# **Data Science Group Project**

# **Akshay Parab**

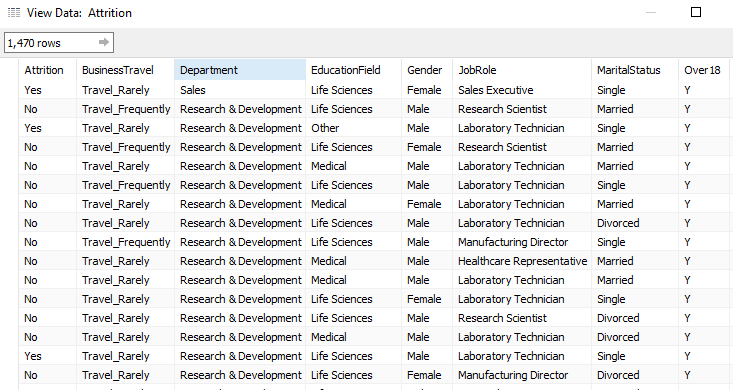
# **Manish Mishra**

# **Manoj Nahak**

# **Royeden Medeira**

# **Data set description**

* The dataset for IBM HR Analytics which contains different attributes of an employee. Employee Number is the primary key.
* **Structure :**1470 observations, 35 variables.



# **Objective**

* To predict which employee is likely to churn



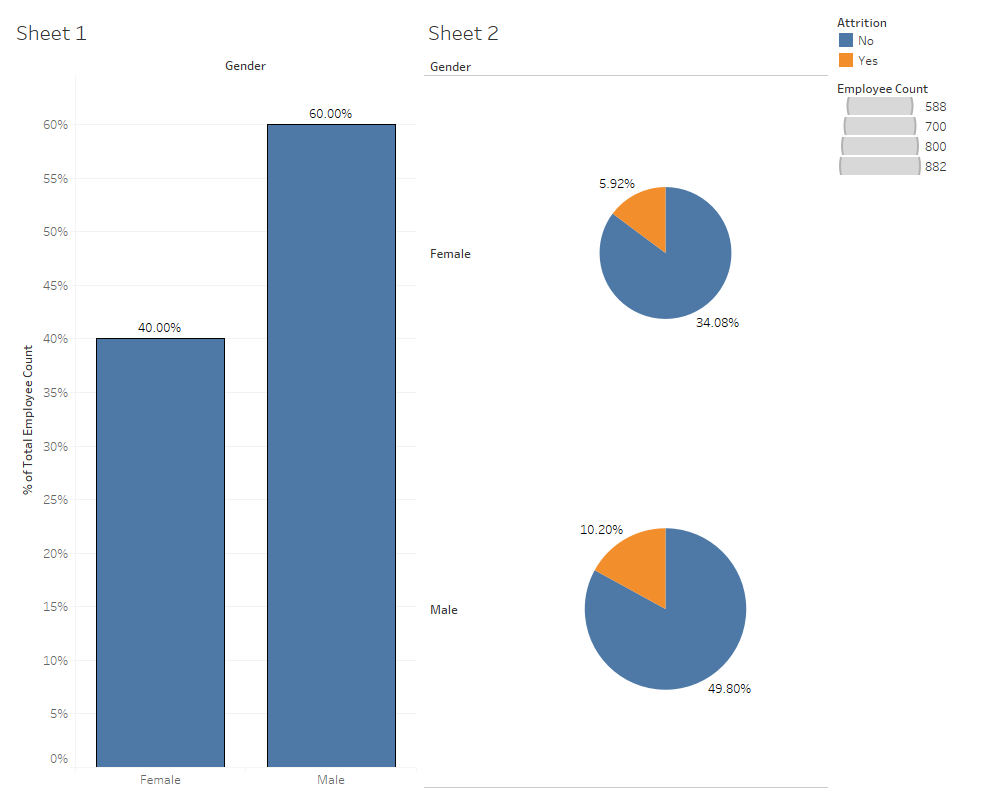
# **Insights**

**Business Travel V/S Attrition**

While a large no. of employees fall under the ‘Travel Rarely’ category (71%), attrition is remarkably high (25%) within the ‘Travel Frequently’ category.

