.786664
## 2        1          0.4400980       0.7181126           3.855503
##   average_monthly_hours time_spent_at_company work_accidents
## 1              199.0602              3.380032     0.17500875
## 2              207.4192              3.876505     0.04732568
##   promotion_last_5years
## 1           0.026251313
## 2           0.005320638
```

Correlation Matrix

```
cor_vars<-hr[,c("satisfaction_level","last_evaluation","number_of_projects","average_monthly_hours","ti
cor(cor_vars)
```

```
##                       satisfaction_level last_evaluation number_of_projects
## satisfaction_level            1.00000000     0.105021214       -0.142969586
## last_evaluation               0.10502121     1.000000000        0.349332589
## number_of_projects           -0.14296959     0.349332589        1.000000000
## average_monthly_hours        -0.02004811     0.339741800        0.417210634
## time_spent_at_company        -0.10086607     0.131590722        0.196785891
## work_accidents                0.05869724    -0.007104289       -0.004740548
## left                         -0.38837498     0.006567120        0.023787185
## promotion_last_5years         0.02560519    -0.008683768       -0.006063958
##                       average_monthly_hours time_spent_at_company
## satisfaction_level             -0.020048113          -0.100866073
## last_evaluation                 0.339741800           0.131590722
## number_of_projects              0.417210634           0.196785891
## average_monthly_hours           1.000000000           0.127754910
## time_spent_at_company           0.127754910           1.000000000
## work_accidents                 -0.010142888           0.002120418
## left                            0.071287179           0.144822175
## promotion_last_5years          -0.003544414           0.067432925
##                       work_accidents        left promotion_last_5years
## satisfaction_level       0.058697241 -0.38837498           0.025605186
## last_evaluation         -0.007104289  0.00656712          -0.008683768
## number_of_projects      -0.004740548  0.02378719          -0.006063958
## average_monthly_hours   -0.010142888  0.07128718          -0.003544414
## time_spent_at_company    0.002120418  0.14482217           0.067432925
## work_accidents           1.000000000 -0.15462163           0.039245435
## left                    -0.154621634  1.00000000          -0.061788107
## promotion_last_5years    0.039245435 -0.06178811           1.000000000
```

```
trans<-cor(cor_vars)
melted_cormat <- melt(trans)

ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



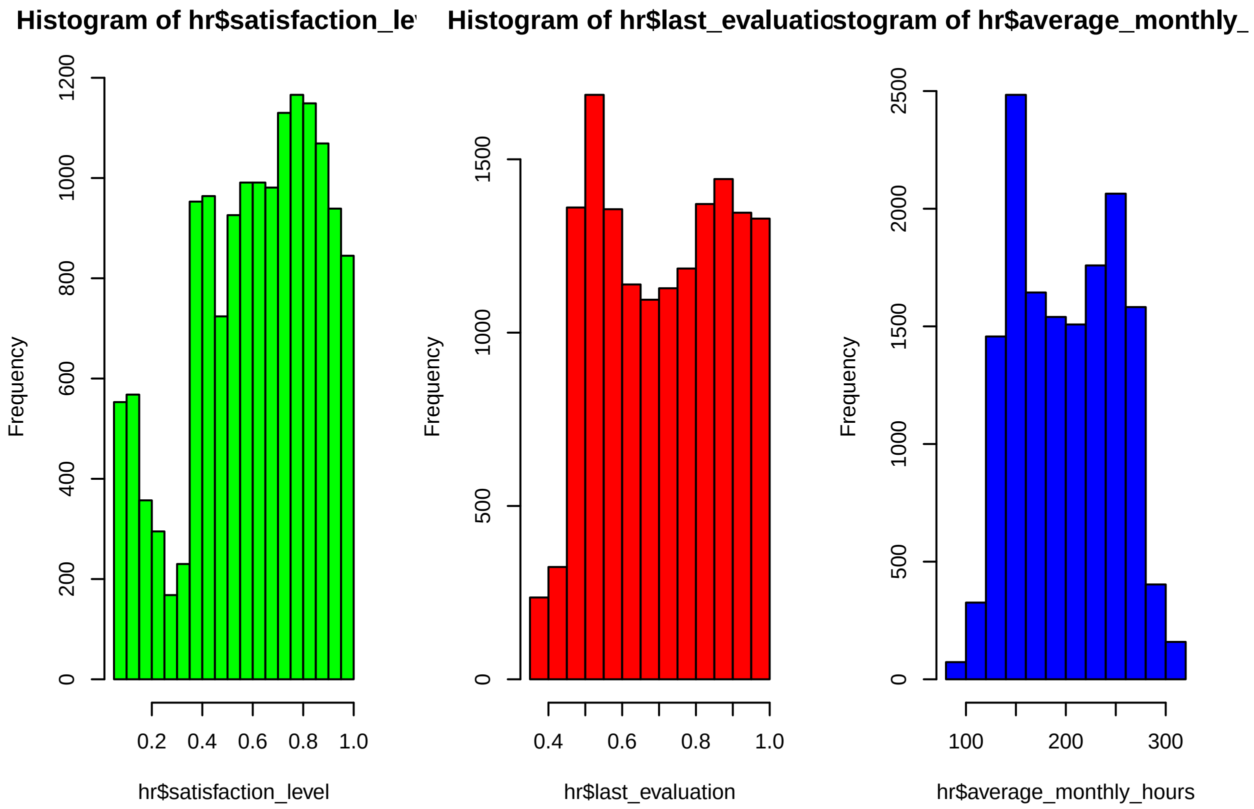
Summary:

From the heatmap, there is a positive correlation between number__project, average__montly__hours, and evaluation. Which appears to indicate that the employees who spent more hours and did more projects were evaluated highly.

For the negative relationships, turnover and satisfaction are highly correlated. This appears to indicate that people tend to leave a company more when they are less satisfied.

Distribution Plots (Satisfaction - Evaluation - AverageMonthlyHours)

```r
# Satisfaction - Evaluation - AverageMonthlyHours
par(mfrow=c(1,3))
hist(hr$satisfaction_level, col="green")
hist(hr$last_evaluation, col="red")
hist(hr$average_monthly_hours, col="blue")
```



