# Human Resources Analytics - Exploratory Analysis

*Data Analysis Team*

*27 November 2017*

## Content



# 1 - Load the data

#### Load the Human Resources Analytics database

```
library(readr)
db <- read_csv("~/Human_Resources_Analytics-Kaggle_DS/final_files/databases/HR_comma_sep_v2.csv") # Rem
```

# 2 - Data cleaning

Check for any missing values in the dataset:

```
sapply(db, function(x) sum(is.na(x)))
```

```
##    satisfaction_level       last_evaluation        number_project
##                     0                     0                     0
##  average_montly_hours    time_spend_company         Work_accident
##                     0                     0                     0
##                  left promotion_last_5years            department
##                     0                     0                     0
##                salary     projects_per_year
##                     0                     0
```

Now we have confirmed that there are no missing values in our variables.

# 3 - Data exploration

## 3.1 - Data overview

```r
# This dataset has about 15,000 employees and 10 variables
dim(db)
```

```
## [1] 14999    11
```

```r
# The following displays the type of each variable:
str(db)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    14999 obs. of  11 variables:
##  $ 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 ...
##  $ department           : chr  "sales" "sales" "sales" "sales" ...
##  $ salary               : chr  "low" "medium" "medium" "low" ...
##  $ projects_per_year    : num  0.667 0.833 1.75 1 0.667 ...
##  - attr(*, "spec")=List of 2
##   ..$ cols   :List of 11
##   .. ..$ satisfaction_level   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ last_evaluation      : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   .. ..$ number_project       : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ average_montly_hours : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ time_spend_company   : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ Work_accident        : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ left                 : list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ promotion_last_5years: list()
##   .. .. ..- attr(*, "class")= chr  "collector_integer" "collector"
##   .. ..$ department           : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ salary               : list()
##   .. .. ..- attr(*, "class")= chr  "collector_character" "collector"
##   .. ..$ projects_per_year    : list()
##   .. .. ..- attr(*, "class")= chr  "collector_double" "collector"
##   ..$ default: list()
##   .. ..- attr(*, "class")= chr  "collector_guess" "collector"
##   ..- attr(*, "class")= chr "col_spec"
```

```r
# Below we can see the statistical summary of each variable:
summary(db)
```

```
##  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
##  Min.   : 2.000     Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 3.000     1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 3.000     Median :0.0000   Median :0.0000
##  Mean   : 3.498     Mean   :0.1446   Mean   :0.2381
##  3rd Qu.: 4.000     3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :10.000     Max.   :1.0000   Max.   :1.0000
##  promotion_last_5years  department           salary
##  Min.   :0.00000       Length:14999       Length:14999
##  1st Qu.:0.00000       Class :character   Class :character
##  Median :0.00000       Mode  :character   Mode  :character
##  Mean   :0.02127
##  3rd Qu.:0.00000
##  Max.   :1.00000
##  projects_per_year
##  Min.   :0.200
##  1st Qu.:0.800
##  Median :1.000
##  Mean   :1.212
##  3rd Qu.:1.500
##  Max.   :3.500
```

How many leavers are there?

```r
table(db$left) # number employees for each level
```

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

```r
prop.table(table(db$left)) # proportion of employees for each level
```

```
##
##         0         1
## 0.7619175 0.2380825
```

```r
round(prop.table(table(db$left)),2) # rounded proportion of employees for each level
```

```
##
##    0    1
## 0.76 0.24
```

In this dataset there are 3,571 leavers and 11,428 stayers. The turnover rate is 24% and the retention rate is 76%.

The below table displays a summary of the variables splitting by Leavers vs Stayers:

```r
cor_vars <- db[,c("satisfaction_level","last_evaluation","number_project","average_montly_hours","time_s
aggregate(cor_vars[,c("satisfaction_level","last_evaluation","number_project","average_montly_hours","ti
```

```
##   Category satisfaction_level last_evaluation number_project
```

```
## 1        0           0.6668096       0.7154734       3.786664
## 2        1           0.4400980       0.7181126       3.855503
##   average_montly_hours time_spend_company Work_accident
## 1             199.0602           3.380032    0.17500875
## 2             207.4192           3.876505    0.04732568
##   promotion_last_5years projects_per_year
## 1           0.026251313         1.2850422
## 2           0.005320638         0.9787408
```

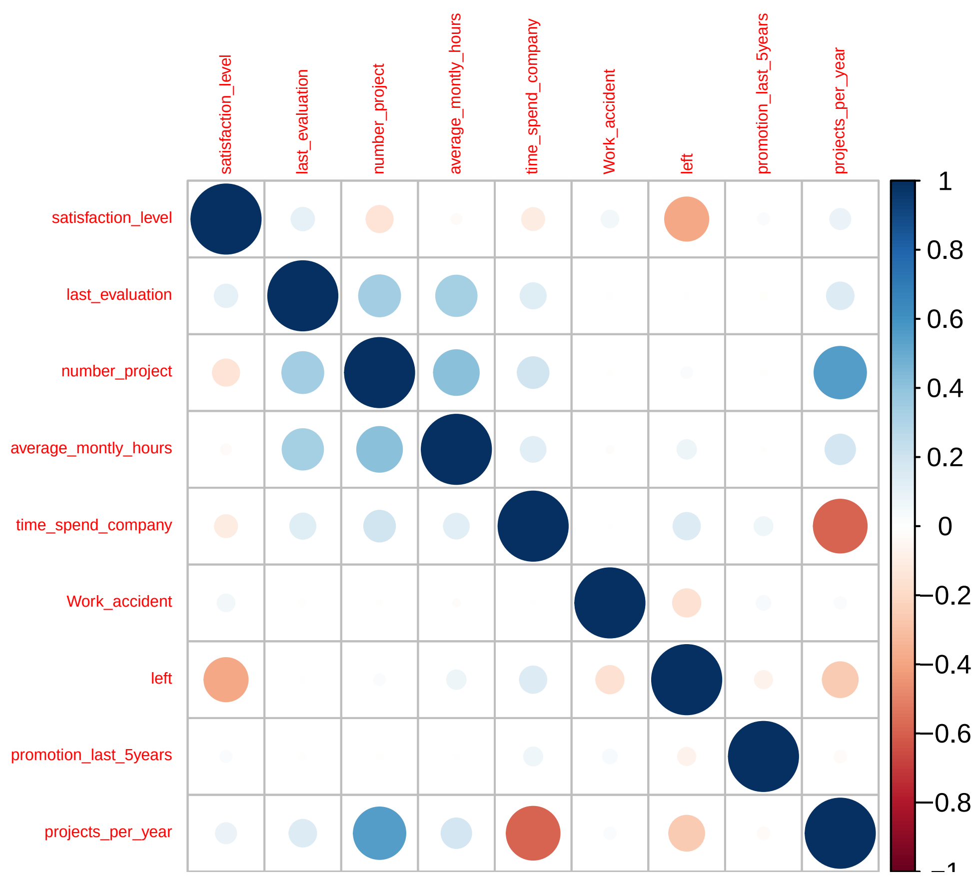
Key higlights: 1 - The turnover rate is 24% 2 - There are approximately 15k employees and 10 variables 3 - The satisfaction level is 61% 4 - Leavers show less satisfaction level, higher monthly hours, higher tenure, lower work accidents, less promotions and less projects per year

## 3.2 - Correlation matrix

```r
# The below code will install the corrplot package if it doesn't exist, and then load it
if (!require(corrplot)) install.packages("corrplot")
library(corrplot)
```

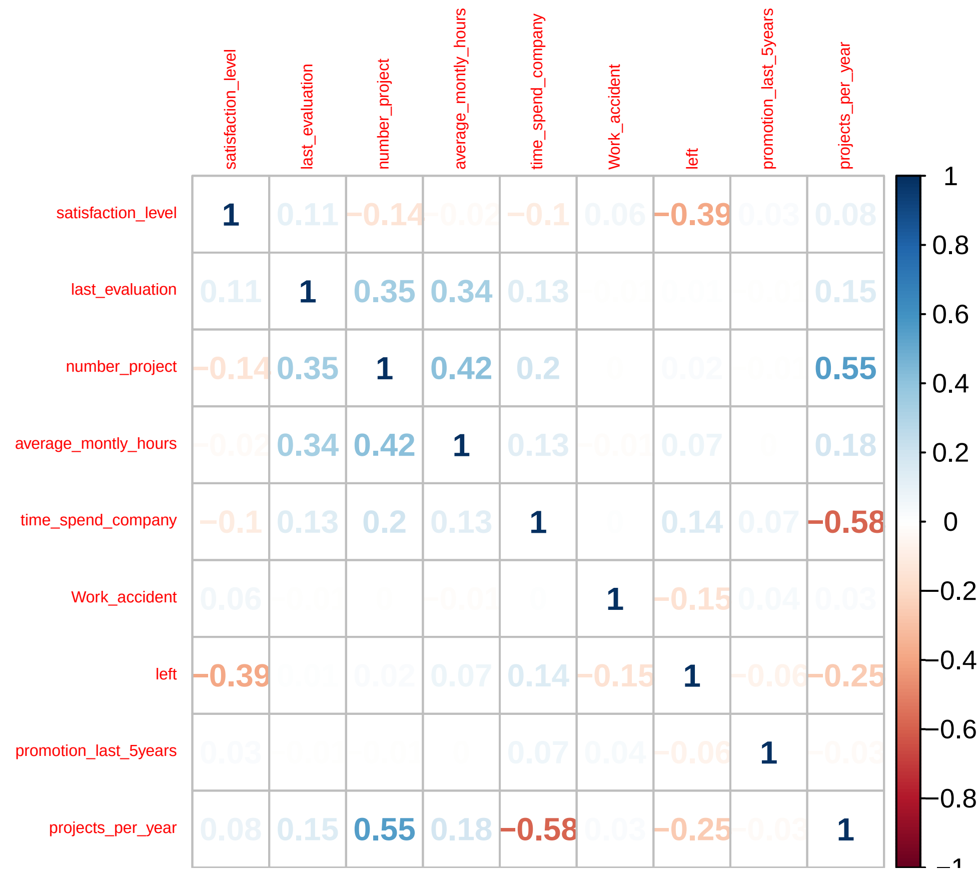
Visual correlation matrix:

```r
cor_vars_summary <- cor(cor_vars)

corrplot(cor_vars_summary,method="circle",tl.cex=0.5)
```



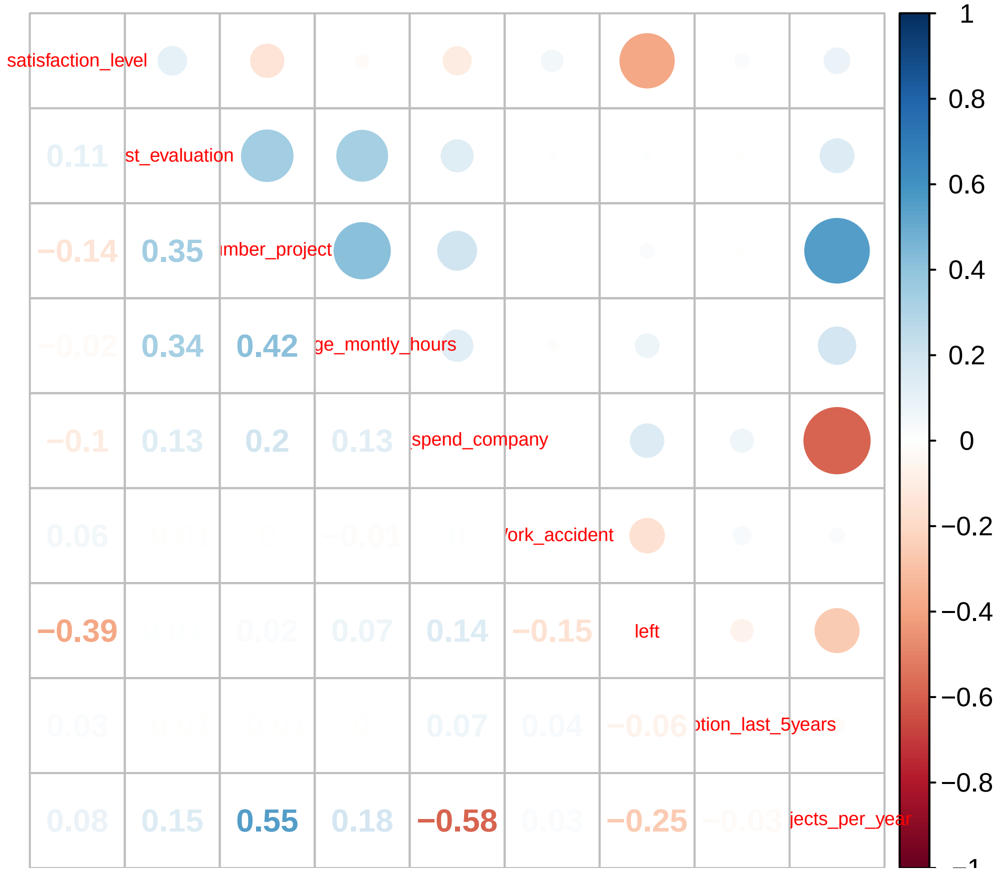

Tabular correlation matrix:

```
corrplot(cor_vars_summary,method="number",tl.cex=0.5)
```