**Gender V/S Attrition**

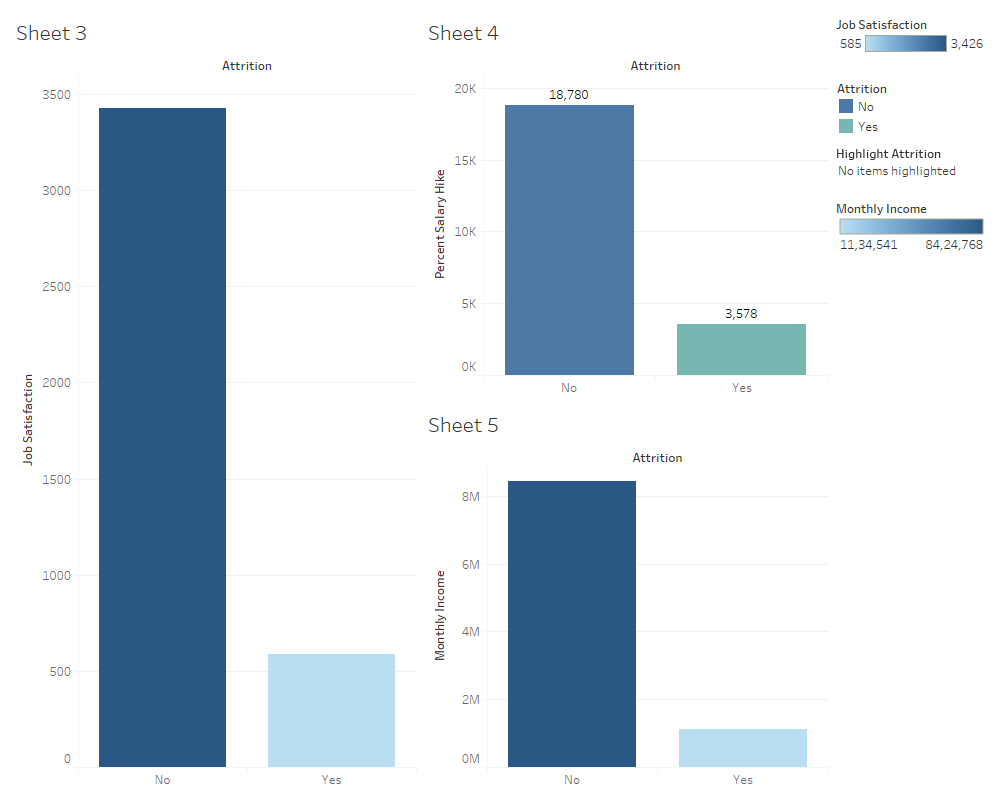
* **The data involves 60% males whereas there were maximum Bachelors and Masters under study.**



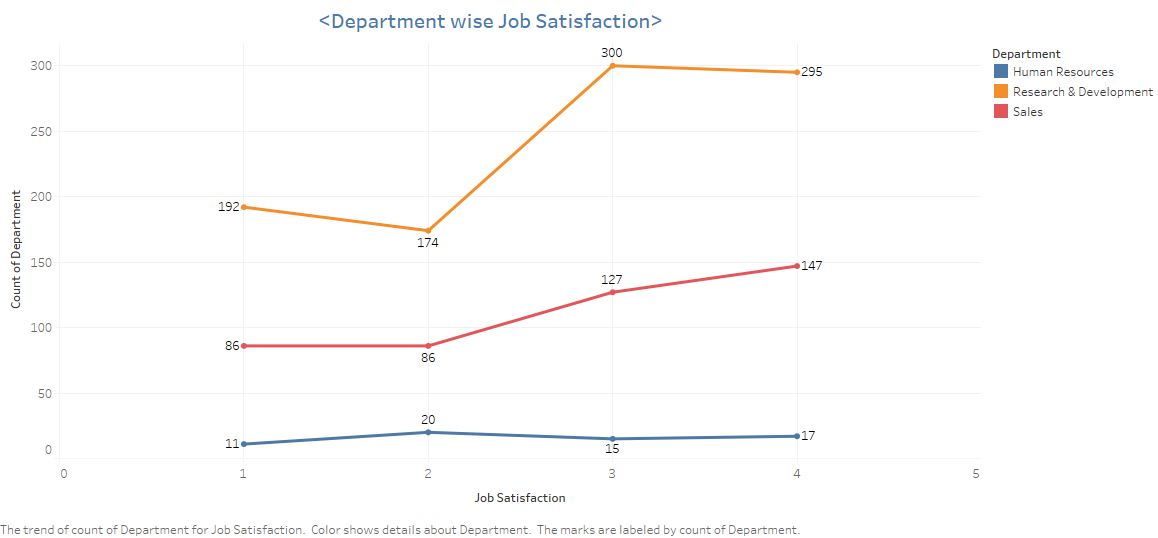
* **Bifurcation of Employees basis on their Education**



* Below we see the impact of different variables like ‘Job Satisfaction’ , ‘monthly income’ and ‘Salary hike’ on Attrition.