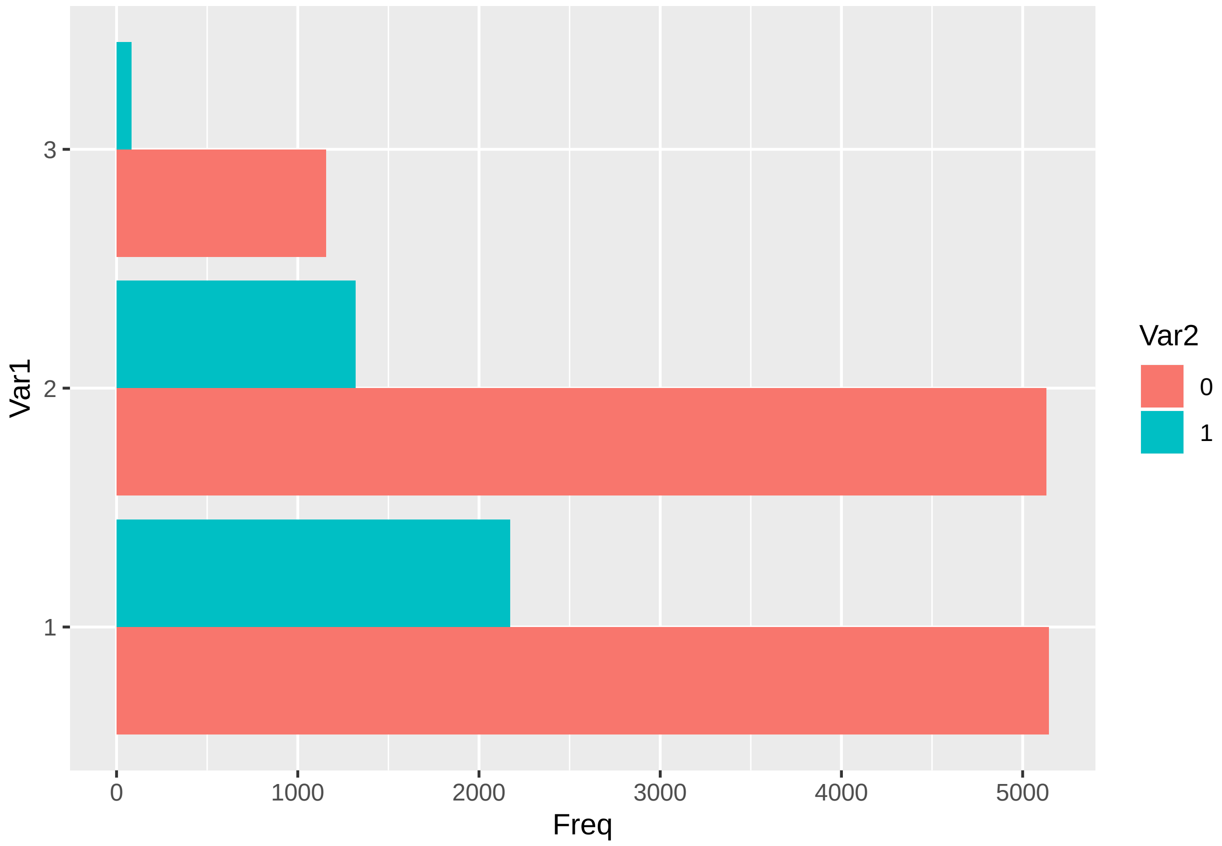
Summary:

- Satisfaction - There is a huge spike for employees with low satisfaction and high satisfaction.
- Evaluation - There is a bimodal distrubtion of employees for low evaluations (less than 0.6) and high evaluations (more than 0.8)
- AverageMonthlyHours - There is another bimodal distribution of employees with lower and higher average monthly hours (less than 150 hours & more than 250 hours)
- The evaluation and average monthly hour graphs both share a similar distribution.
- Employees with lower average monthly hours were evaluated less and vice versa.

```
# Salary V.S. Turnover

vis_1<-table(hr$salary,hr$left)
#print(vis_1)
d_vis_1<-as.data.frame(vis_1)
print(d_vis_1)
```

```
##   Var1 Var2 Freq
## 1    1    0 5144
## 2    2    0 5129
## 3    3    0 1155
## 4    1    1 2172
## 5    2    1 1317
## 6    3    1   82
```

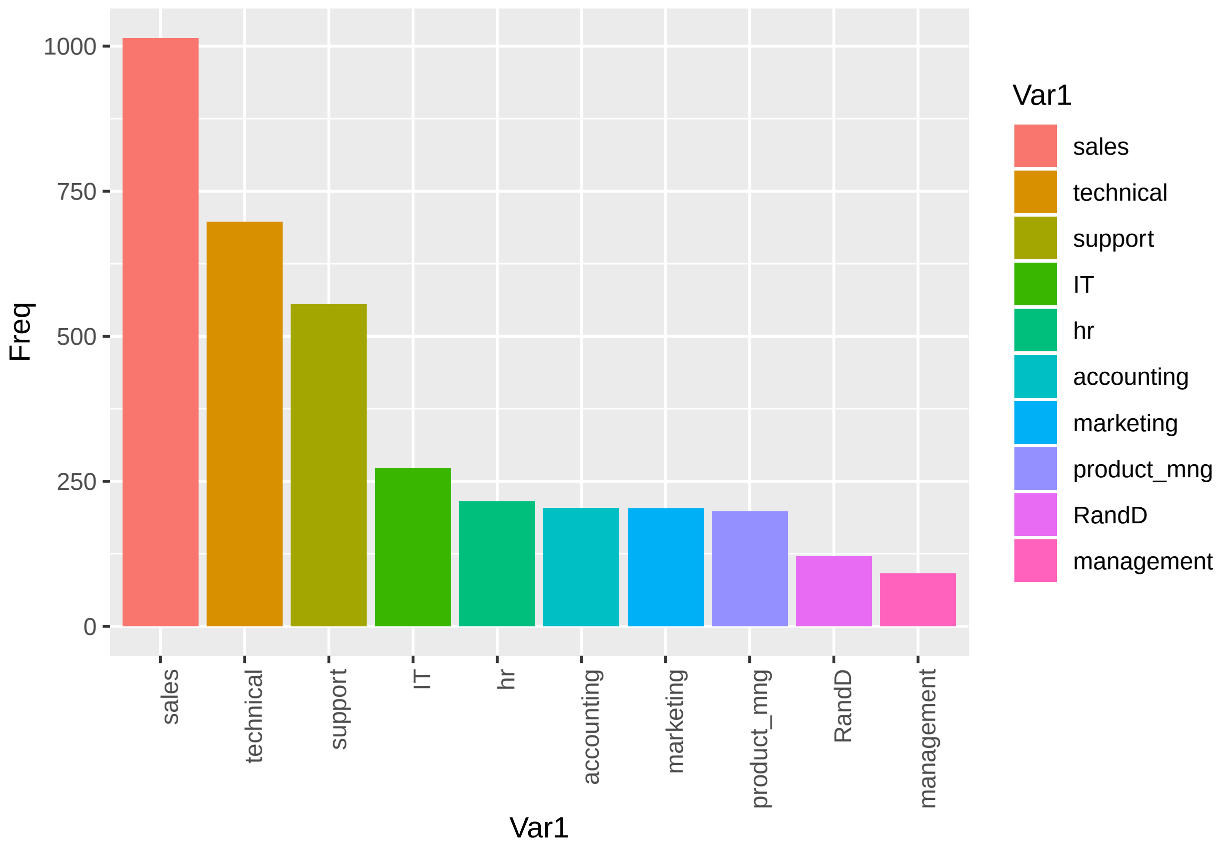
```r
p<-ggplot(d_vis_1, aes(x=Var1,y=Freq,fill=Var2)) +
 geom_bar(position="dodge",stat='identity') + coord_flip()

print(p)
```



Summary:

- Majority of employees who left either had low or medium salary.
- Barely any employees left with high salary