Mixed correlation matrix:

```
corrplot.mixed(cor_vars_summary,tl.cex=0.6)
```

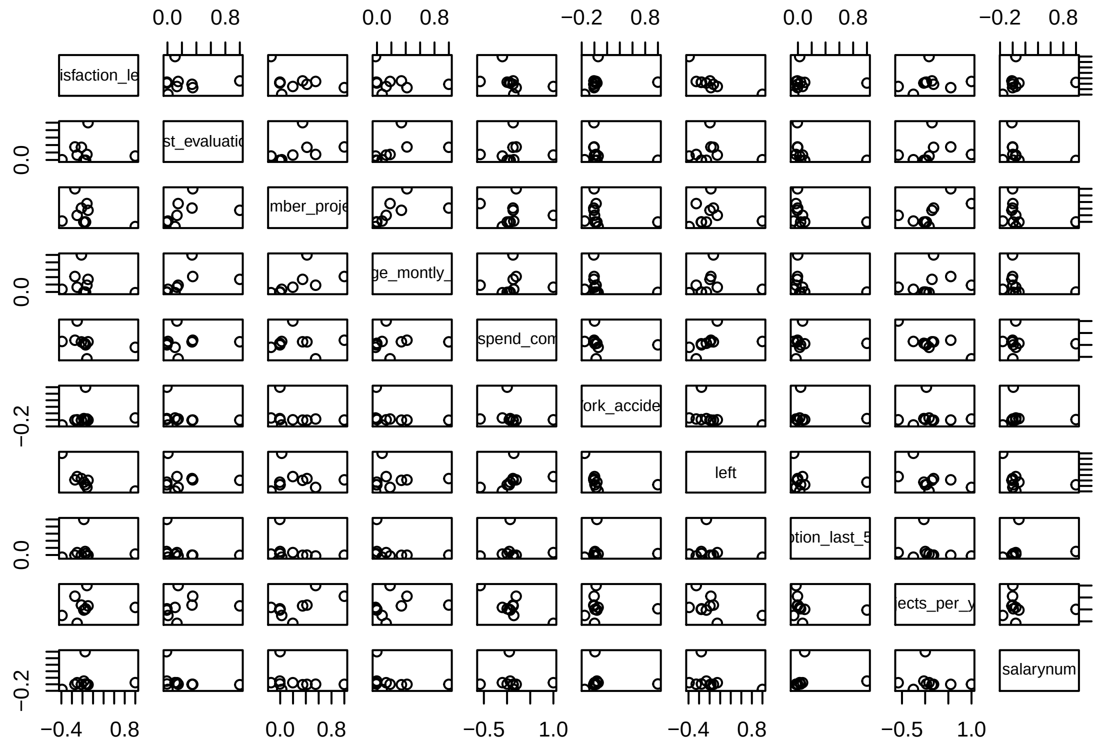



Moderately Positively Correlated Variables: - number_project vs last_evaluation: 0.35 - number_project vs average_monthly_hours: 0.42 - number_project vs projects_per_year: 0.55

Moderately Negatively Correlated Variables: - time_spend_company vs projects_per_year: -0.58 - left vs satisfaction_level: -0.39

We made a corrplot in order to see the relation between our variables. Now we will do a scatterplot too. To do this, we need change our categorical variables into numeric variables.

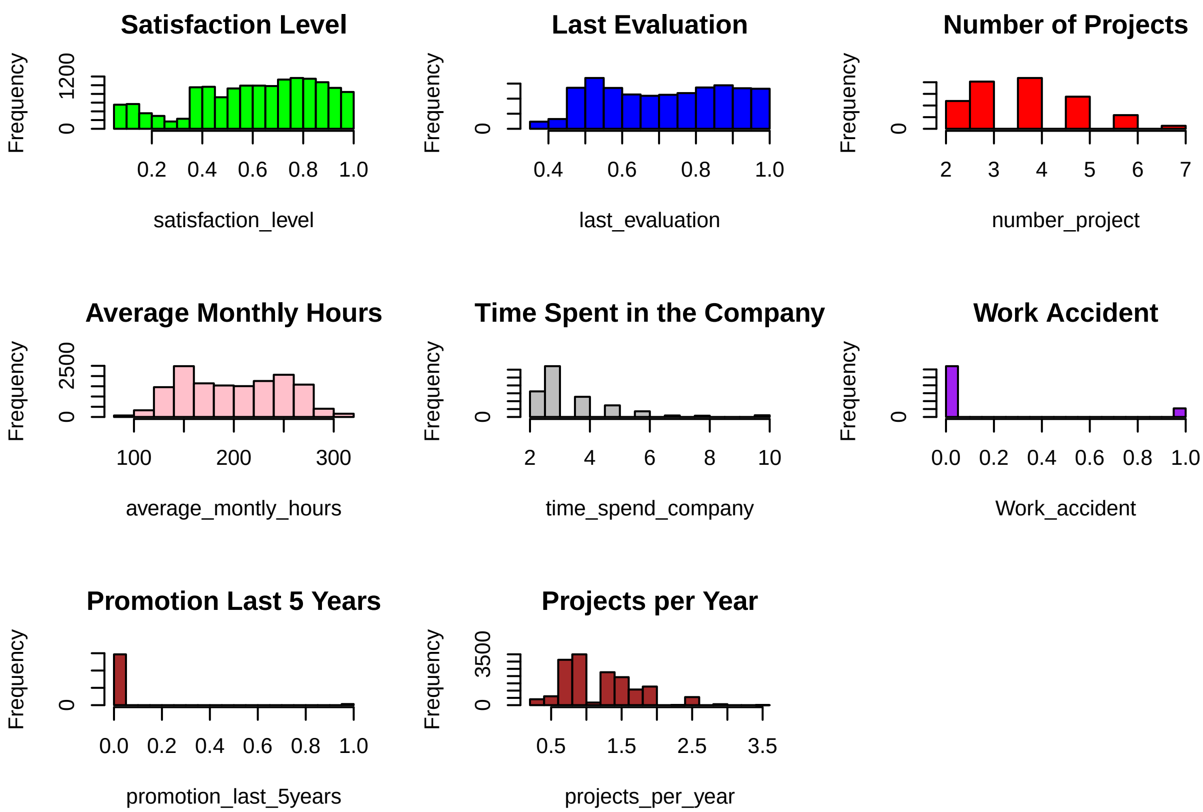
```
require(car)
db$salarynum<-recode(db$salary, "'low'=1")
db$salarynum<-recode(db$salarynum, "'medium'=2")
db$salarynum<-recode(db$salarynum, "'high'=3")
db$salarynum<-as.numeric(db$salarynum)

library(dplyr)
library(graphics)
library(corrplot)
HR_dataset_corrplot <- select(db, -department, -salary)
HR_correlation <- cor(HR_dataset_corrplot)
library(corrplot)

par(mfrow=c(1,1))
pairs(HR_correlation)
```



### 3.3 - Exploration through visualisation

**DISTRIBUTION PLOTS (HISTOGRAMS):**

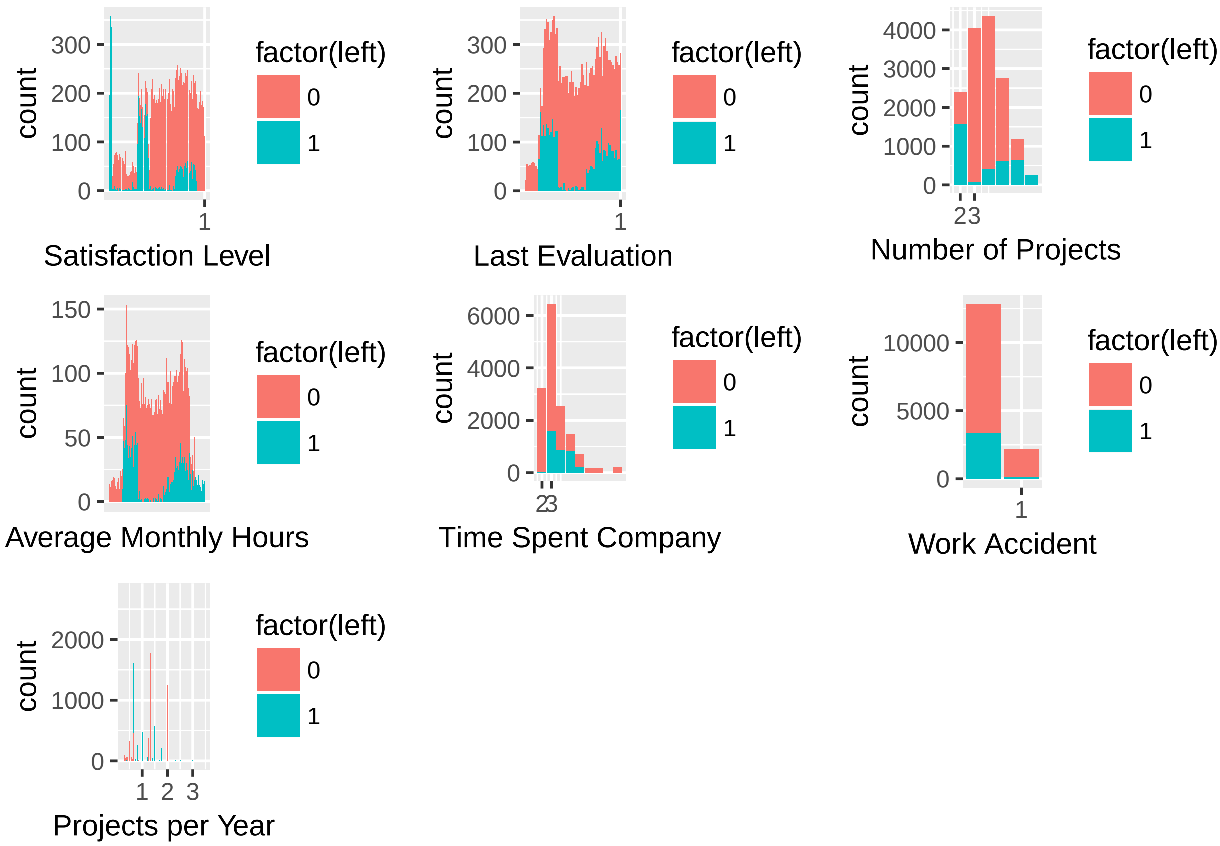
```
attach(db) # In this way we avoid including the database name in every line of code
par(mfrow=c(3,3))
hist(satisfaction_level,main="Satisfaction Level" ,col="green")
hist(last_evaluation,main= "Last Evaluation",col="blue")
hist(number_project,main="Number of Projects", col="red")
hist(average_montly_hours,main="Average Monthly Hours", col="pink")
hist(time_spend_company,main="Time Spent in the Company",col="grey")
hist(Work_accident,main="Work Accident",col="purple")
hist(promotion_last_5years,main="Promotion Last 5 Years",col="brown")
hist(projects_per_year,main="Projects per Year",col="brown")
```




**STACKED BAR PLOTS:**

```
# The below code will install the ggplot package if it doesn't exist, and then load it
if (!require(ggplot2)) install.packages("ggplot2")
library(ggplot2)

ggplot_satisfaction_level<-ggplot(db,aes(x=satisfaction_level,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
  labs(x="Satisfaction Level")

ggplot_last_evaluation<-ggplot(db,aes(x=last_evaluation,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
  labs(x="Last Evaluation")

ggplot_NumberProjects<-ggplot(db,aes(x=number_project,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
  labs(x="Number of Projects")

ggplot_avghours<-ggplot(db,aes(x=average_montly_hours,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
```

```
  labs(x="Average Monthly Hours")

ggplot_time_company<-ggplot(db,aes(x=time_spend_company,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
  labs(x="Time Spent Company")

ggplot_work_accident<-ggplot(db,aes(x=Work_accident,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
  labs(x="Work Accident")

ggplot_promotion5years<-ggplot(db,aes(x=promotion_last_5years,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
  labs(x="Promotion Last 5 Years")

ggplot_projectsYear<-ggplot(db,aes(x=projects_per_year,fill=factor(left)))+
  geom_bar(stat='count',position='stack')+
  scale_x_continuous(breaks=c(1:3))+
  labs(x="Projects per Year")

# The below code will install the gridExtra package if it doesn't exist, and then load it
if (!require(gridExtra)) install.packages("gridExtra")
library(gridExtra)


grid.arrange(ggplot_satisfaction_level, ggplot_last_evaluation,ggplot_NumberProjects,
             ggplot_avghours,ggplot_time_company,ggplot_work_accident,
             ggplot_projectsYear, ncol=3,nrow=3)
```




Subset the data for the density plots:

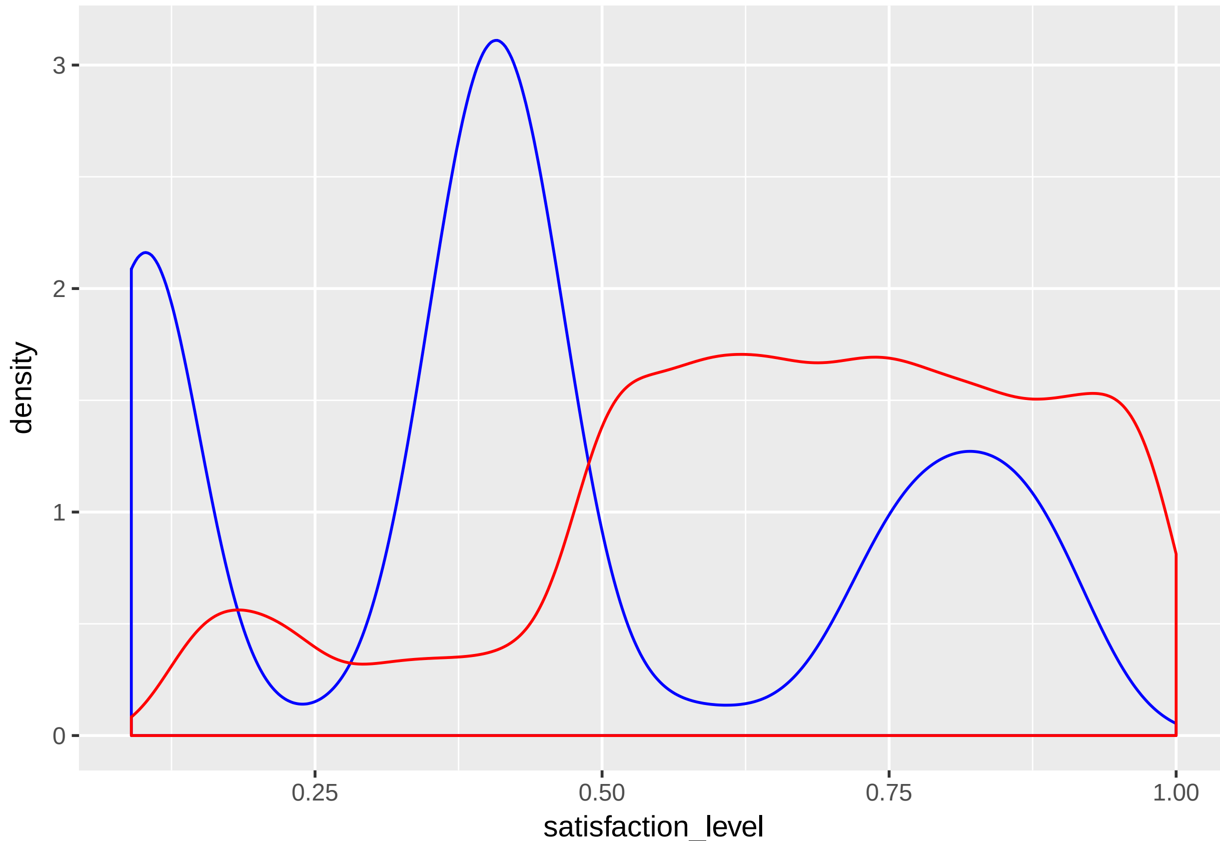
```
left_data<-subset(db,left==1)
stay_data<-subset(db,left==0)
```

KEY HIGHLIGHTS:

1 - Satisfacion Level: most leavers have either a very low satisfaction level or medium satisfaction. Although there in important chunk of leavers with high turnover as well.