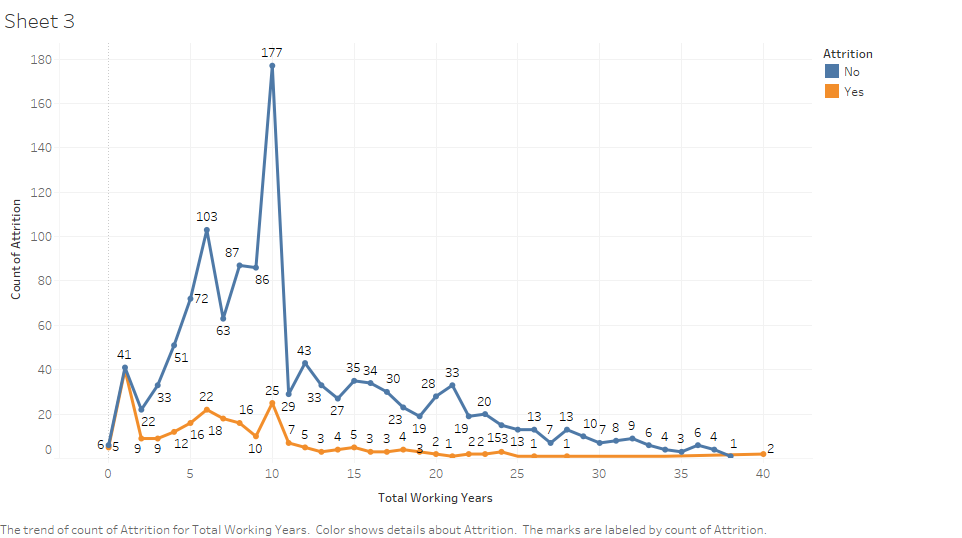


**Department wise Job Satisfaction**



**Work Exp V/S Attrition**

**From the below plots we can say that as Experience increases Attrition becomes steady.**

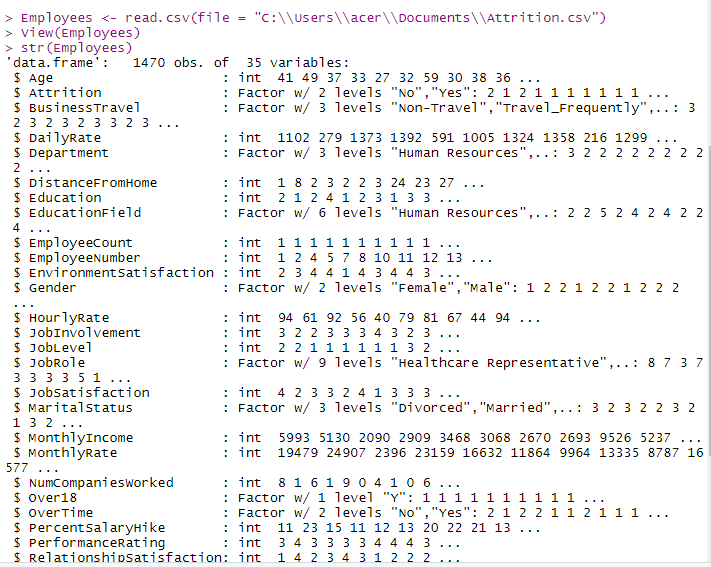