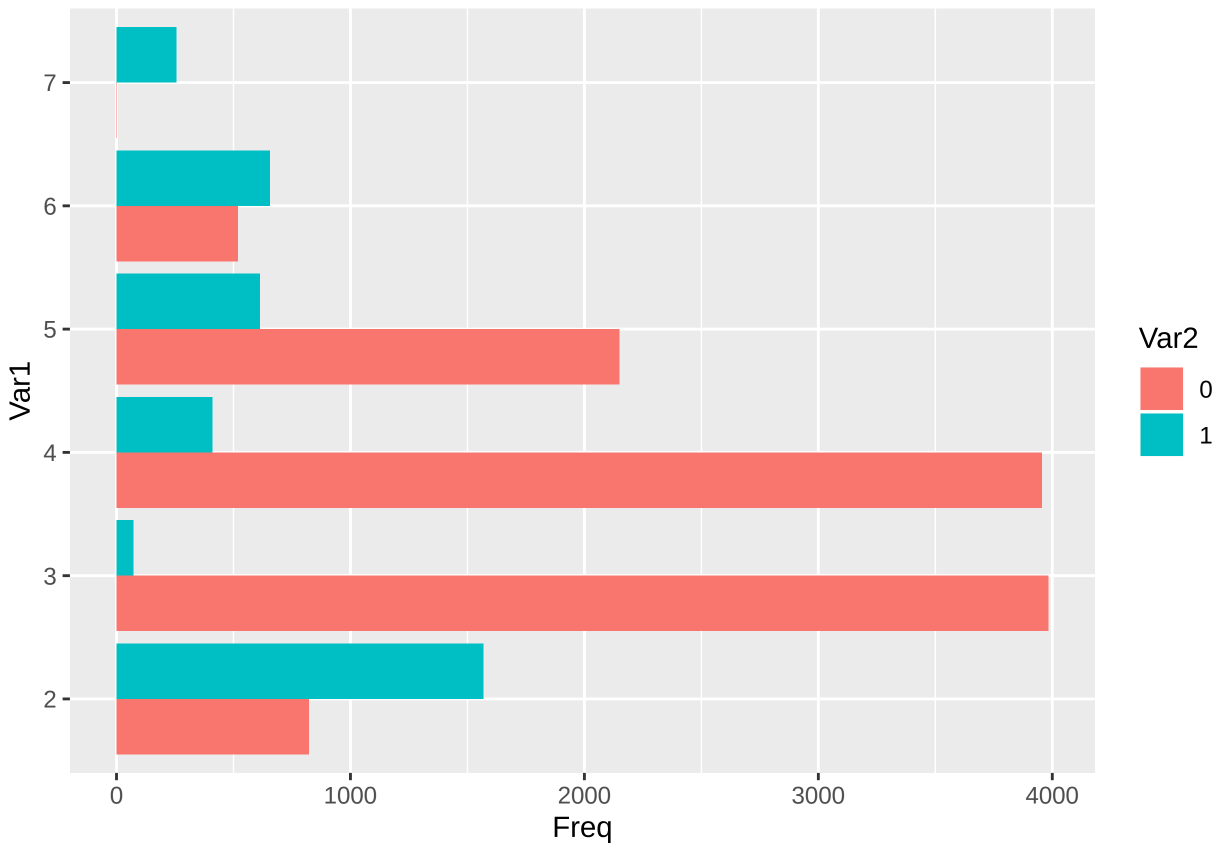
```r
# Department V.S. Turnover

vis_2<-table(hr$department,hr$left)
d_vis_2<-as.data.frame(vis_2)
d_vis_2<-subset(d_vis_2,Var2==1)
#print(d_vis_2)
d_vis_2$Var1 <- factor(d_vis_2$Var1, levels = d_vis_2$Var1[order(-d_vis_2$Freq)])
p<-ggplot(d_vis_2, aes(x=Var1,y=Freq,fill=Var1)) +
 geom_bar(stat='identity') +theme(axis.text.x = element_text(angle = 90, hjust = 1))

print(p)
```



Summary:

- The sales, technical, and support department were the top 3 departments to have employee turnover
- The management department had the smallest amount of turnover

```
#Turnover V.S. ProjectCount

vis_3<-table(hr$number_of_projects,hr$left)
d_vis_3<-as.data.frame(vis_3)
#print(d_vis_1)
p<-ggplot(d_vis_3, aes(x=Var1,y=Freq,fill=Var2)) +
 geom_bar(position="dodge",stat='identity') + coord_flip()

print(p)
```

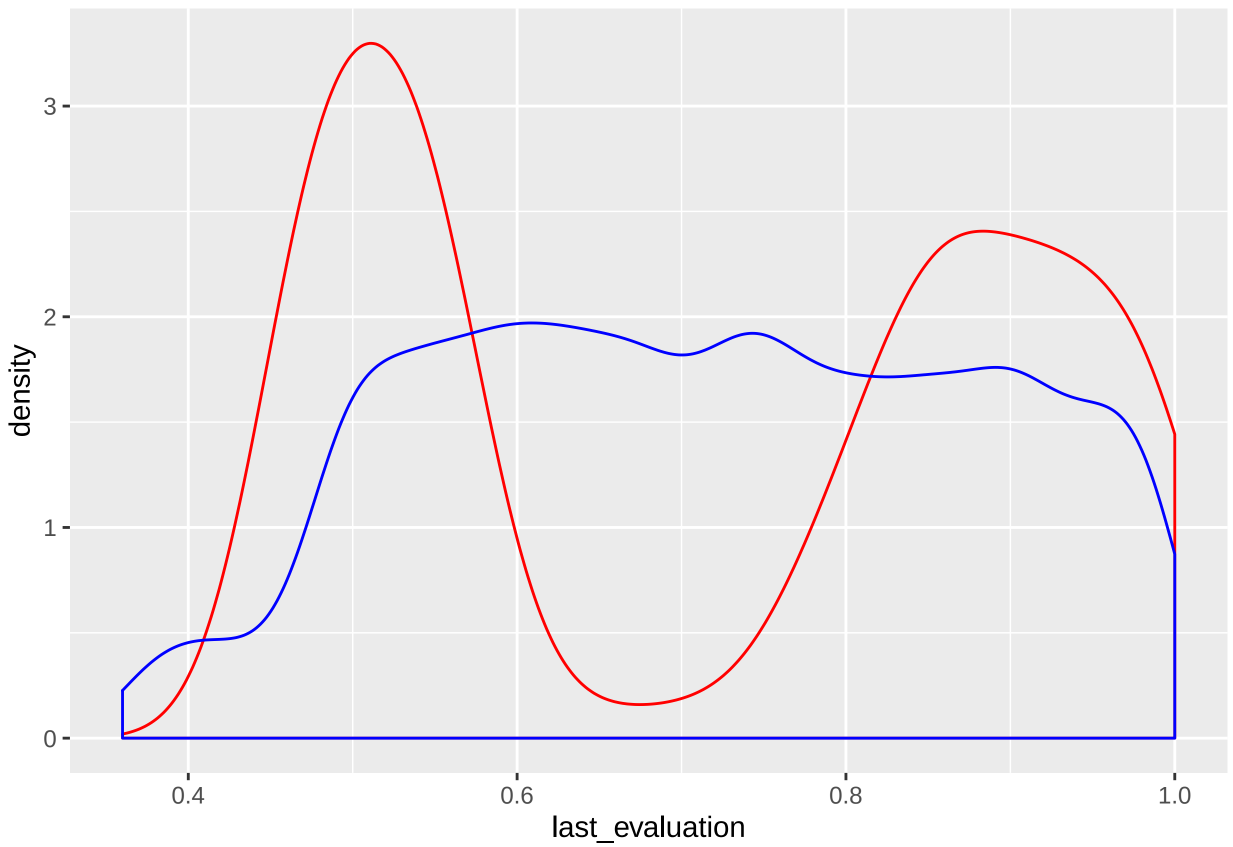


Summary:

- More than half of the employees with 2,6, and 7 projects left the company
- Majority of the employees who did not leave the company had 3,4, and 5 projects
- All of the employees with 7 projects left the company
- There is an increase in employee turnover rate as project count increases