This can be more clearly seen in the following density plot:

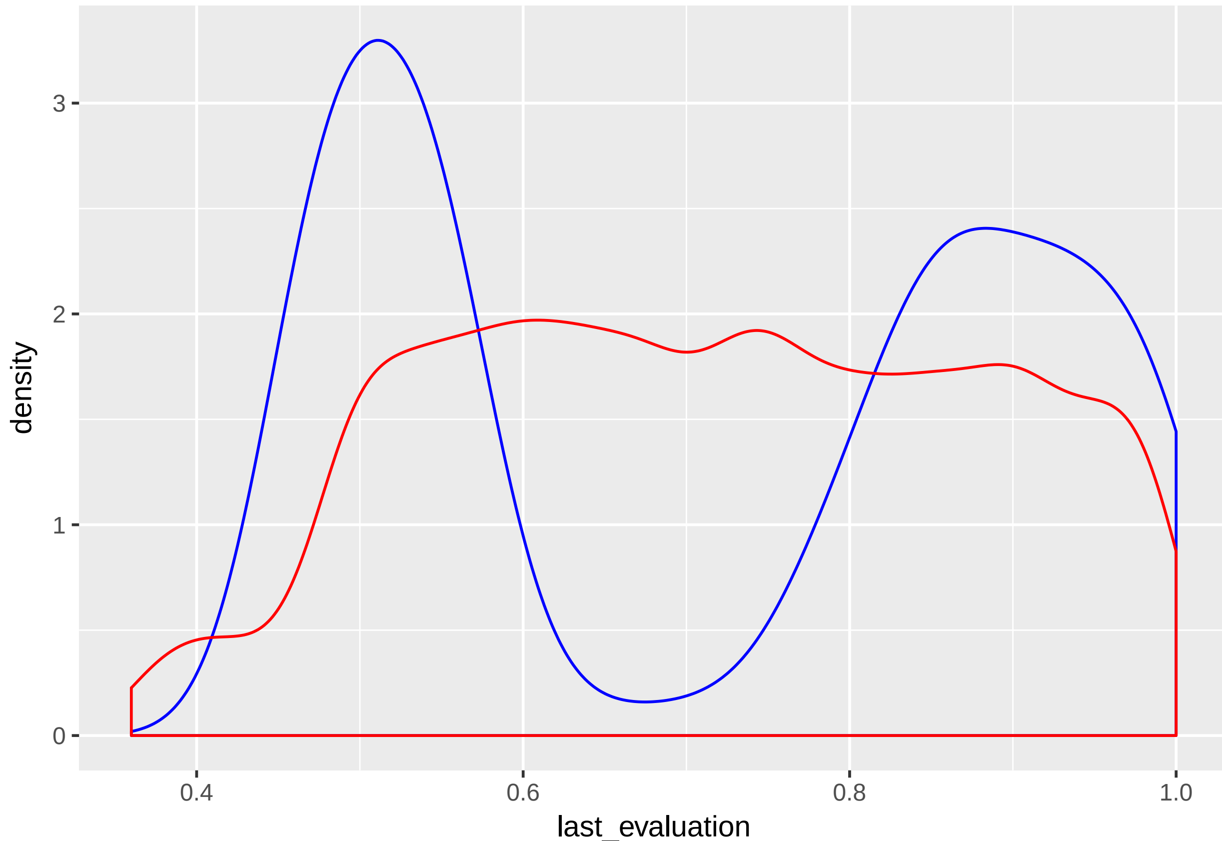
```
ggplot() + geom_density(aes(x=satisfaction_level),colour="blue",data=left_data) +
  geom_density(aes(x=satisfaction_level), colour="red",data=stay_data)
```




2 - Last Evaluation: most leavers have either a low or high evaluation score.

This can be more clearly seen in the following density plot:

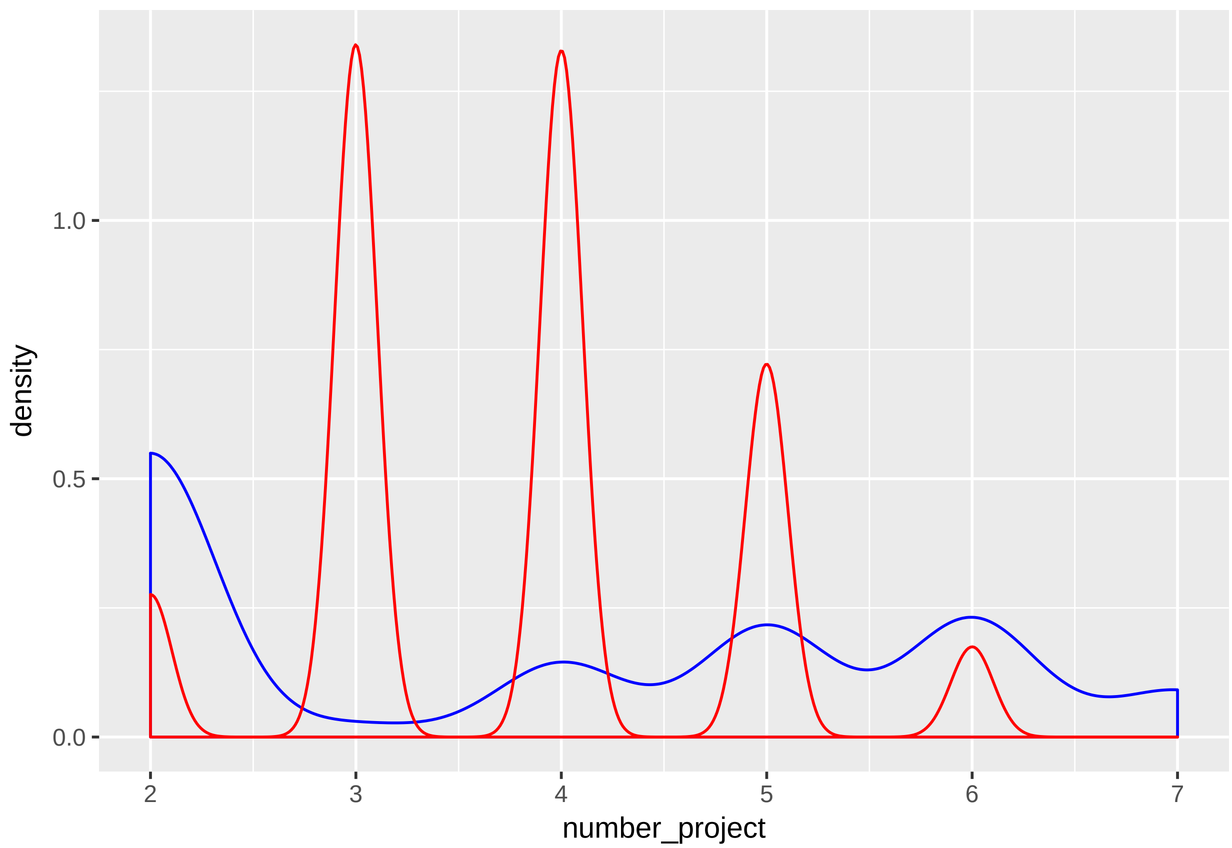
```
ggplot() + geom_density(aes(x=last_evaluation),colour="blue",data=left_data) +
  geom_density(aes(x=last_evaluation), colour="red",data=stay_data)
```




3 - Number Projects: employees with too few (<2) or too many (>4) projects leave more.

This can be more clearly seen in the following density plot:

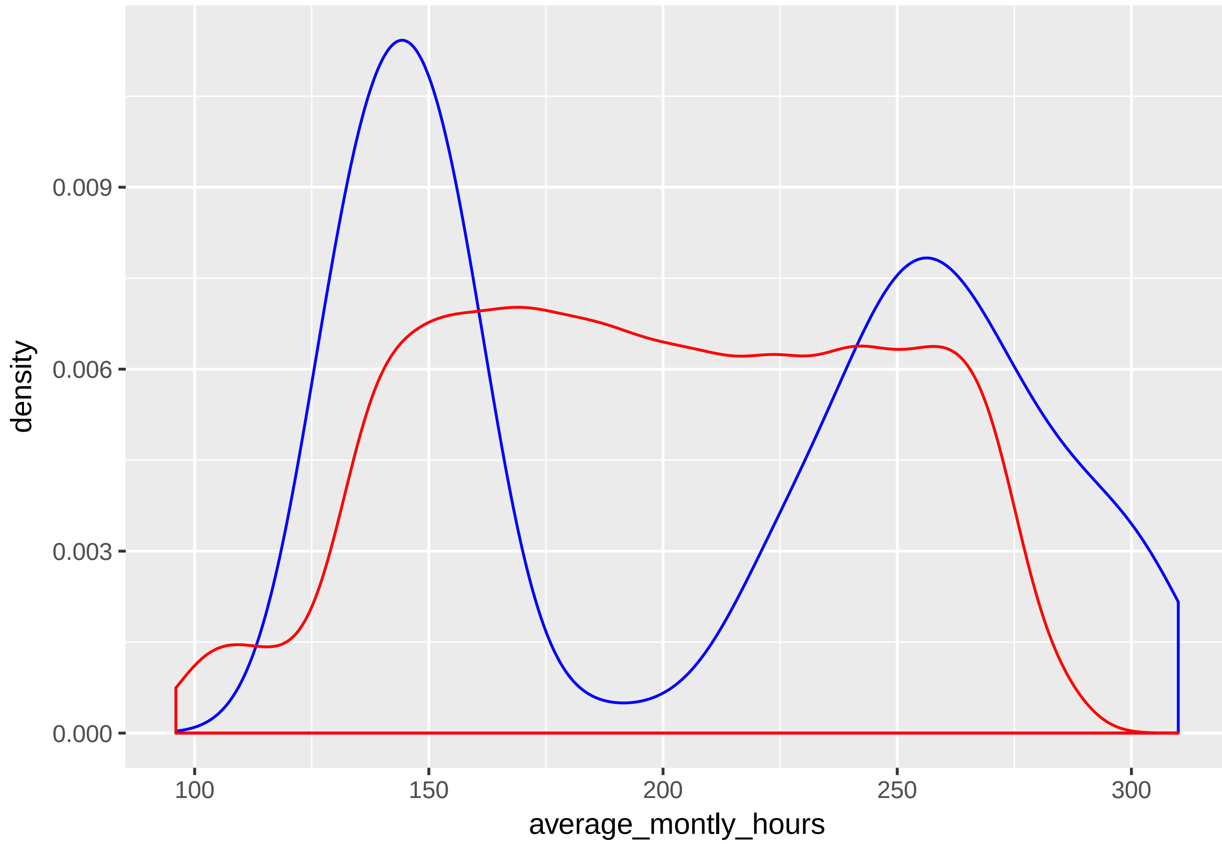
```
ggplot() + geom_density(aes(x=number_project),colour="blue",data=left_data) +
  geom_density(aes(x=number_project), colour="red",data=stay_data)
```




4 - Average Monthly Hours: employees with low (<150) or high (>250) numbers of hours leave the organisation.