**Overtime V/S Attrition**

As expected, Overtime does have a major impact on attrition levels

# **Analysis in R**

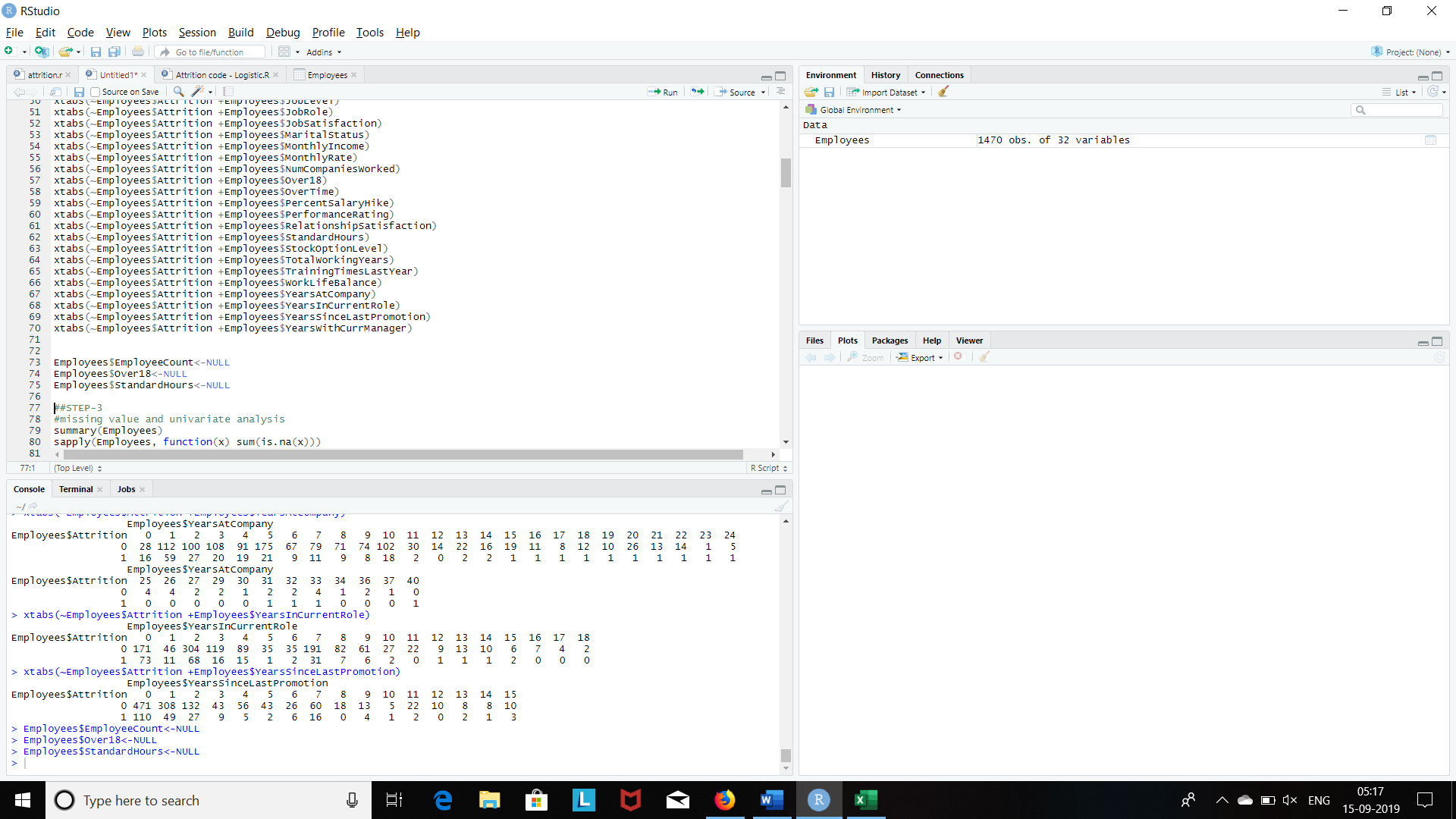
* **Structure of data after Importing in R.**
* **The dataset consists of Categorical and Numerical variables.**
* **Attrition is the target Variable.**