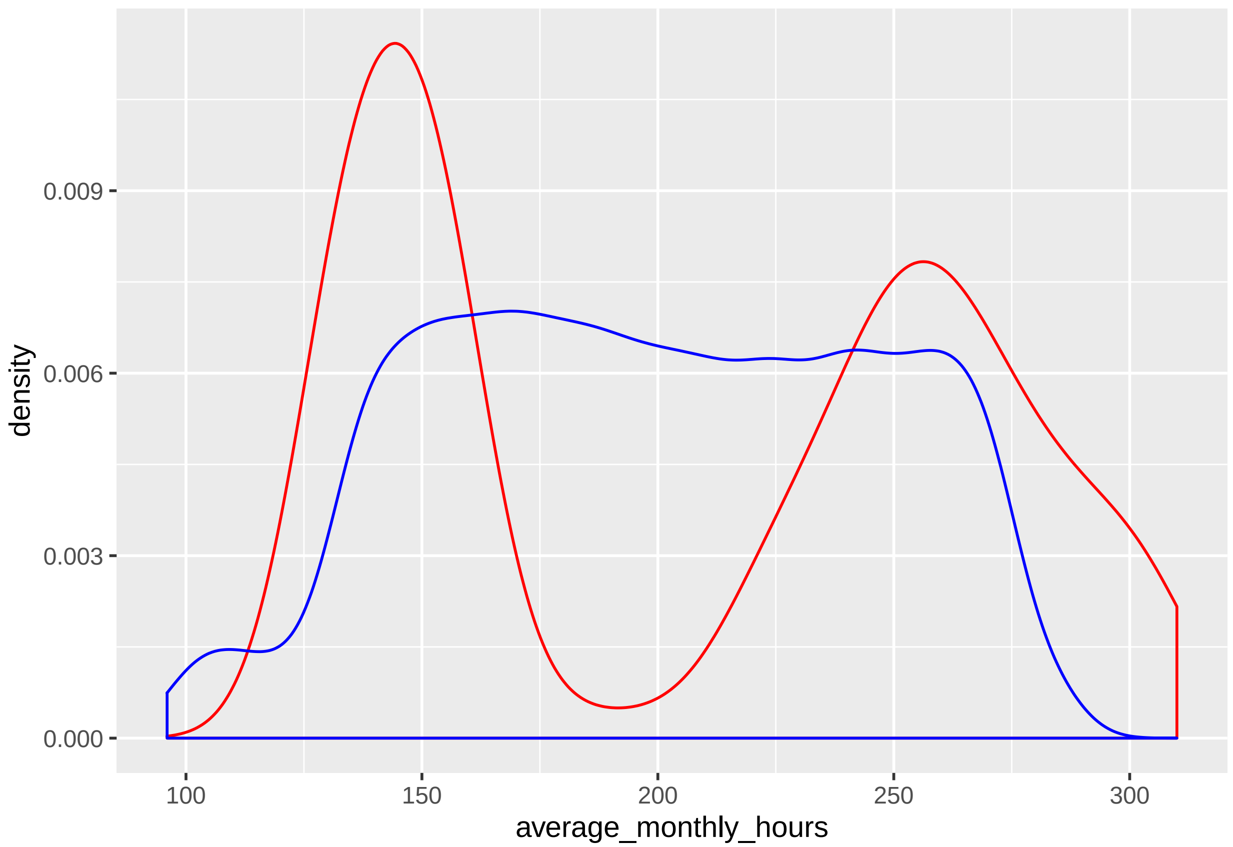
Kernel Density Plot

```
# Kernel Density Plot
left_data<-subset(hr,left==1)
stay_data<-subset(hr,left==0)
ggplot() + geom_density(aes(x=last_evaluation), colour="red", data=left_data) +
  geom_density(aes(x=last_evaluation), colour="blue", data=stay_data)
```



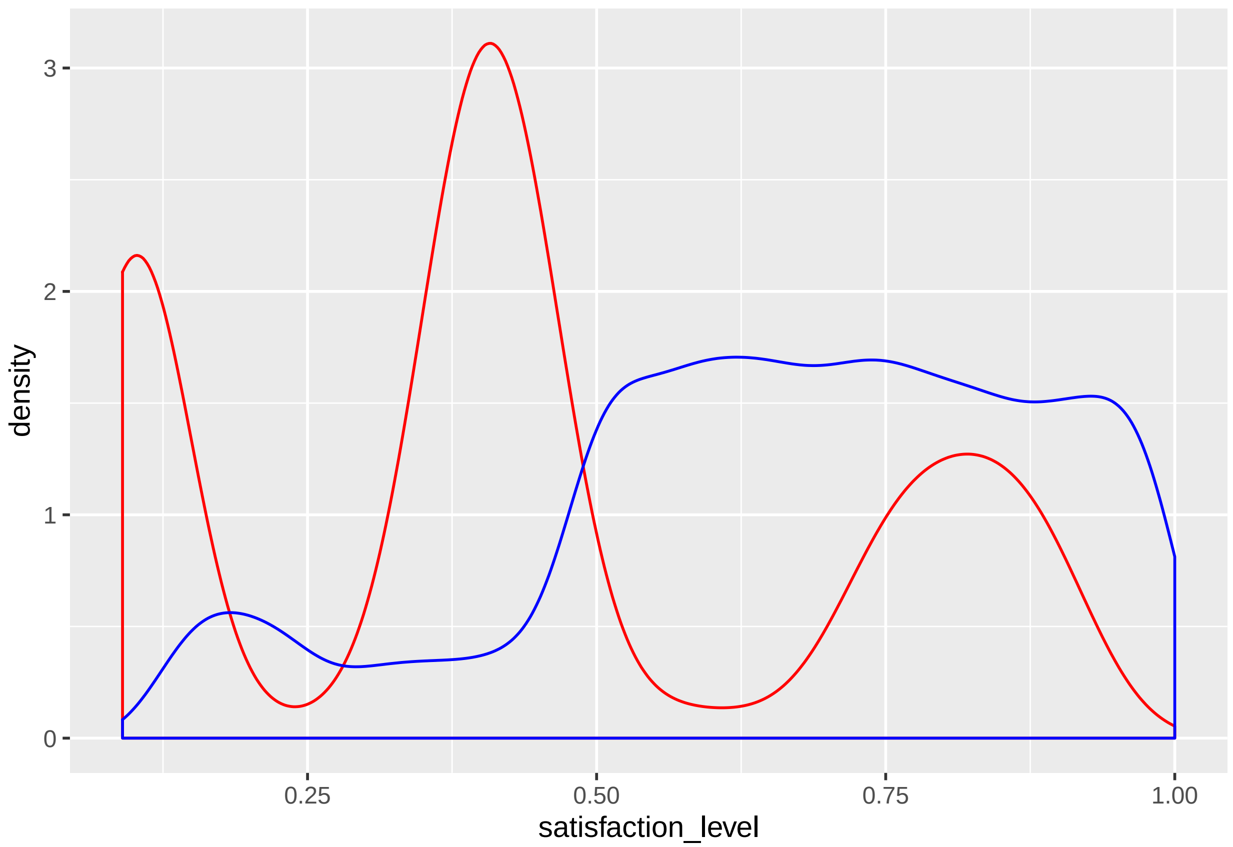
Summary: - There is a biomodal distribution for those that had a turnover. - Employees with low performance tend to leave the company more - Employees with high performance tend to leave the company more - The sweet spot for employees that stayed is within 0.6-0.8 evaluation

```
#KDEPlot: Kernel Density Estimate Plot

ggplot() + geom_density(aes(x=average_monthly_hours), colour="red", data=left_data) +
  geom_density(aes(x=average_monthly_hours), colour="blue", data=stay_data)
```



Summary: - Another bi-modal distribution for employees that turnovered - Employees who had less hours of work (~150hours or less) left the company more - Employees who had too many hours of work (~250 or more) left the company - Employees who left generally were underworked or overworked.