This can be more clearly seen in the following density plot:

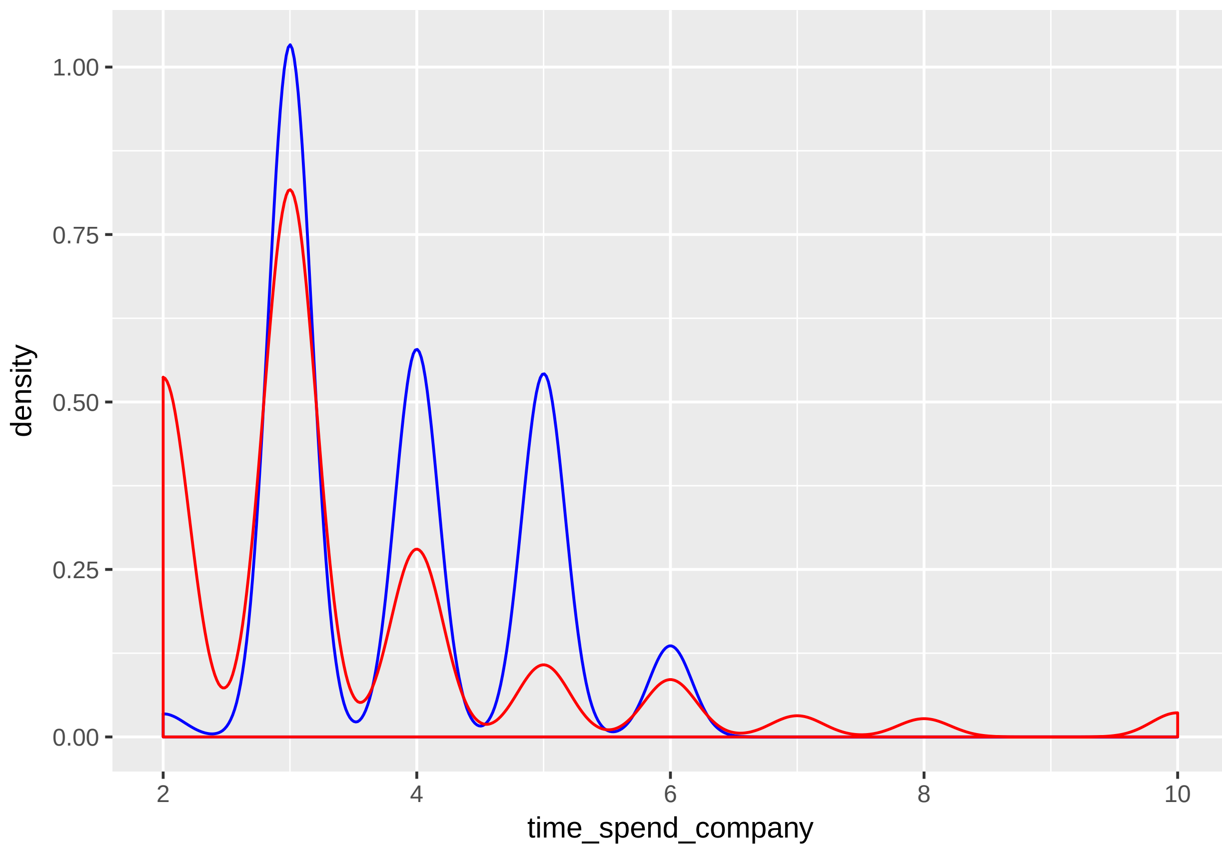
```
ggplot() + geom_density(aes(x=average_montly_hours),colour="blue",data=left_data) +
  geom_density(aes(x=average_montly_hours), colour="red",data=stay_data)
```




5 - Time Spent Company: employees with a tenure of 3-6 years leave more the organisation.

This can be more clearly seen in the following density plot:

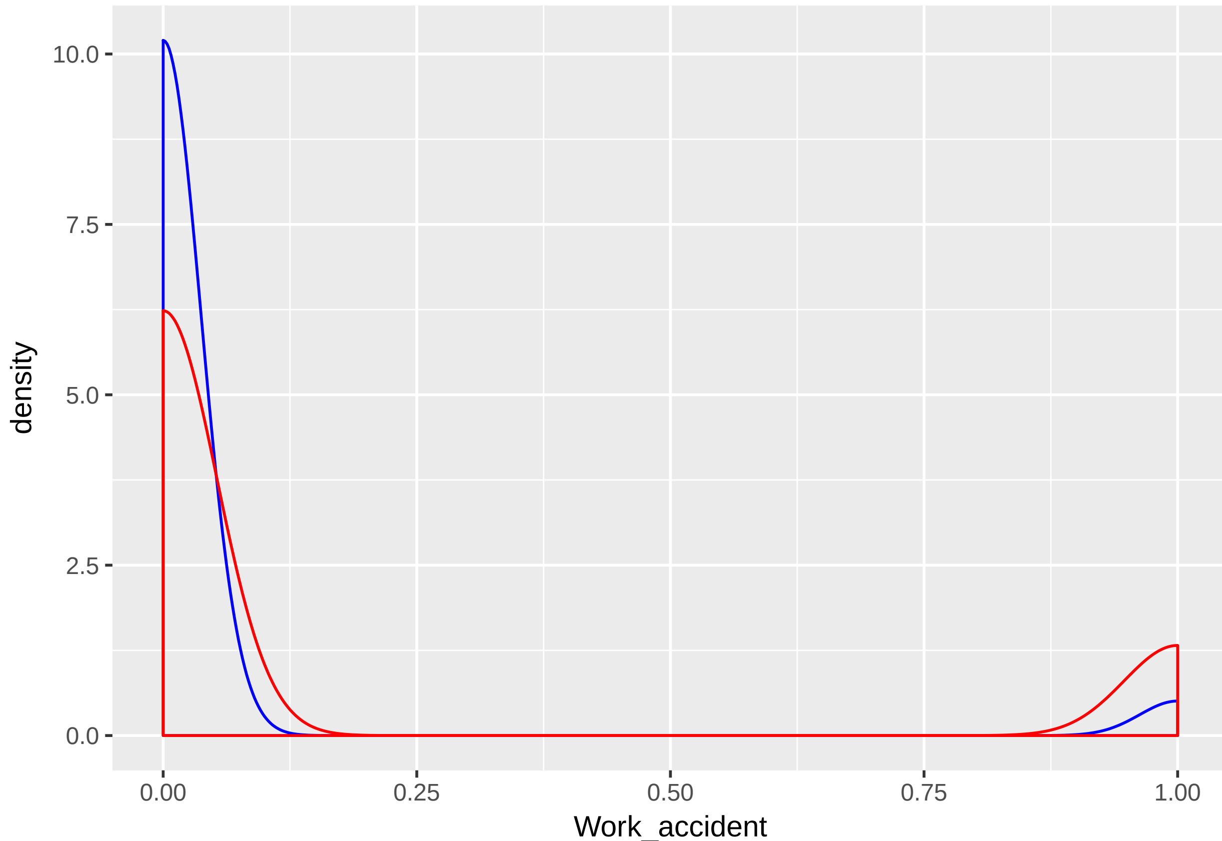
```
ggplot() + geom_density(aes(x=time_spend_company),colour="blue",data=left_data) +
  geom_density(aes(x=time_spend_company), colour="red",data=stay_data)
```




6 - Work Accident: leavers are more likely to not to have had a work accident.

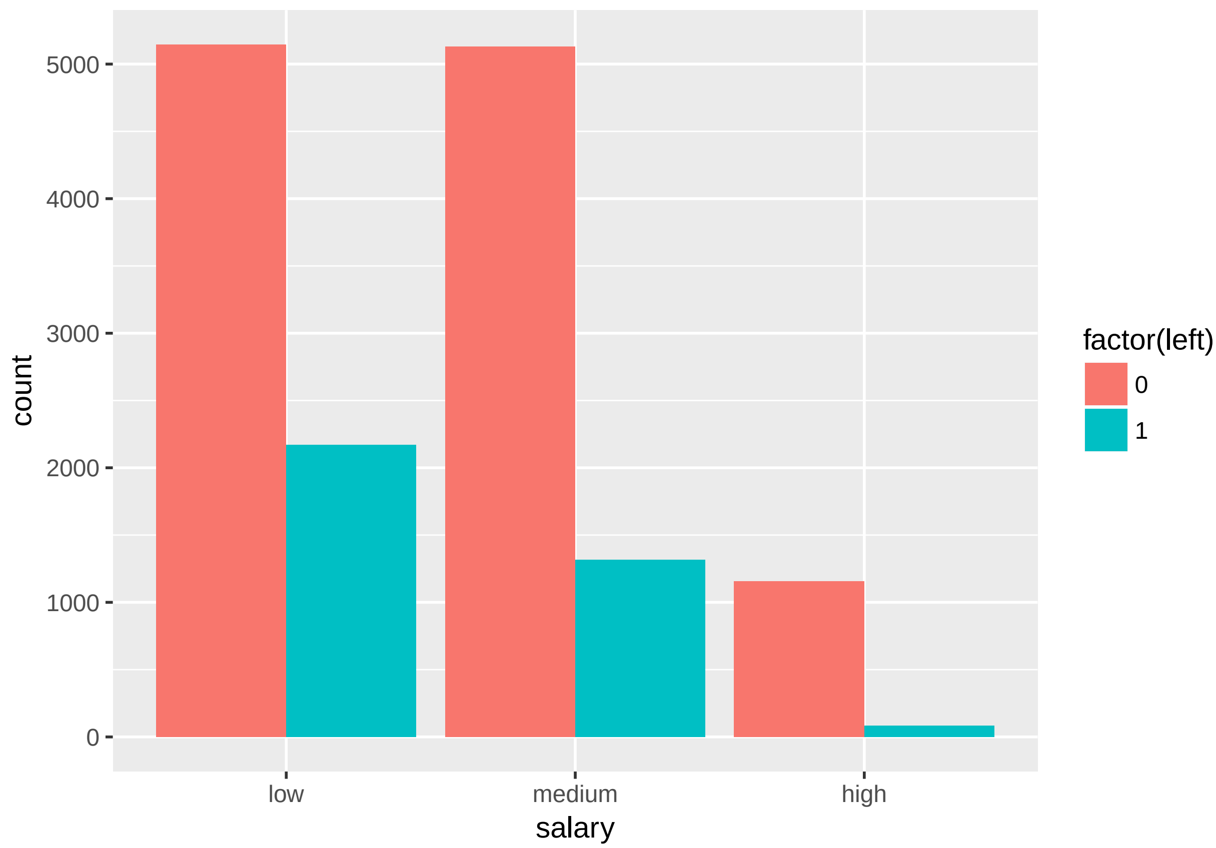
This can be more clearly seen in the following density plot:

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




Turnover VS Salary Level:

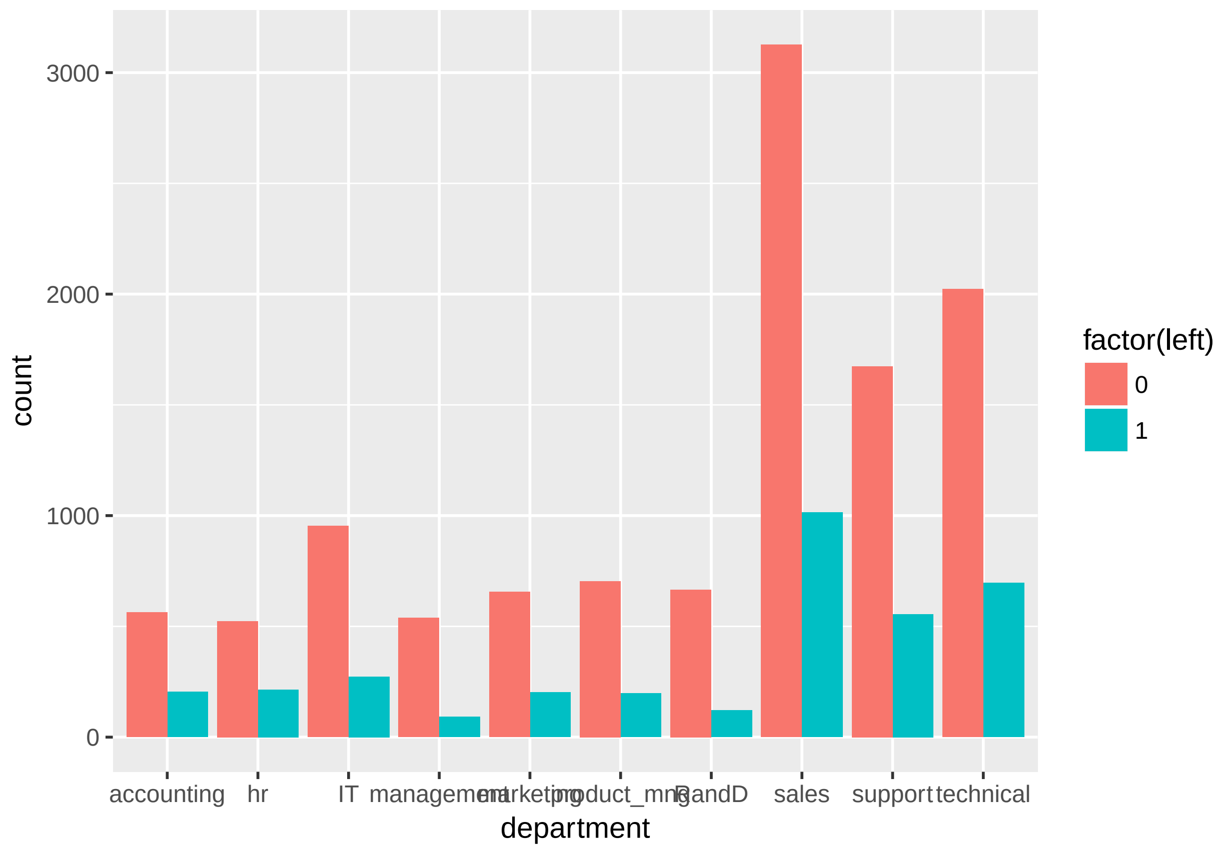
```
db$salary<-factor(db$salary,levels=c("low","medium","high")) # Manual ordering of the levels of the vari

ggplot(db,aes(x=salary,fill=factor(left)))+
  geom_bar(stat='count',position='dodge')
```




We see how most leavers have a low or medium salary

Turnover VS Department:

```
ggplot(db,aes(x=department,fill=factor(left)))+
  geom_bar(stat='count',position='dodge')
```




Most leavers come from the Sales, Support and Technical departments.