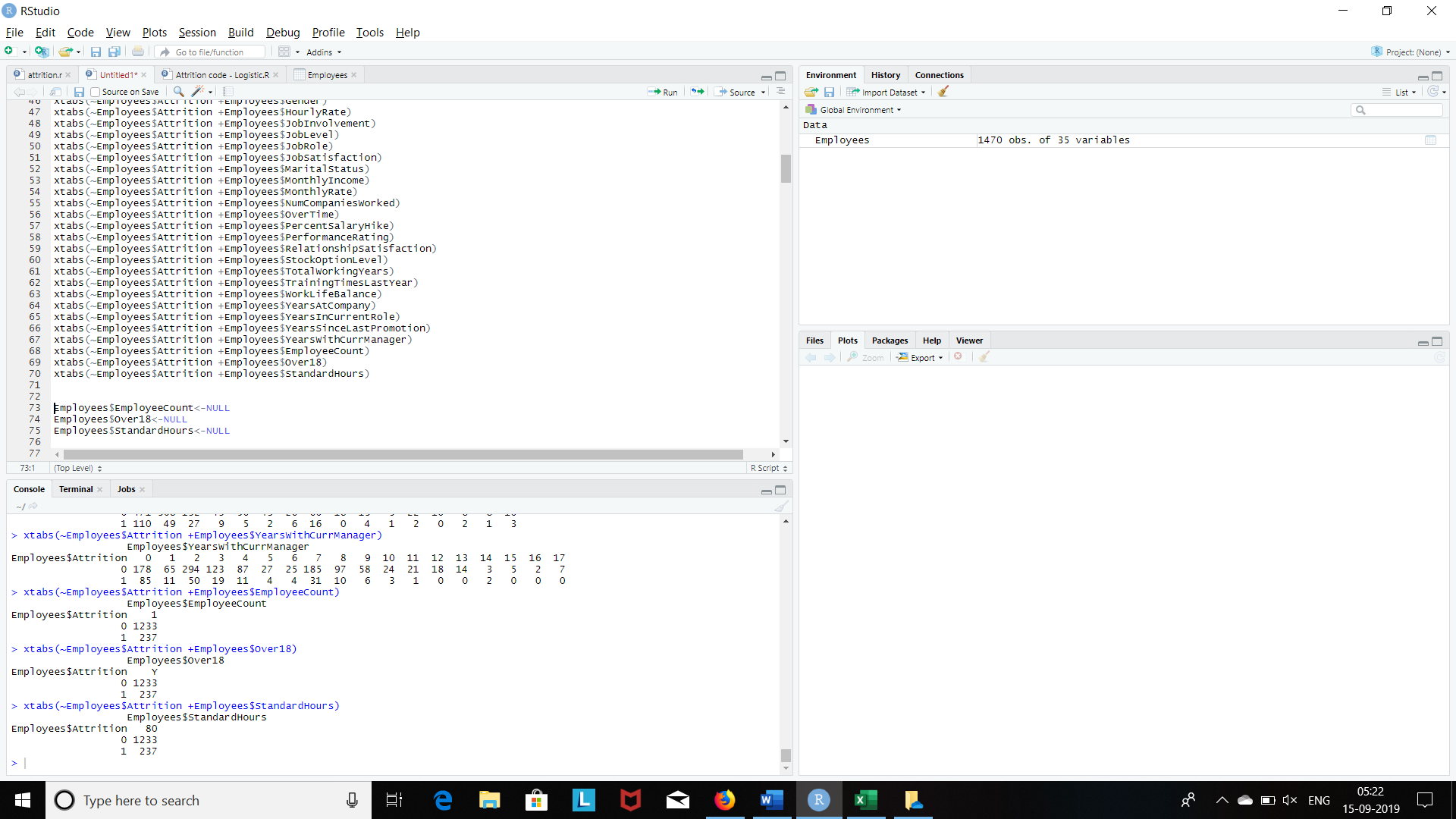


**Feature Selection**

* **To remove variables with zero or less impact.**
* **Using Business logic, ‘table’ function and ‘Xtabs’ function to identify and exclude variables which are not required**