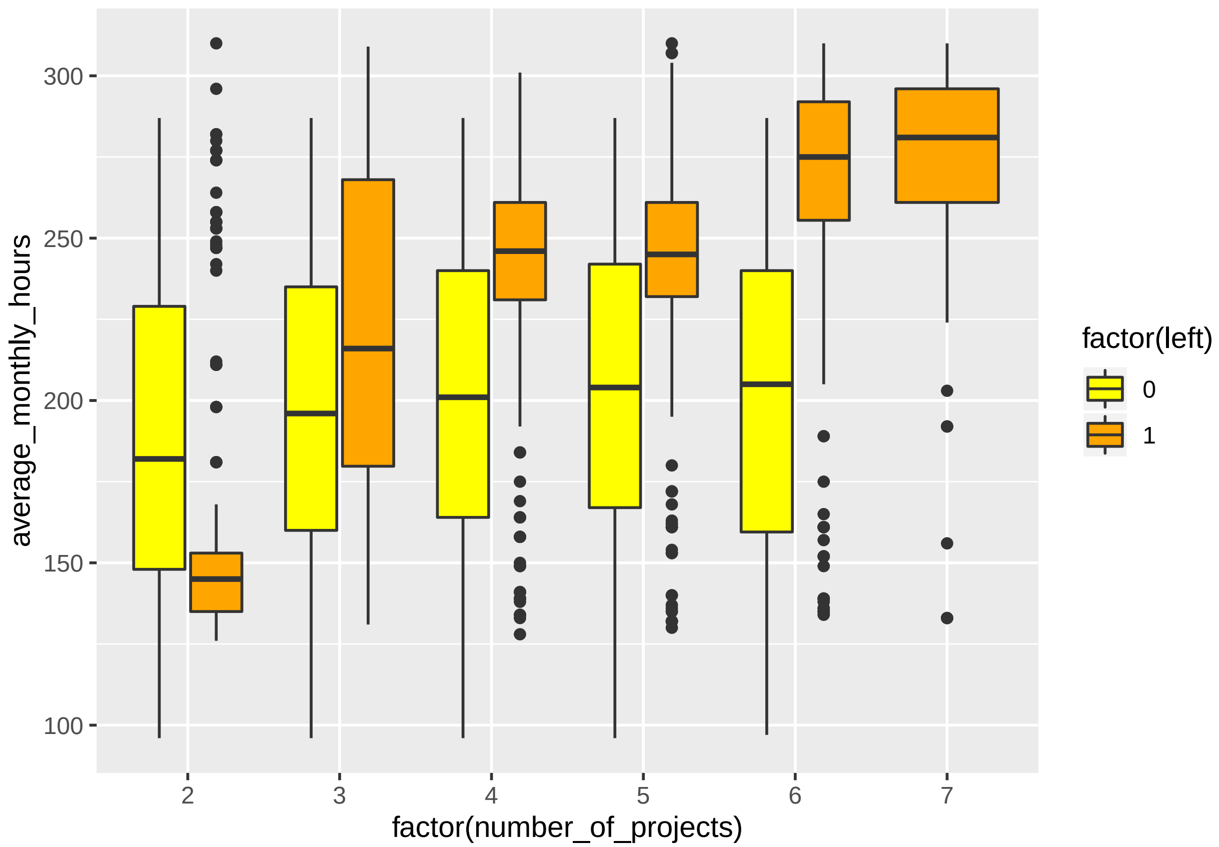
```
#KDEPlot: Kernel Density Estimate Plot
ggplot() + geom_density(aes(x=satisfaction_level), colour="red", data=left_data) +
  geom_density(aes(x=satisfaction_level), colour="blue", data=stay_data)
```



Summary: - There is a tri-modal distribution for employees that turnovered - Employees who had really low satisfaction levels (0.2 or less) left the company more - Employees who had low satisfaction levels (0.3~0.5) left the company more - Employees who had really high satisfaction levels (0.7 or more) left the company more

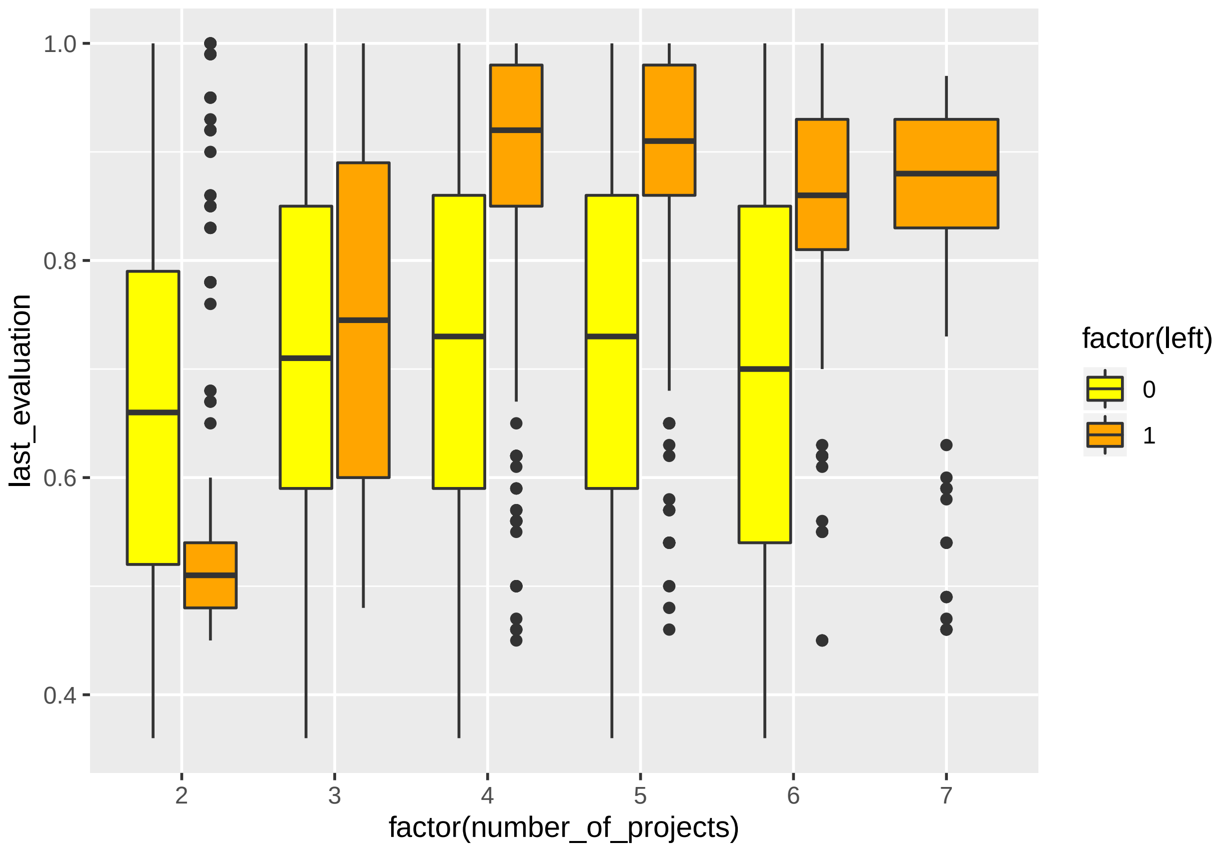
BOXPLOT

```
#ProjectCount VS AverageMonthlyHours [BOXPLOT]

p<-ggplot(hr, aes(x = factor(number_of_projects), y = average_monthly_hours, fill = factor(left))) +
  geom_boxplot() + scale_fill_manual(values = c("yellow", "orange"))
print(p)
```