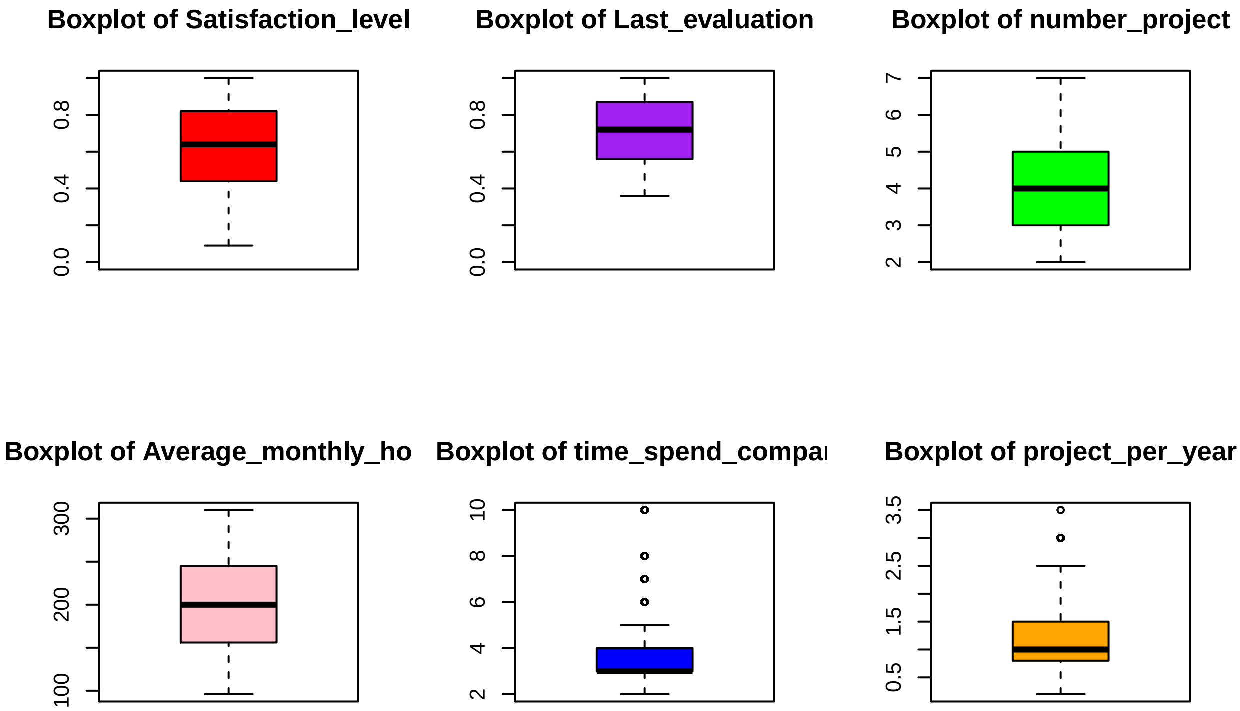
### BOXPLOTS:

Now we are going to do some boxplots of the cuantitative variables, to have a "more accurate" plot with more information like the median or the first and third quartile. We can see too the normality of variables.

#### Single Boxplots

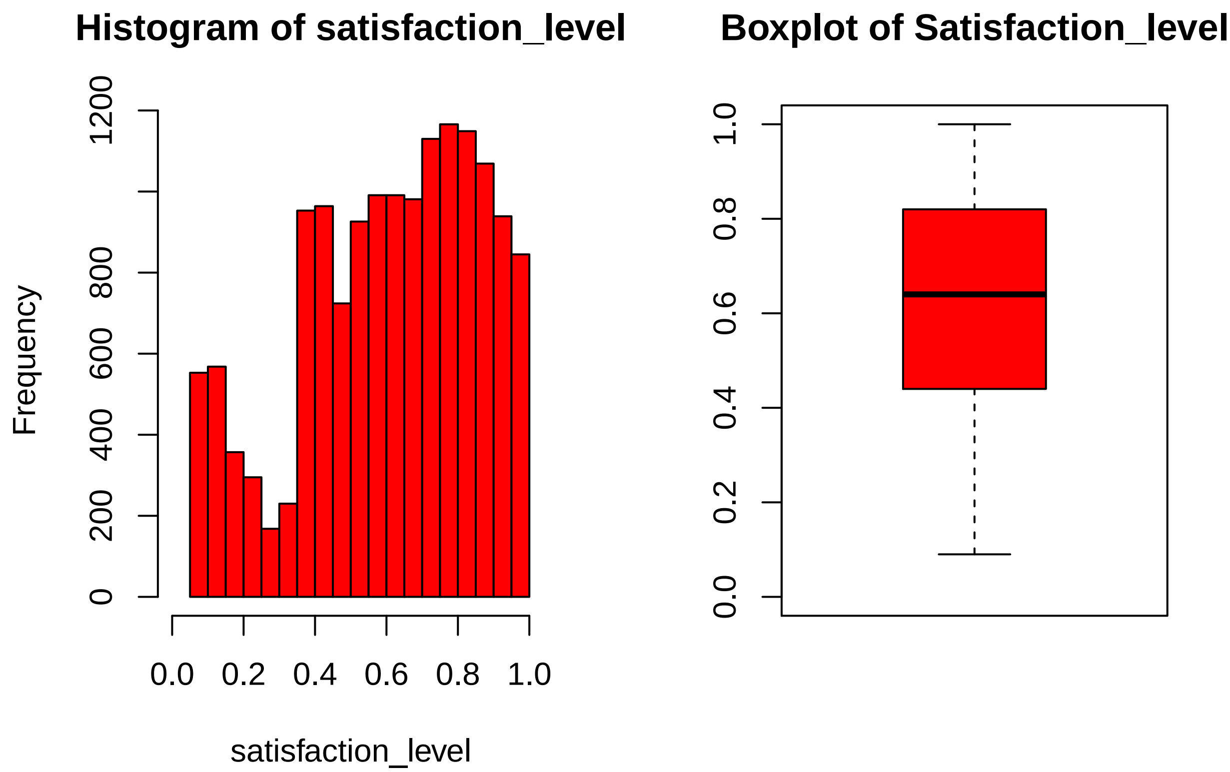
With the single boxplots we can see general visual representation of each variable.

```
attach(db)
par(mfrow=c(2,3))
satisbox<-boxplot(satisfaction_level, col = "red", ylim= c(0,1), main = "Boxplot of Satisfaction_level"
evalbox<-boxplot(last_evaluation, col = "purple", ylim= c(0,1), main = "Boxplot of Last_evaluation")
numbox<-boxplot(number_project, col="green", main = "Boxplot of number_project")
averbox<-boxplot(average_montly_hours, col = "pink", main = "Boxplot of Average_monthly_hours")
timesbox<-boxplot(time_spend_company, col = "blue", main = "Boxplot of time_spend_company")
proyebox<-boxplot(projects_per_year, col = "orange", main = "Boxplot of project_per_year")
```

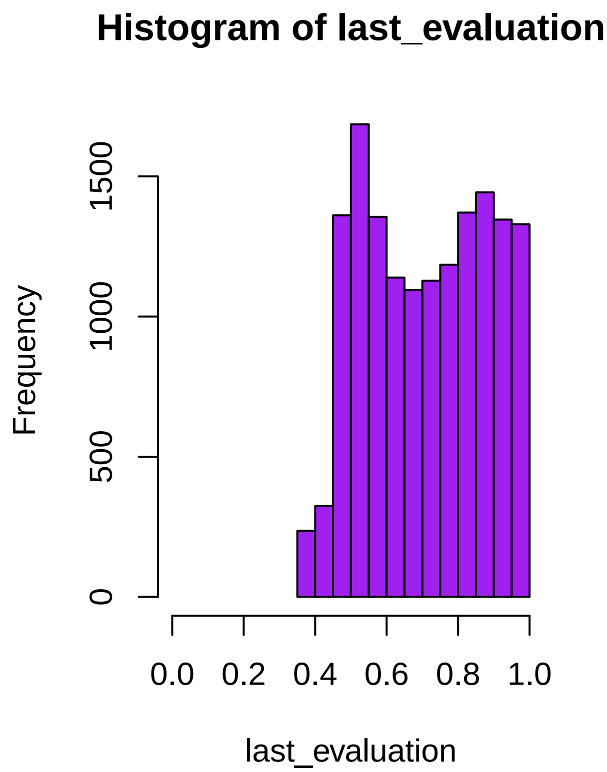


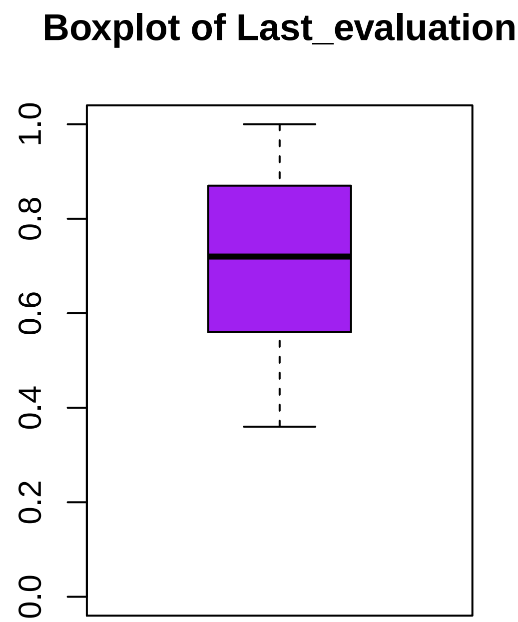

We can compare this boxplots with the histograms.

```
par(mfrow=c(1,2))
hist(satisfaction_level, col = "red", xlim = c(0,1))
boxplot(satisfaction_level, col = "red", ylim= c(0,1), main = "Boxplot of Satisfaction_level")
```

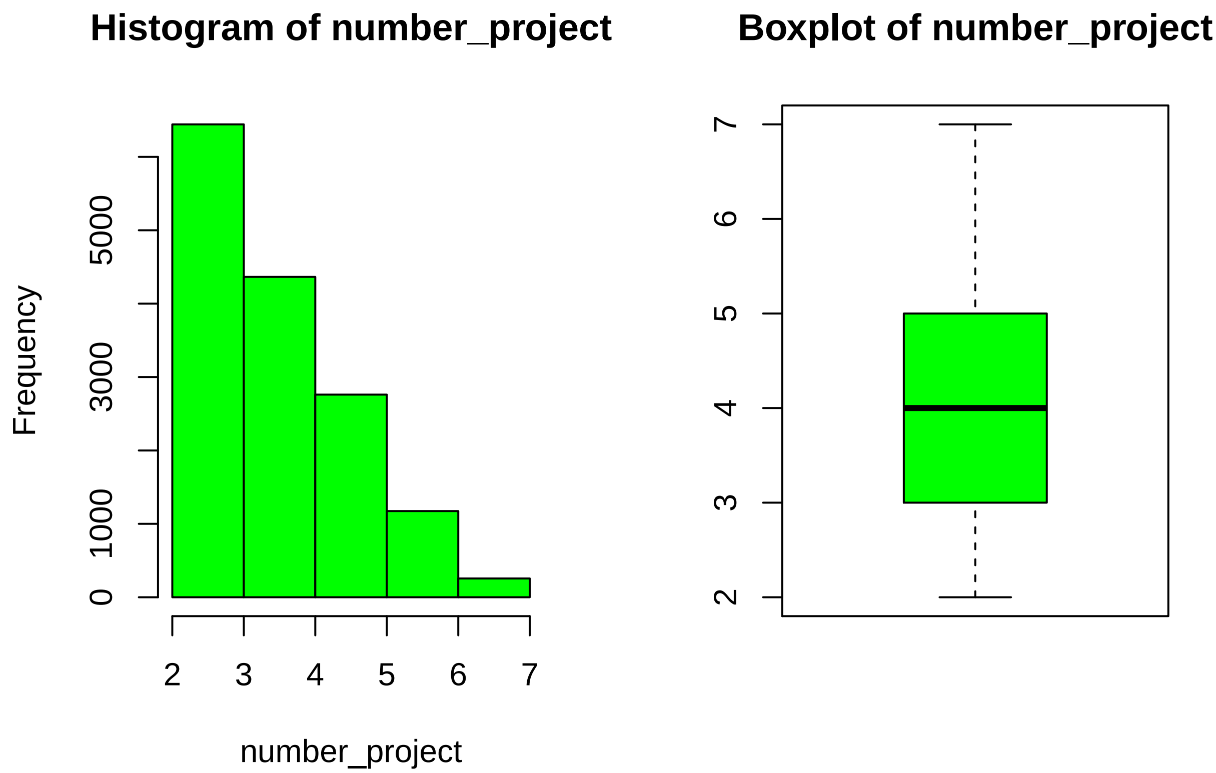



```r
hist(last_evaluation, col = "purple", xlim = c(0,1))
boxplot(last_evaluation, col = "purple", ylim= c(0,1), main = "Boxplot of Last_evaluation")
```