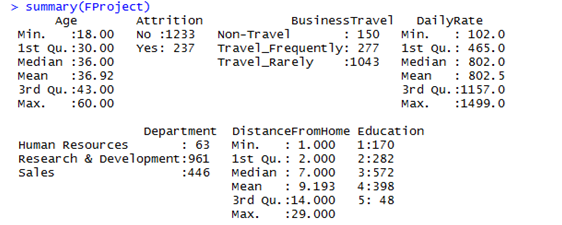
**Dropping the unwanted variables**

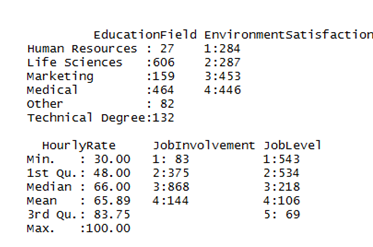




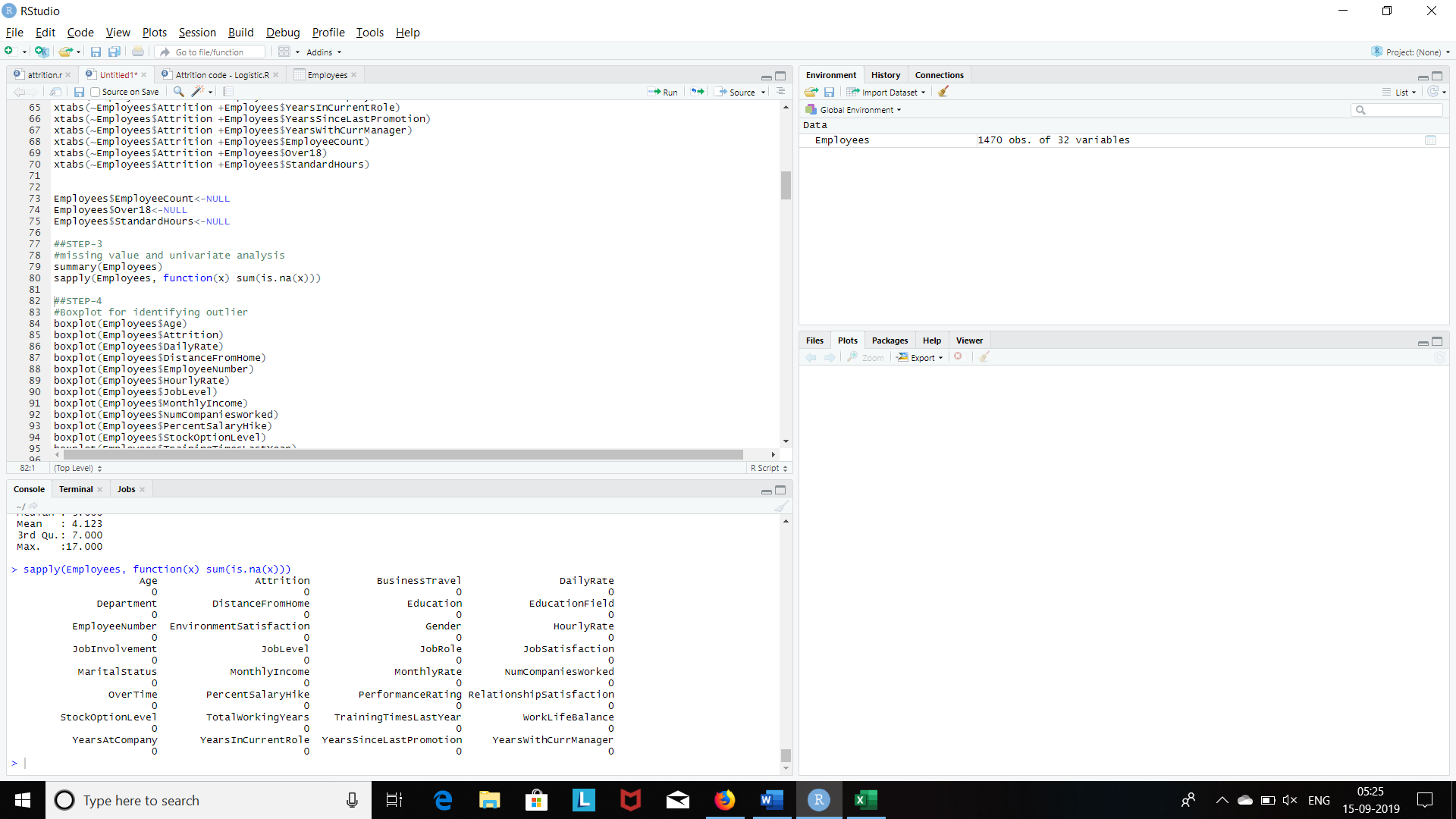
# **Exploratory Data Analysis**

**Summary**





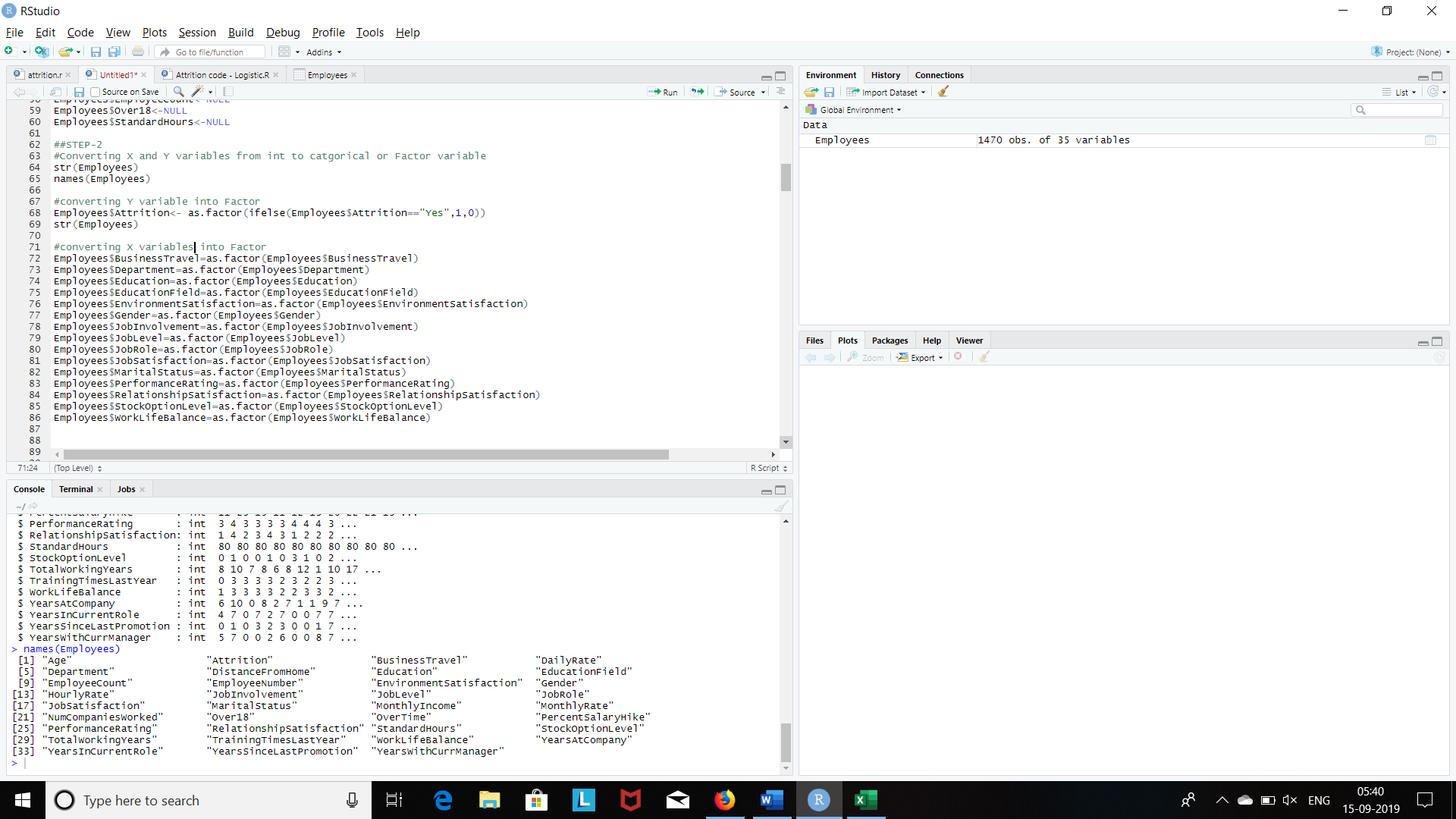
Checking missing Values in the data



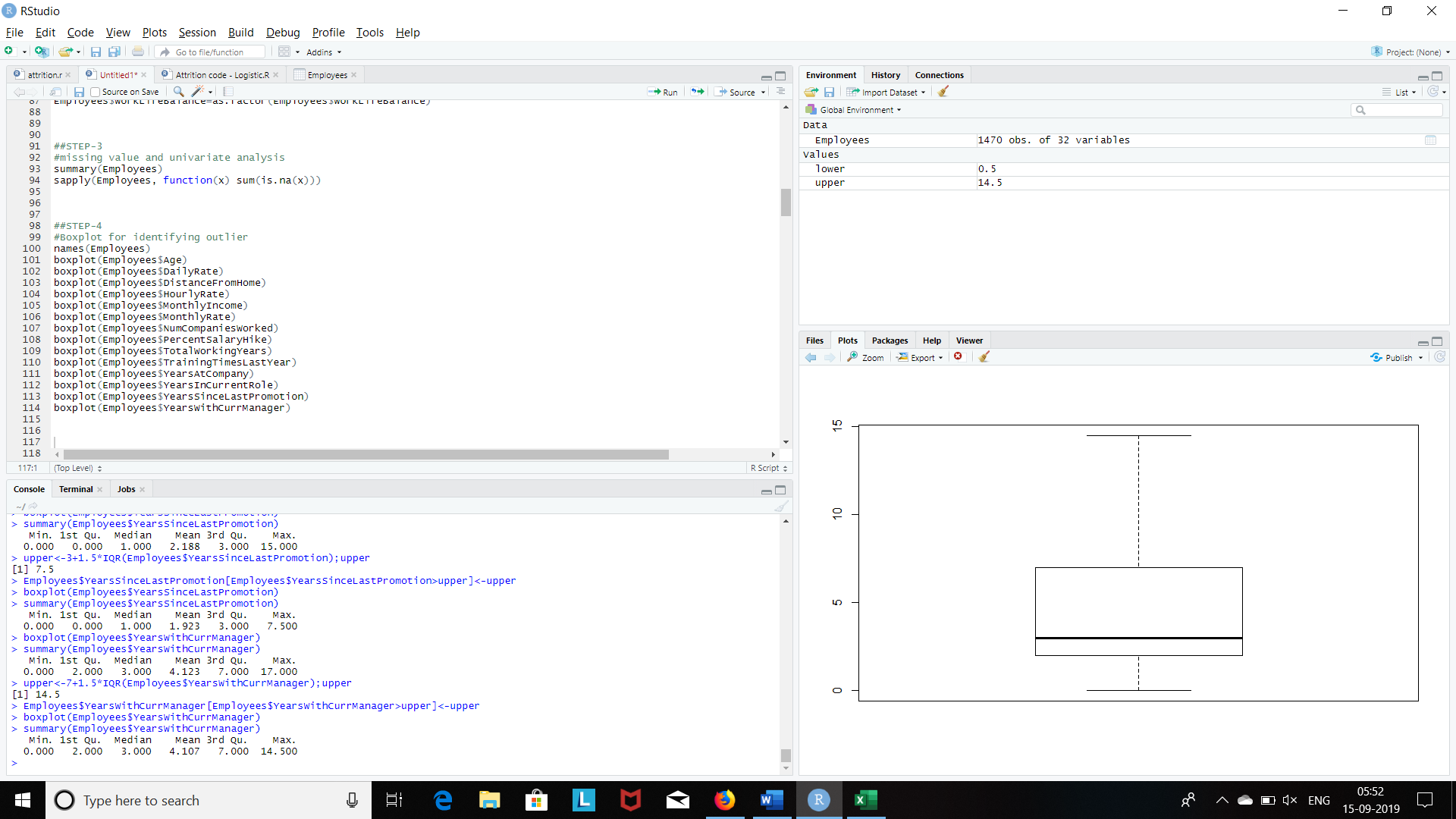
Converting ‘Y’ variable into Factor