Summary: - As project count increased, so did average monthly hours - Something weird about the boxplot graph is the difference in averageMonthlyHours between people who had a turnver and did not. - Looks like employees who did not have a turnover had consistent averageMonthlyHours, despite the increase in projects - In contrast, employees who did have a turnover had an increase in averageMonthlyHours with the increase in projects

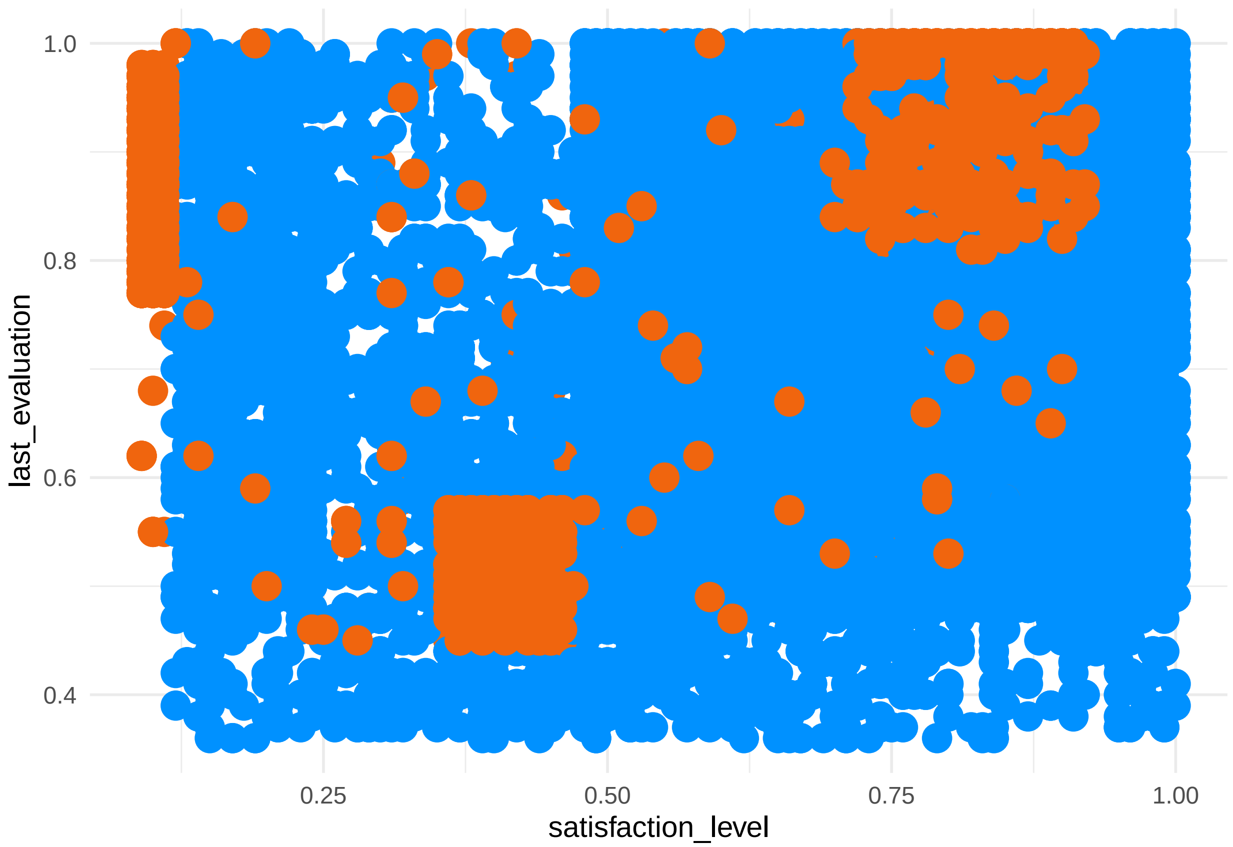
```
#ProjectCount VS Evaluation
#Looks like employees who did not leave the company had an average evaluation of around 70% even with d
#There is a huge skew in employees who had a turnover though. It drastically changes after 3 projectCou
#Employees that had two projects and a horrible evaluation left. Employees with more than 3 projects an

p<-ggplot(hr, aes(x = factor(number_of_projects), y = last_evaluation, fill = factor(left))) +
  geom_boxplot() + scale_fill_manual(values = c("yellow", "orange"))
print(p)
```



Summary: Looks like employees who did not leave the company had an average evaluation of around 70% even with different projectCounts There is a huge skew in employees who had a turnover though. It drastically changes after 3 projectCounts. Employees that had two projects and a horrible evaluation left. Employees with more than 3 projects and super high evaluations left. What I find strange with this graph is with the turnover group. There is an increase in evaluation for employees who did more projects within the turnover group. But, again for the non-turnover group, employees here had a consistent evaluation score despite the increase in project counts.

Satisfaction VS Evaluation

```
ggplot(hr, aes(satisfaction_level, last_evaluation, color = left)) +
  geom_point(shape = 16, size = 5, show.legend = FALSE) +
  theme_minimal() +
  scale_color_gradient(low = "#0091ff", high = "#f0650e")
```



Summary:

There are 3 distinct clusters for employees who left the company

- Cluster 1 (Hard-working and Sad Employee): Satisfaction was below 0.2 and evaluations were greater than 0.75. Which could be a good indication that employees who left the company were good workers but felt horrible at their job.
- Cluster 2 (Bad and Sad Employee): Satisfaction between about 0.35~0.45 and evaluations below ~0.58. This could be seen as employees who were badly evaluated and felt bad at work.
- Cluster 3 (Hard-working and Happy Employee): Satisfaction between 0.7~1.0 and evaluations were greater than 0.8. Which could mean that employees in this cluster were "ideal". They loved their work and were evaluated highly for their performance.

Feature Importance selection using BORUTA

Boruta is a feature selection algorithm. Precisely, it works as a wrapper algorithm around Random Forest. This package derive its name from a demon in Slavic mythology who dwelled in pine forests. Feature selection is a crucial step in predictive modeling. This technique achieves supreme importance when a data set comprised of several variables is given for model building.

Boruta can be your algorithm of choice to deal with such data sets. Particularly when one is interested in understanding the mechanisms related to the variable of interest, rather than just building a black box predictive model with good prediction accuracy.