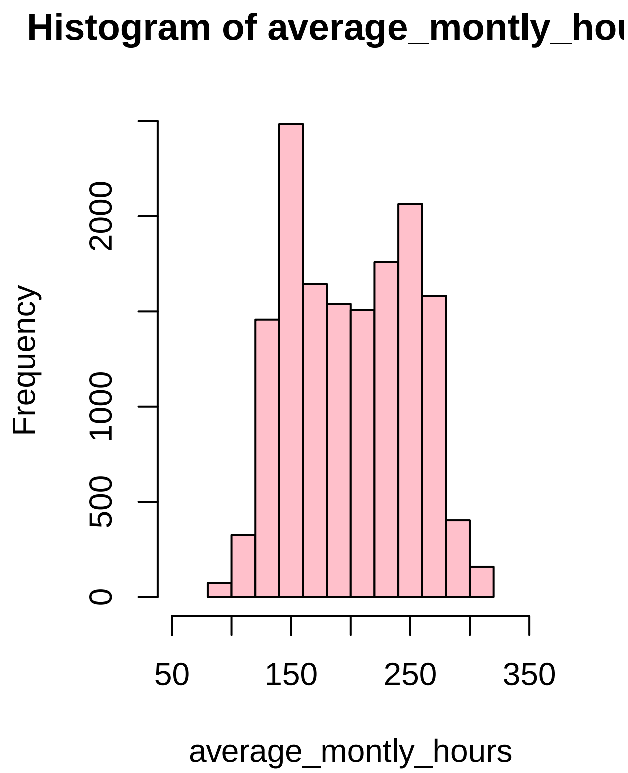


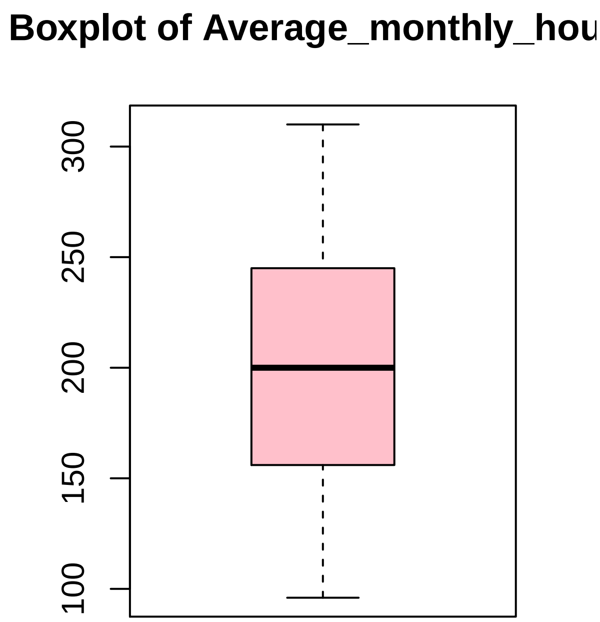




```
hist(number_project, col="green", breaks = 7)
boxplot(number_project, col="green", main = "Boxplot of number_project")
```

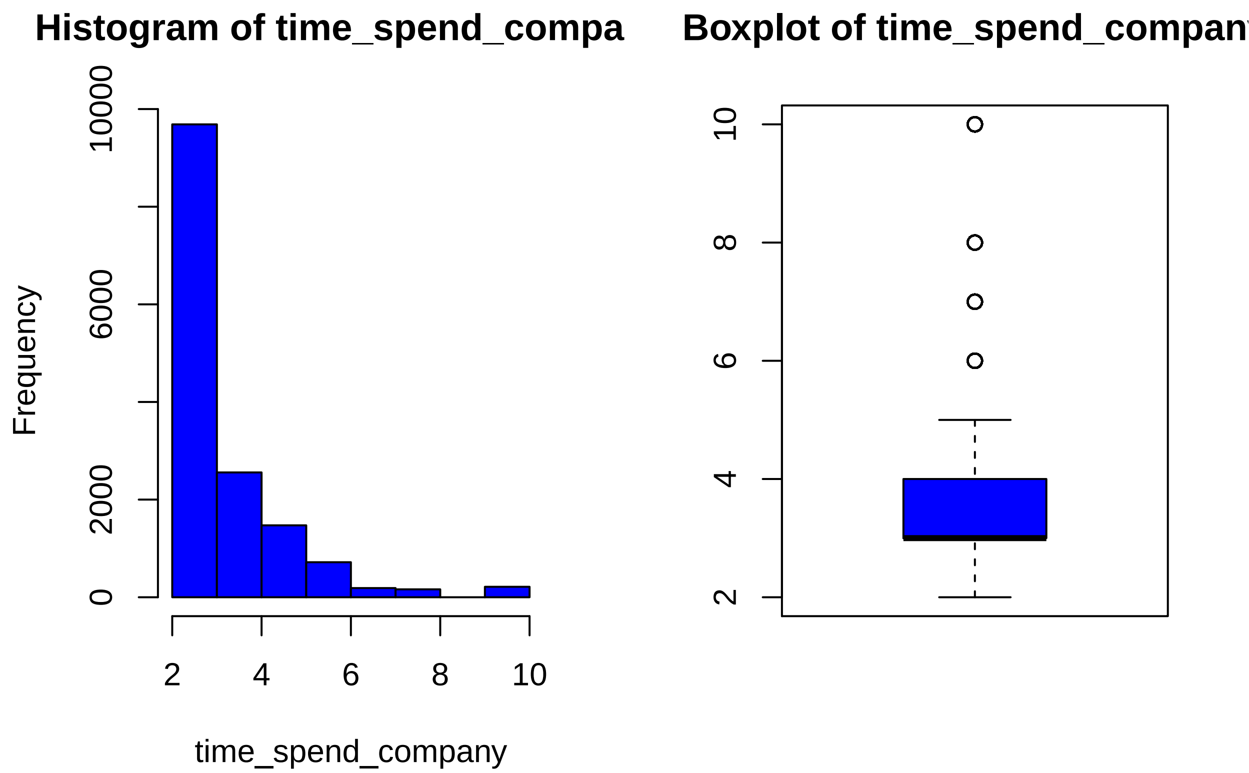



```r
hist(average_montly_hours, col = "pink", xlim= c(50, 350))
boxplot(average_montly_hours, col = "pink", main = "Boxplot of Average_monthly_hours")
```

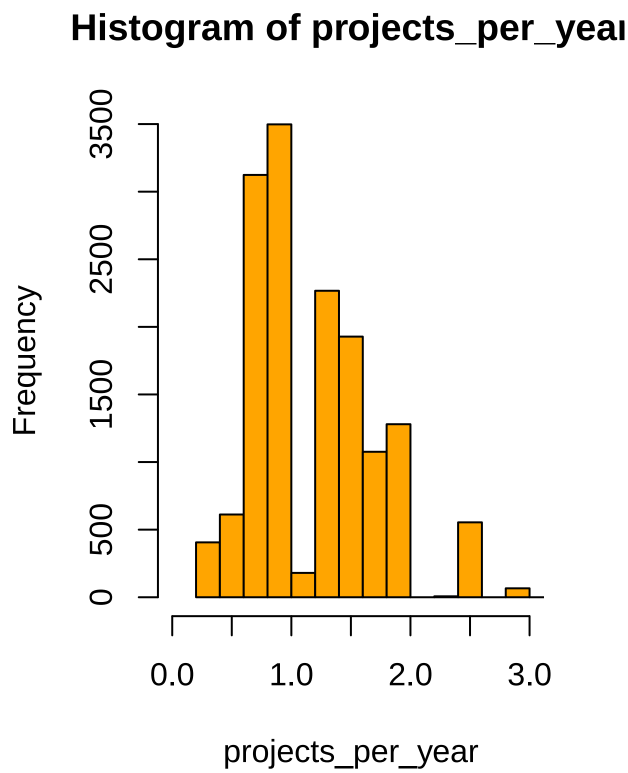


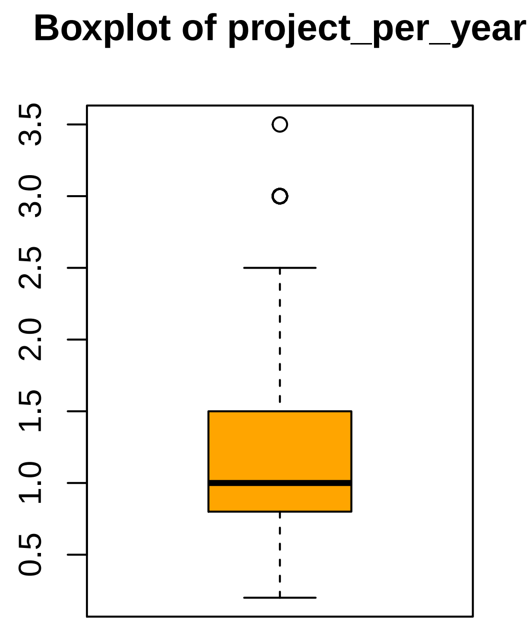




```
hist(time_spend_company, col = "blue", breaks = 8)
boxplot(time_spend_company, col = "blue", main = "Boxplot of time_spend_company")
```




```r
hist(projects_per_year, col = "orange", xlim= c(0,3))
boxplot(projects_per_year, col = "orange", main = "Boxplot of project_per_year")
```



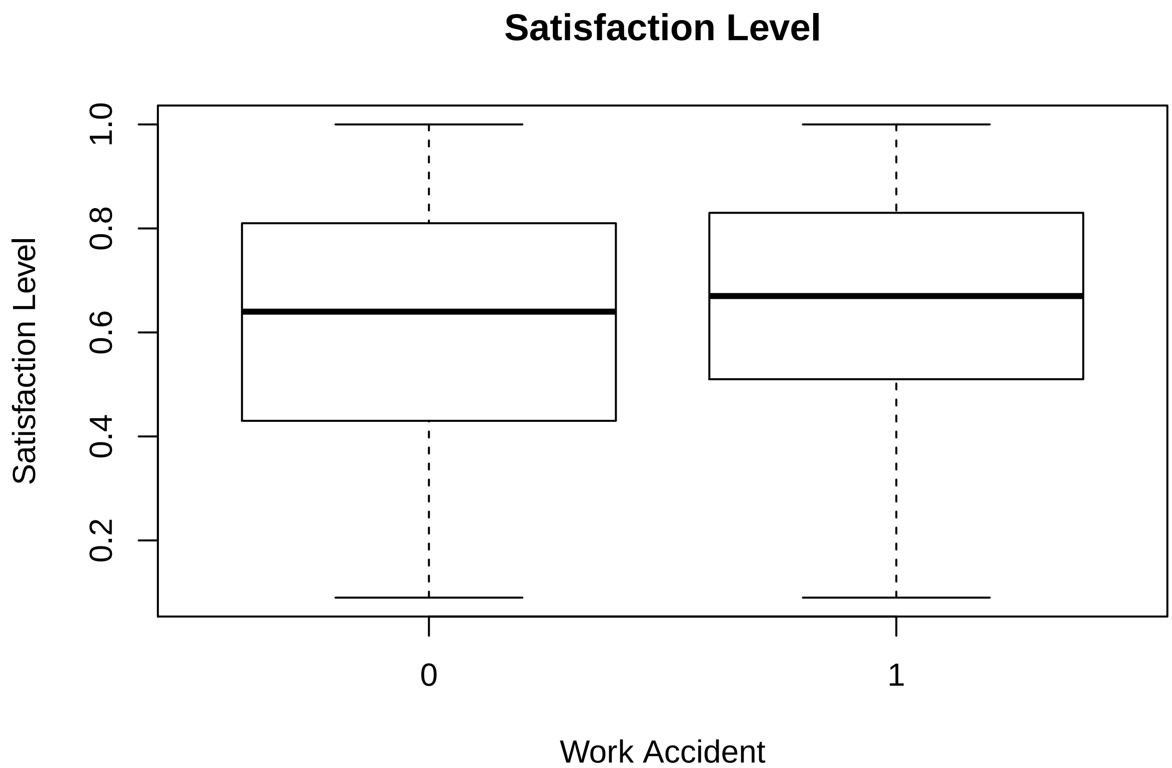

As we can see, there are only two variables that aren't normal: time_spend_company and project_per_year.

#### Relation boxplots

Satisfaction level

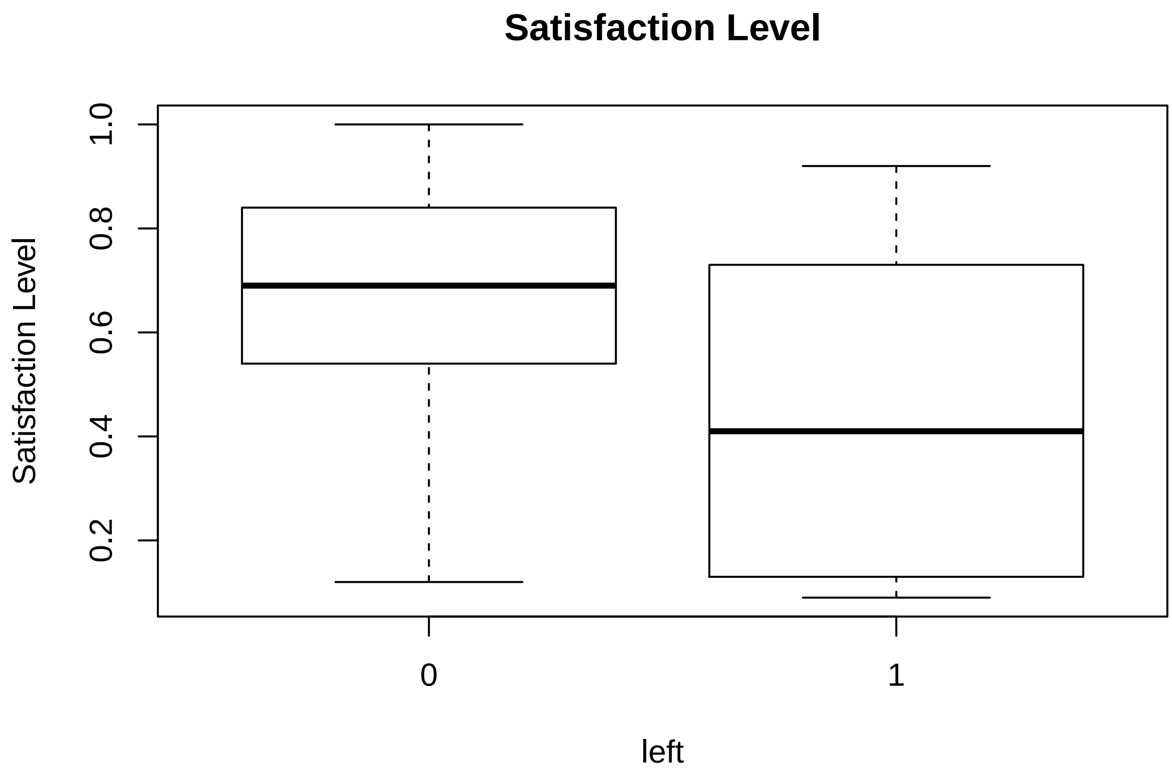
7 - The people that had an accident, has a better satisfaction level, and as said before, leave less than the people that hadnÂ´t. Maybe this is because they are less burned.

```
boxplot(satisfaction_level~Work_accident,data=db, main="Satisfaction Level",
        xlab="Work Accident", ylab="Satisfaction Level")
```