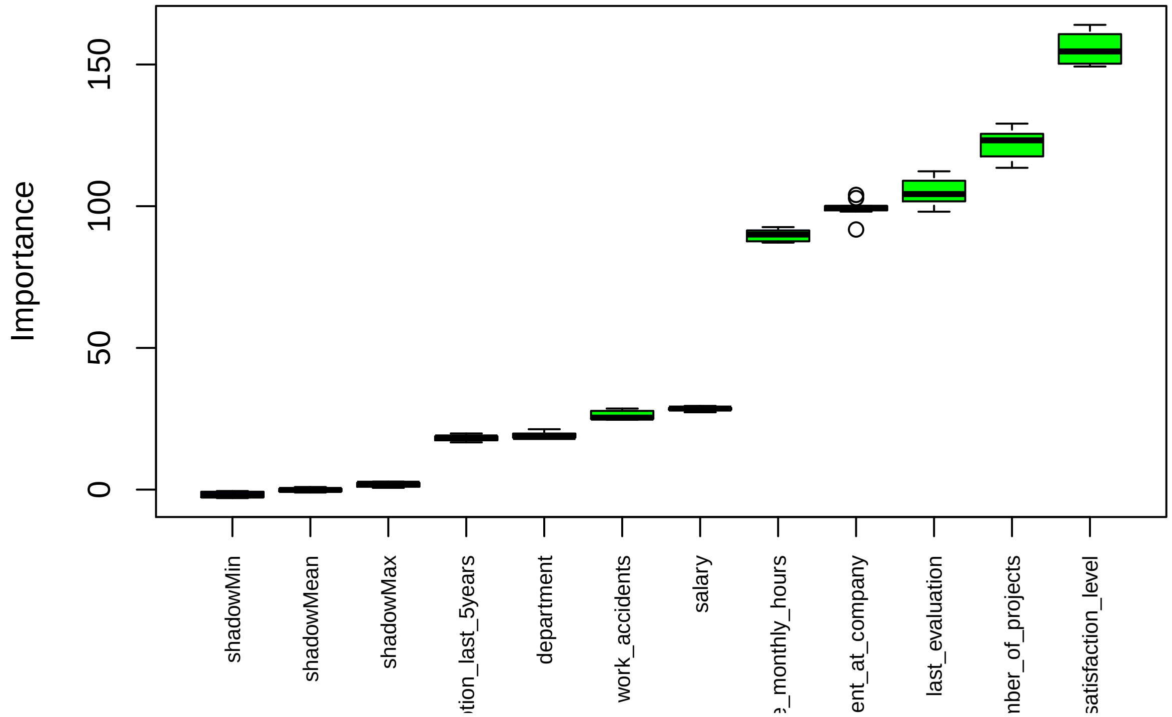
```r
hr$left<-as.factor(hr$left)
boruta.train <- Boruta(left~., data = hr, doTrace = 2)
```

```
##  1. run of importance source...

##  2. run of importance source...

##  3. run of importance source...

##  4. run of importance source...

##  5. run of importance source...

##  6. run of importance source...

##  7. run of importance source...

##  8. run of importance source...

##  9. run of importance source...

##  10. run of importance source...

## After 10 iterations, +26 secs:

##  confirmed 9 attributes: average_monthly_hours, department, last_evaluation, number_of_projects, prom

##  no more attributes left.
```

```r
print(boruta.train)
```

```
## Boruta performed 10 iterations in 25.95373 secs.
##  9 attributes confirmed important: average_monthly_hours, department,
## last_evaluation, number_of_projects, promotion_last_5years and 4 more;
##  No attributes deemed unimportant.
```

```r
plot(boruta.train, xlab = "", xaxt = "n")

lz<-lapply(1:ncol(boruta.train$ImpHistory),function(i)
boruta.train$ImpHistory[is.finite(boruta.train$ImpHistory[,i]),i])
names(lz) <- colnames(boruta.train$ImpHistory)
Labels <- sort(sapply(lz,median))
axis(side = 1,las=2,labels = names(Labels),
at = 1:ncol(boruta.train$ImpHistory), cex.axis = 0.7)
```



Key Observations: The above graph clearly represents the factors which serve as the top reasons for emplpoyee who left the company:

- Satisfaction level: it already had a negative corellation with the outcome. People with low satisfaction were most likely to leave even when compared with evaluations(Evident cluster was formed with respect to low satisfaction)
- Salary and the role they played has one of the least impact on attrition
- Pressure due to the number of projects and how they were evaluated also holds key significance in determining attrition

DATA MODELING OR MACHINE LEARNING

Logistic Regression Analysis

```
#Creating training and test sets for the logistic regression
smp_size <- floor(0.75 * nrow(hr))

## set the seed to make your partition reproductible
set.seed(123)
train_ind <- sample(seq_len(nrow(hr)), size = smp_size)

train <- hr[train_ind, ]
test <- hr[-train_ind, ]

dim(test)
```

```
## [1] 3750   10
```