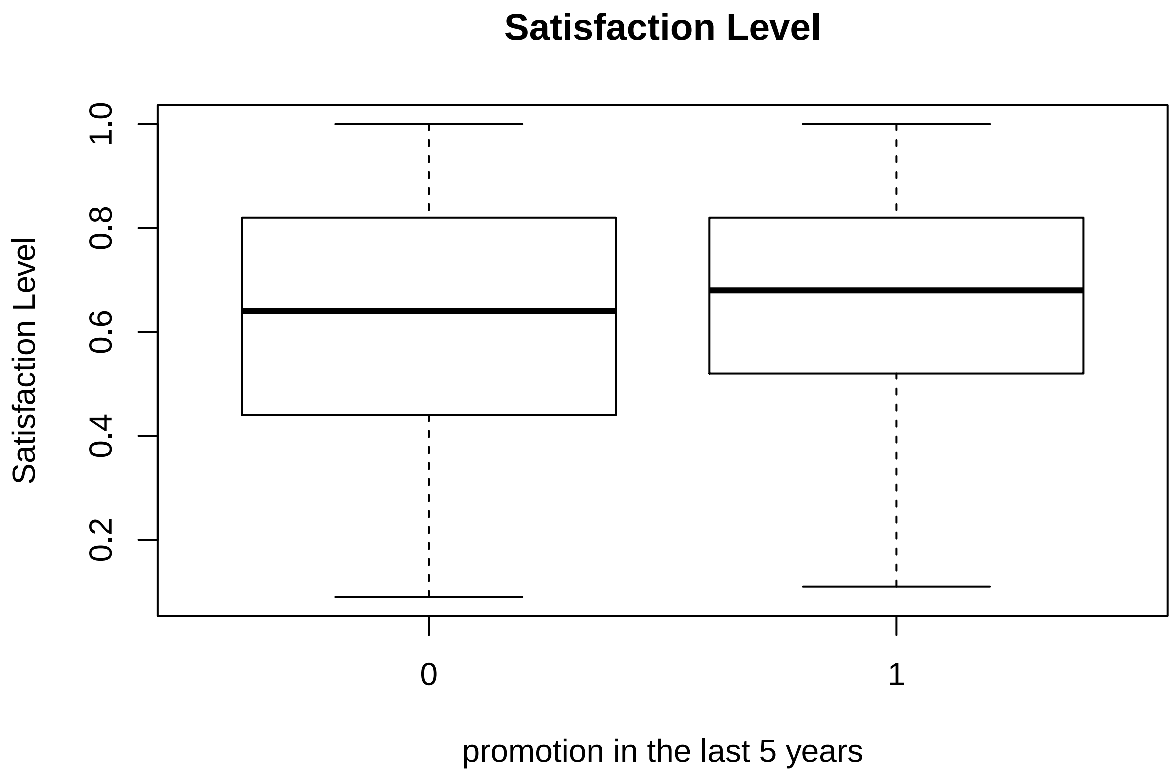

8 - As said before, leavers have a low or medium satisfaction level. We can see here that the median of the satisfaction level of the leavers is lower than non leavers.

```
boxplot(satisfaction_level~left,data=db, main="Satisfaction Level",
        xlab="left", ylab="Satisfaction Level")
```




9 - The people that were recently promoted is slightly happier.

```
boxplot(satisfaction_level~promotion_last_5years,data=db, main="Satisfaction Level",
        xlab="promotion in the last 5 years", ylab="Satisfaction Level")
```



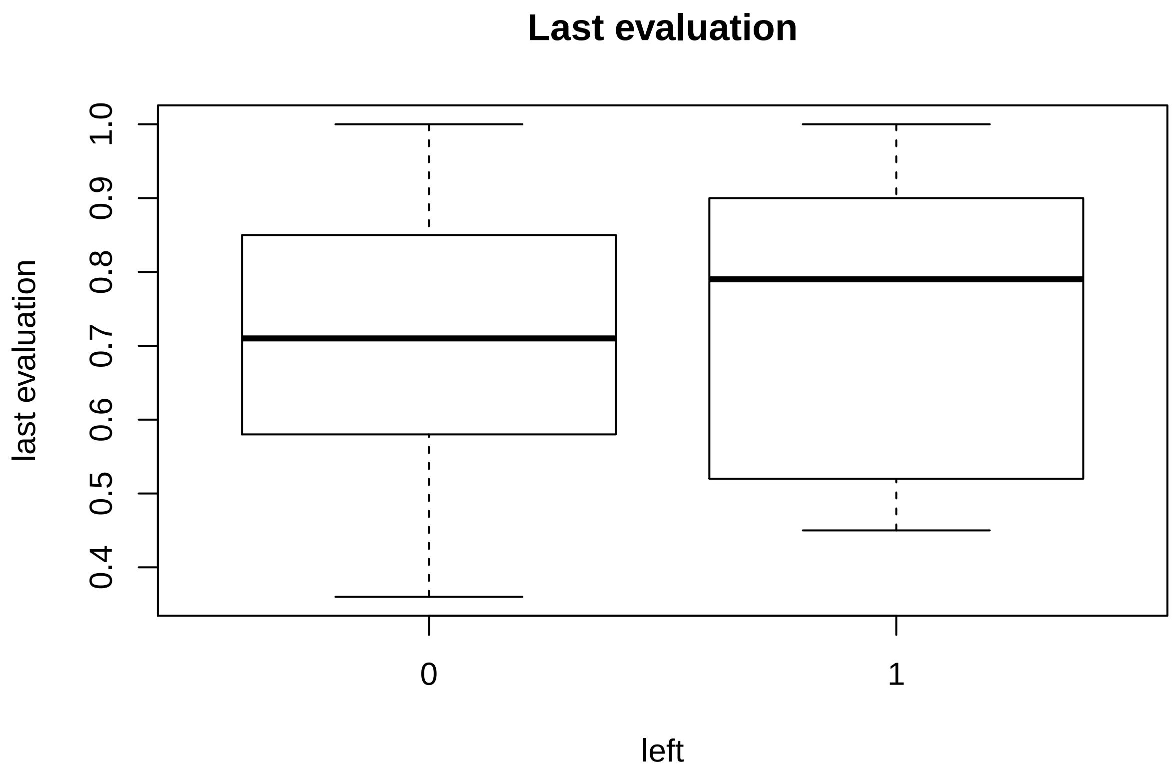

10 - Besides of leaving the company, this plot shows that the people with less salary is less happy in the work.

```
boxplot(satisfaction_level~salary,data=db, main="Satisfaction Level",
        xlab="Salary", ylab="Satisfaction Level")
```


Last evaluation

10 - This plot isn´t very reliable because the leavers population is very asymetric. Besides that, the plot shows us that the people with the best performance is the people that it's going from the company. Before it was seen that the people with low evaluation leaves too.

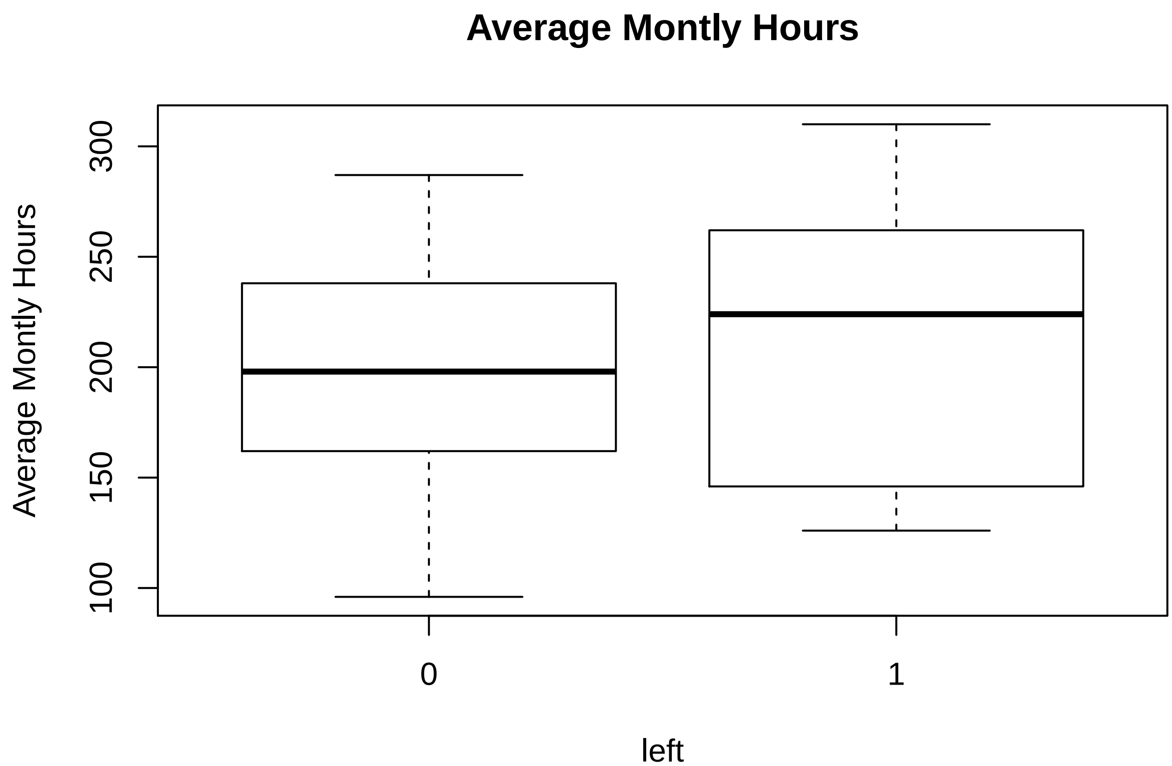
```
boxplot(last_evaluation~left,data=db, main="Last evaluation",
        xlab="left", ylab="last evaluation")
```




Monthly hours

11 - The same reliabilility with this plot. The people with more montly hours, is probably burned, and for that leaves the company. As said before, the people with little hours leaves too.

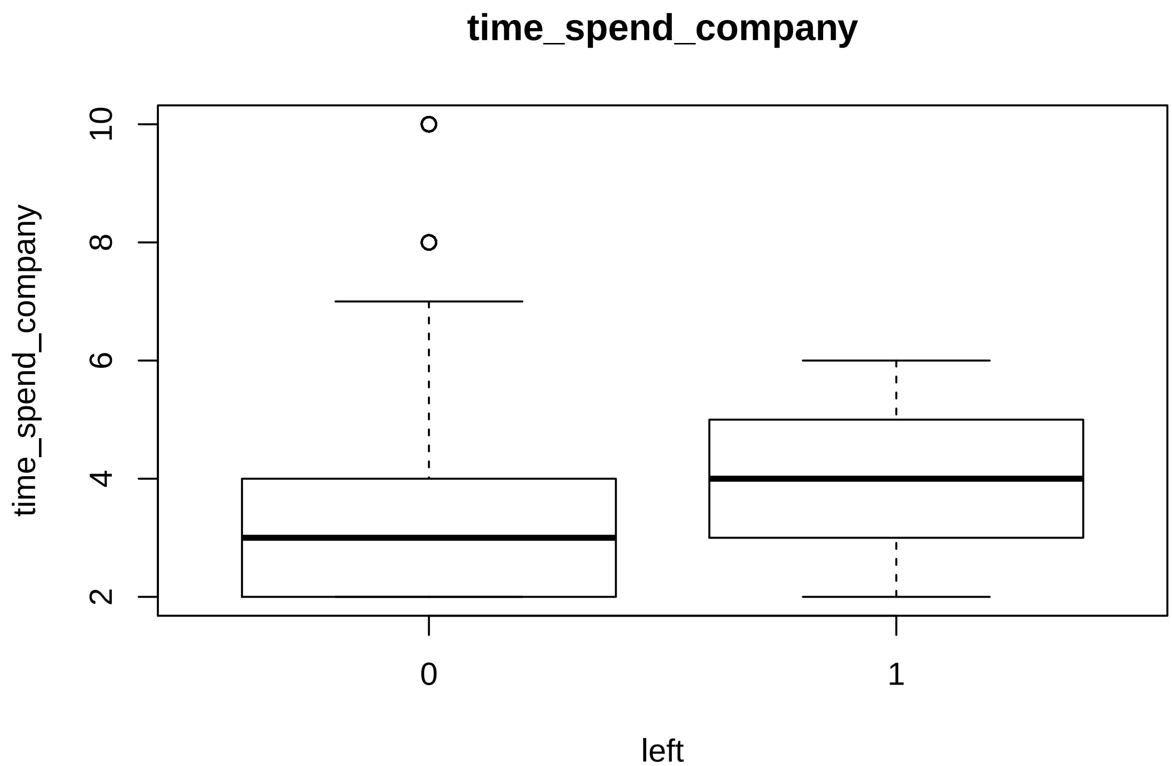
```
boxplot(average_montly_hours~left,data=db, main="Average Montly Hours",
        xlab="left", ylab="Average Montly Hours")
```




Time in the company

12 - The people who were more time in the company were probable more burned and left. There are outliers though. The people that has been more time in the company doesnÂ´t leave it.