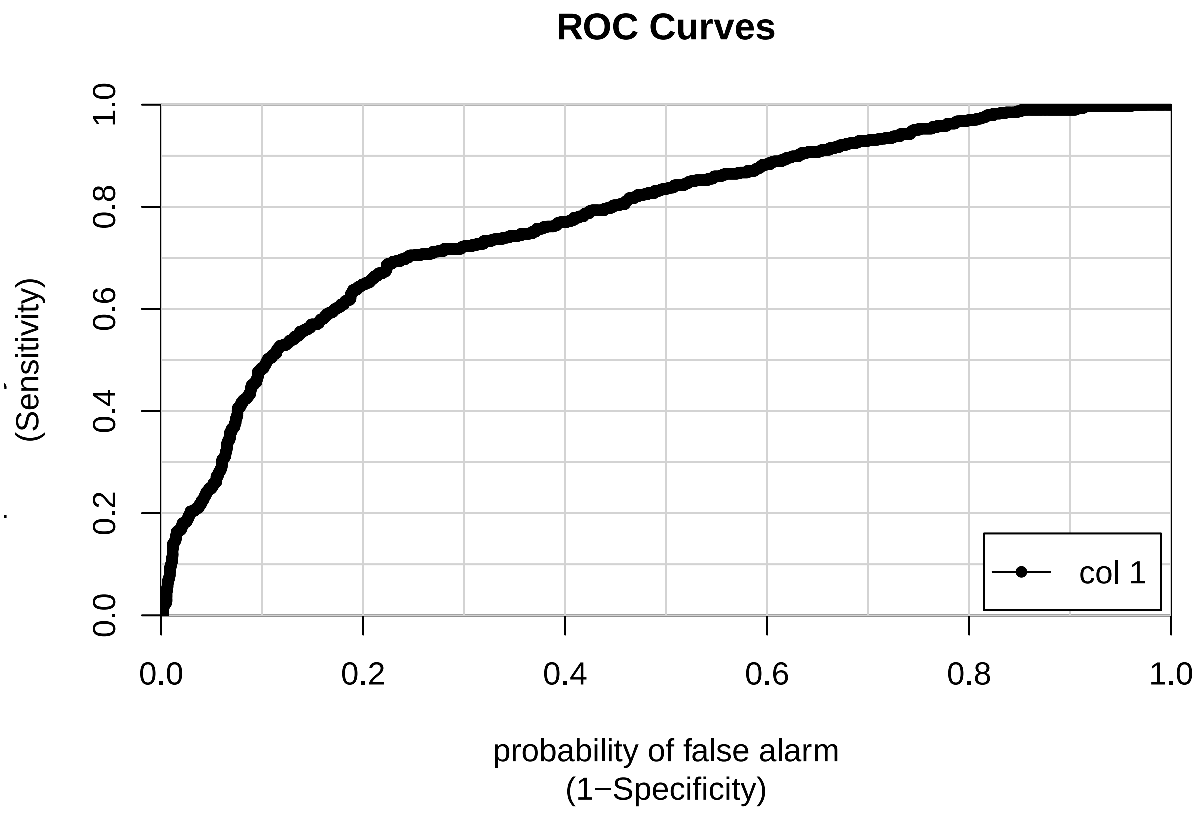
```
dim(train)
```

```
## [1] 11249    10
```

```
#Training the model


logit_model<-glm(left~satisfaction_level+last_evaluation+average_monthly_hours+salary+department+number_

summary(logit_model)
```

```
##
## Call:
## glm(formula = left ~ satisfaction_level + last_evaluation + average_monthly_hours +
##     salary + department + number_of_projects, family = binomial(),
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9239  -0.6874  -0.4562  -0.2180   2.6502
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.5734835  0.1772972   8.875  < 2e-16 ***
## satisfaction_level    -4.1968372  0.1112847 -37.713  < 2e-16 ***
## last_evaluation        0.9230620  0.1666772   5.538 3.06e-08 ***
## average_monthly_hours  0.0049401  0.0005742   8.603  < 2e-16 ***
## salary                -0.6125474  0.0423526 -14.463  < 2e-16 ***
## departmenthr           0.1838508  0.1469440   1.251  0.21088
## departmentIT          -0.1971932  0.1362658  -1.447  0.14786
## departmentmanagement  -0.4675557  0.1756244  -2.662  0.00776 **
## departmentmarketing   -0.1650042  0.1489049  -1.108  0.26781
## departmentproduct_mng -0.2500995  0.1475145  -1.695  0.09000 .
## departmentRandD       -0.6738925  0.1639978  -4.109 3.97e-05 ***
## departmentsales       -0.0605827  0.1146123  -0.529  0.59709
## departmentsupport     -0.0742057  0.1226838  -0.605  0.54528
## departmenttechnical    0.0010061  0.1194927   0.008  0.99328
## number_of_projects    -0.2722031  0.0235576 -11.555  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 12289  on 11248  degrees of freedom
## Residual deviance: 10145  on 11234  degrees of freedom
## AIC: 10175
##
## Number of Fisher Scoring iterations: 5
```

```
test$logit_model<-predict(logit_model,test)
#head(test)

colAUC(test$logit_model,test$left, plotROC=TRUE)
```



```
##                [,1]
## 0 vs. 1 0.7782613
```

```
#An approach to identify the cut-off for the predicted probabilities
#is to use a binned table of the proabablities. See the exact threshold
#where 0's and 1's are getting classified with high precision and recall
#you can use the two commented lines below to get the threshold manually
#test$logit_model_bin <- cut2(test$logit_model,g=12)

#CrossTable(test$left, test$logit_model_bin,prop.chisq=FALSE,prop.r=FALSE,prop.t=FALSE)

#Now using that threshold created the predicted values for each record
test$prediction<-ifelse(test$logit_model>=-.95,1,0)

#Make use of the confusion matrix to calculate accuracy, precision and recall
#CrossTable(test$left, test$prediction,prop.chisq=FALSE,prop.r=FALSE,prop.t=FALSE)
conf_mat<-table(test$left,test$prediction)

#print(conf_mat)
#class(conf_mat)
```

```
accuracy<-(conf_mat[1,1]+conf_mat[2,2])/(conf_mat[1,1]+conf_mat[2,2]+conf_mat[1,2]+conf_mat[2,1])
recall<-(conf_mat[2,2])/(conf_mat[1,2]+conf_mat[2,2])
precision<-(conf_mat[2,2])/(conf_mat[2,2]+conf_mat[2,1])

print(c("Accuracy:",accuracy))
```

```
## [1] "Accuracy:"         "0.753866666666667"
```

```
print(c("Precision:",precision))
```

```
## [1] "Precision:"        "0.687363834422658"
```

```
print(c("Recall:",recall))
```

```
## [1] "Recall:"          "0.49802683504341"
```

```
#red <- prediction(test$prediction, test$left)
#P.perf <- performance(pred, "prec", "rec")
#lot (RP.perf)
```

Fold Cross Validation for Logistic Regression

```
# define training control
train_control <- trainControl(method="cv", number=10)
train$left<-as.factor(train$left)
# fix the parameters of the algorithm
grid <- expand.grid()
# train the model
model <- train(left~., data=train, trControl=train_control, method="glm",family=binomial())
#model <- train(left~satisfaction_level+last_evaluation+number_project+exp_in_company+average_montly_ho
# summarize results
print(model)
```

```
## Generalized Linear Model
##
## 11249 samples
##     9 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 10123, 10124, 10124, 10123, 10124, 10125, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.792249  0.3270712
```

Logistic Regression V.S. Random Forest V.S. Decision Tree V.S. AdaBoost Model

```
# NOTE: By adding in "class_weight = balanced", the Logistic Auc increased by about 10%! This adjusts t


# Decision Tree Model
library(rpart.plot)
trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(3333)
dtree_fit <- train(left ~., data = train, method = "rpart",
                   parms = list(split = "information"),
                   trControl=trctrl,
                   tuneLength = 10)
print(dtree_fit)
```

```
## CART
##
## 11249 samples
##     9 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 10123, 10124, 10124, 10125, 10124, 10125, ...
## Resampling results across tuning parameters:
##
##   cp           Accuracy   Kappa
##   0.004523181  0.9754048  0.9299500
##   0.005653977  0.9725006  0.9220481
##   0.007915567  0.9708410  0.9176045
##   0.009800226  0.9691223  0.9131300
##   0.016584998  0.9671073  0.9077632
##   0.033923860  0.9596996  0.8883828
##   0.052393517  0.9496248  0.8631413
##   0.076140219  0.9252695  0.7817966
##   0.186392763  0.8561104  0.6062652
##   0.241236336  0.7901124  0.2587873
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was cp = 0.004523181.
```

```
#plot decision tree
#p<-prp(dtree_fit$finalModel, box.palette = "Reds", tweak = 1.2)
#print(p)
```

```
# Random Forest Model

train$left<-as.factor(train$left)


ctrl = trainControl(method="repeatedcv", number=10, repeats=5, selectionFunction = "oneSE")


rf_model<-train(left~.,data=train,method="rf",
               trControl=ctrl,
               prox=TRUE,allowParallel=TRUE)
print("random forest")
```

```
## [1] "random forest"
```

```
print(rf_model)
```

```
## Random Forest
##
## 11249 samples
##     9 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 10124, 10125, 10124, 10125, 10123, 10125, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.9742554  0.9261248
##    9    0.9907190  0.9739650
##   17    0.9890120  0.9692626
##
## Accuracy was used to select the optimal model using  the one SE rule.
## The final value used for the model was mtry = 9.
```

Modeling the Data

The best model performance out of the four (Decision Tree Model, AdaBoost Model, Logistic Regression Model, Random Forest Model) was Random Forest!

Summary:

With all of this information, this is what Bob should know about his company and why his employees probably left: 1. Employees generally left when they are underworked (less than 150hr/month or 6hr/day) 2. Employees generally left when they are overworked (more than 250hr/month or 10hr/day) 3. Employees with either really high or low evaluations should be taken into consideration for high turnover rate 4. Employees with low to medium salaries are the bulk of employee turnover 5. Employees that had 2,6, or 7 project count was at risk of leaving the company 6. Employee satisfaction is the highest indicator for employee turnover. 7. Employee that had 4 and 5 yearsAtCompany should be taken into consideration for high turnover rate

Recommendation:

Satisfaction level is the major impact on whether employees stay or leave the company. Improve work life balance by having the right number of projects. Employees with 3-4 projects assigned tend to stay. Similarly, number of average hours a month plays a role in employees leaving or staying. Provide training so that their evaluation score can improve. The data shows that employees with a low evaluation score are likely to leave.