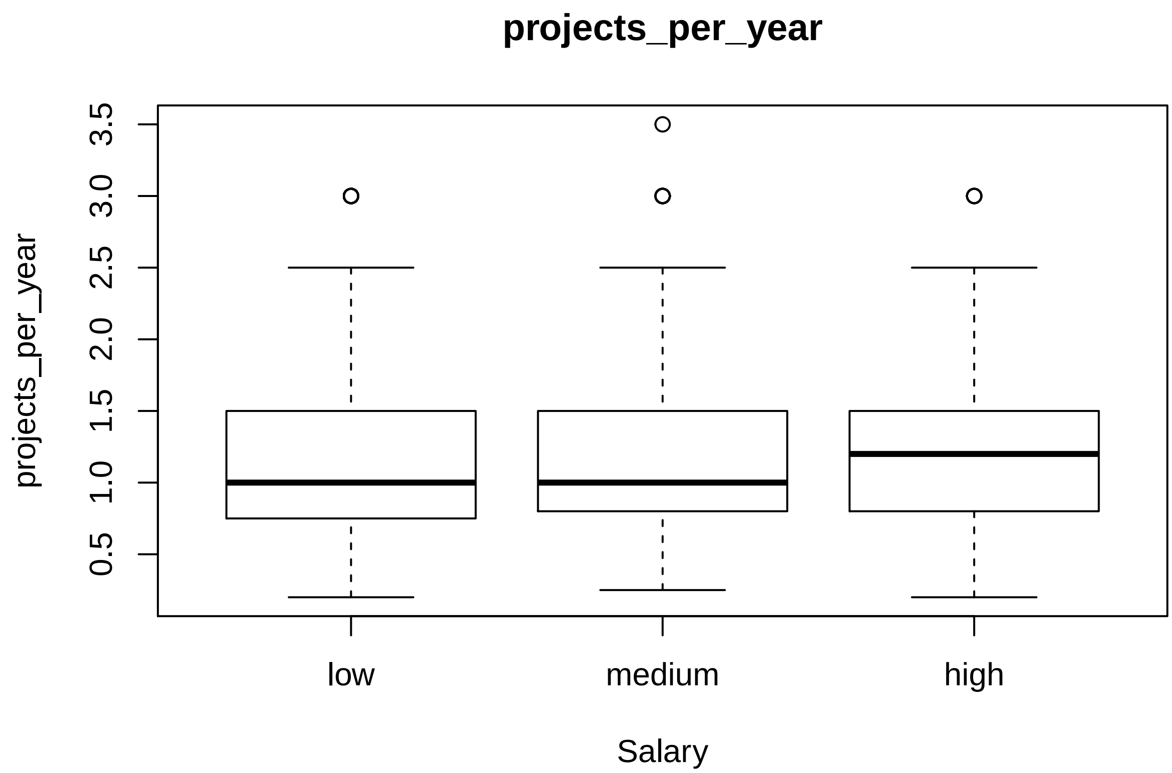
```
boxplot(time_spend_company~left,data=db, main="time_spend_company",
        xlab="left", ylab="time_spend_company")
```




Projects per year

13 - In general, they give more salary if you have mor projects per year.

```
boxplot(projects_per_year~salary,data=db, main="projects_per_year",
        xlab="Salary", ylab="projects_per_year")
```




## FREQUENCY TABLES

Frequency tables are used to compare qualitative variables. In this case, because of the different number of people in the left and not-left are so different, the chi-square method is inefficient. For that, we are going to do proportion tables.

Left

14 - Before it was said the department with more leavers. We will see now the proportion of them. In the management and RandD department there are less proportion of leavers. The department with more proportion of them, is the hr department.

```
attach(db)
tabla<-xtabs(~left+department)
ptabla<-prop.table(tabla, 2)
ptabla
```

```
##     department
## left accounting        hr        IT management marketing product_mng
##    0  0.7340287 0.7090663 0.7775061  0.8555556 0.7634033   0.7804878
##    1  0.2659713 0.2909337 0.2224939  0.1444444 0.2365967   0.2195122
##     department
## left     RandD     sales   support technical
##    0 0.8462516 0.7550725 0.7510094 0.7437500
##    1 0.1537484 0.2449275 0.2489906 0.2562500
```

```r
summary(ptabla)
```

```
## Call: xtabs(formula = ~left + department)
## Number of cases in table: 10
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 0.11121, df = 9, p-value = 1
##  Chi-squared approximation may be incorrect
```

15 - Here we can see, as before, that when the people has more salary, there are less probability to left the company.

```r
tabla<-xtabs(~left+salary)
ptabla<-prop.table(tabla, 2)
ptabla
```

```
##     salary
## left        low     medium       high
##    0 0.70311646 0.79568725 0.93371059
##    1 0.29688354 0.20431275 0.06628941
```

```r
summary(ptabla)
```

```
## Call: xtabs(formula = ~left + salary)
## Number of cases in table: 3
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 0.17558, df = 2, p-value = 0.916
##  Chi-squared approximation may be incorrect
```

16 - Unless the people with 2 projects, when the people has more projects the probability of leaving increases, confirming what was said before.

```r
tabla<-xtabs(~left+number_project)
ptabla<-prop.table(tabla, 2)
ptabla
```

```
##     number_project
## left          2          3          4          5          6          7
##    0 0.34380235 0.98224414 0.90630011 0.77834118 0.44207836 0.00000000
##    1 0.65619765 0.01775586 0.09369989 0.22165882 0.55792164 1.00000000
```

```r
summary(ptabla)
```

```
## Call: xtabs(formula = ~left + number_project)
## Number of cases in table: 6
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 2.9418, df = 5, p-value = 0.709
##  Chi-squared approximation may be incorrect
```

We can see that when the people have more salary, it is more probable that that employee has been promoted.

```r
tabla<-xtabs(~salary+promotion_last_5years)
ptabla<-prop.table(tabla, 1)
ptabla
```

```
##         promotion_last_5years
## salary             0           1
```

```
##   low    0.990978677 0.009021323
##   medium 0.971920571 0.028079429
##   high   0.941794665 0.058205335
```

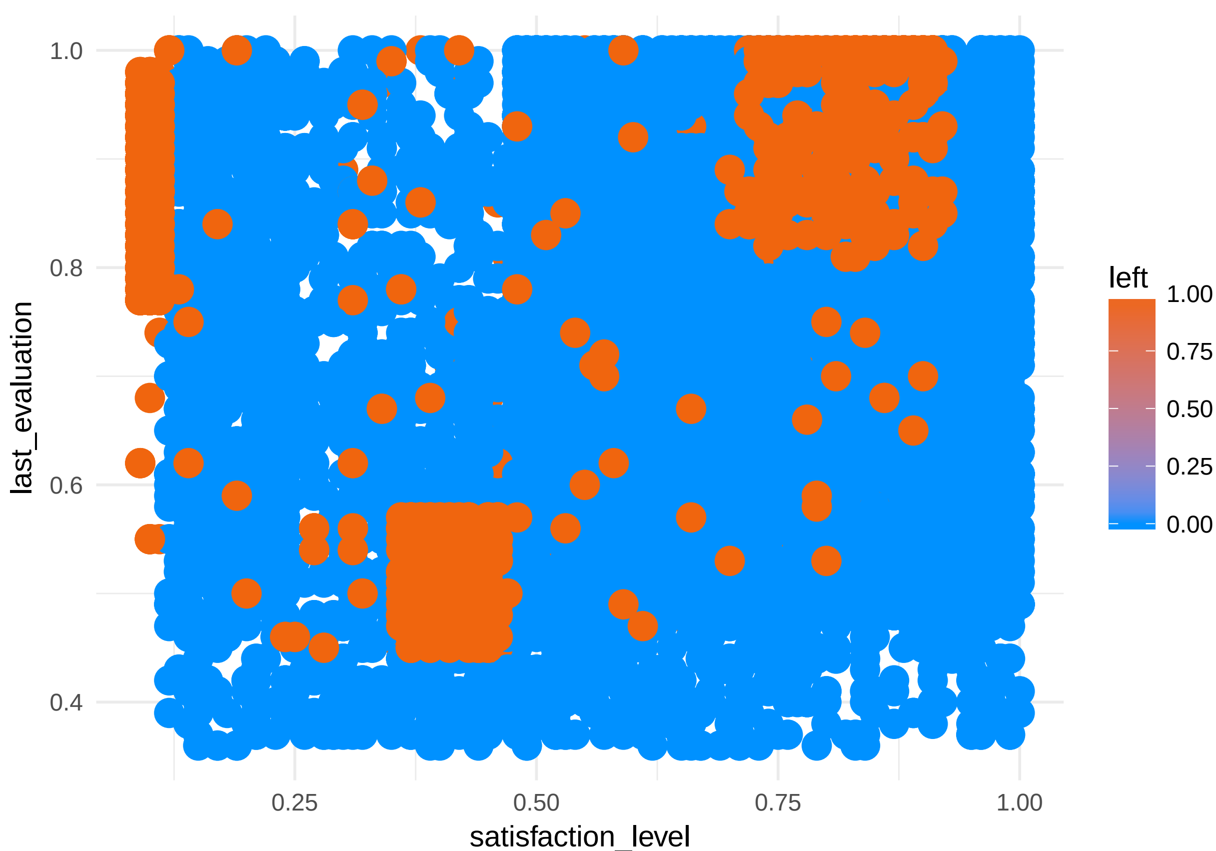
```r
summary(ptabla)
```

```
## Call: xtabs(formula = ~salary + promotion_last_5years)
## Number of cases in table: 3
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 0.03999, df = 2, p-value = 0.9802
##  Chi-squared approximation may be incorrect
```

In none of each analysis we have had a significant effect in the chi-squared analysis.

### 3.4 - Exploration of clusters or leavers

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




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. - Cluster 2 (Bad and Sad Employee): Satisfaction between about 0.35~0.45 and evaluations below ~0.58. - Cluster 3 (Hard-working and Happy Employee): Satisfaction between 0.7~1.0 and evaluations were greater than 0.8.

# 4 - Feature selection with Boruta analysis

Feature importance:

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.

How does it work?

Below is the step wise working of boruta algorithm:

Firstly, it adds randomness to the given data set by creating shu???led copies of all features (which are called shadow features). Then, it trains a random forest classi???er on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.

At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z score than the maximum Z score of its shadow features) and constantly removes features which are deemed highly unimportant.

Finally, the algorithm stops either when all features gets con???rmed or rejected or it reaches a speci???ed limit of random forest runs.

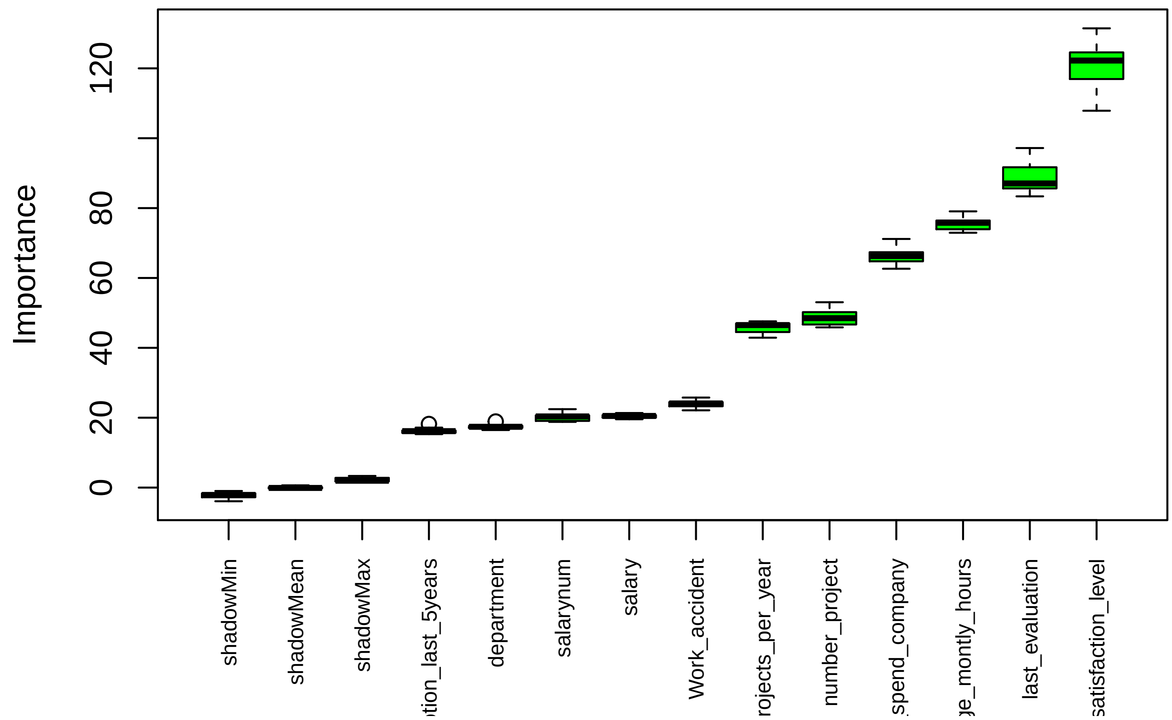
```
# The below code will install the boruta package if it doesn't exist, and then load it
if (!require(Boruta)) install.packages("Boruta")
library(Boruta)
```

The following code will perform the Boruta analysis:

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

print(boruta.train)
```

```
## Boruta performed 11 iterations in 2.47843 mins.
##  11 attributes confirmed important: average_montly_hours,
## department, last_evaluation, number_project, projects_per_year and
## 6 more;
##  No attributes deemed unimportant.
```

```
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)
```




Boruta analysis result: - The 10 variables are deemed important. - The most important variables are satisfaction level, last evaluation and monthly hours worked.

Boruta is an easy to use package as there aren't many parameters to tune / remember. You shouldn't use a data set with missing values to check important variables using Boruta. It'll blatantly throw errors.

A good explanation of Boruta can be found here: www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/