* **Employees$Attrition<- as.factor(ifelse(Employees$Attrition=="Yes",1,0))**

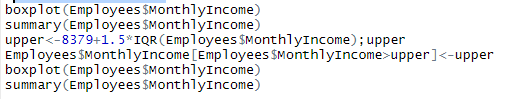
Converting ‘X’ variables from integer to categorical or Factor variable



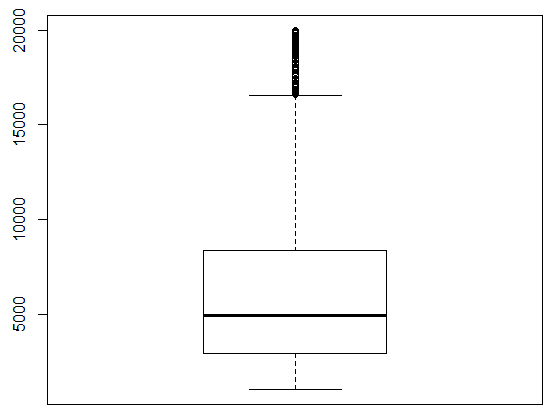
Boxplot for identifying outlier



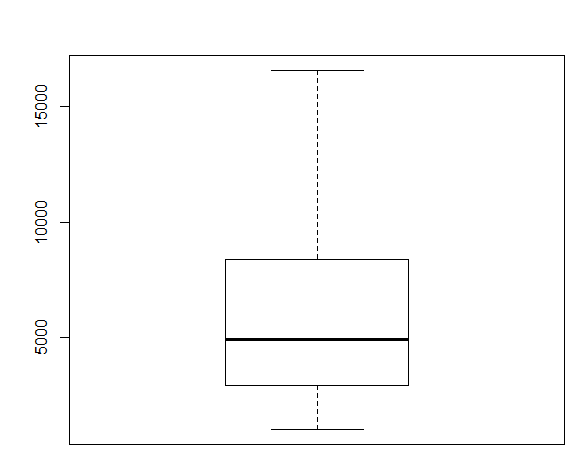
Treatment of outliers



# **Before Treatment**



# **After Treatment**



OBJECTIVE: To study the effect of different factors on Attrition

Logistics regression (Logistic regression is a regression model where the dependent variable is Categorical)

* **In this case we have only two values Attrition within Employees whether Yes / No.**

Y= 1 (Yes)

Y= 0 (No)

**Independent Variables:**

**f1: Gender**

**f2: Age**

**f3: Education**

**f4: Environment Statisfaction**

**f5: JobInvolvement**

**f6: JobSatisfaction**

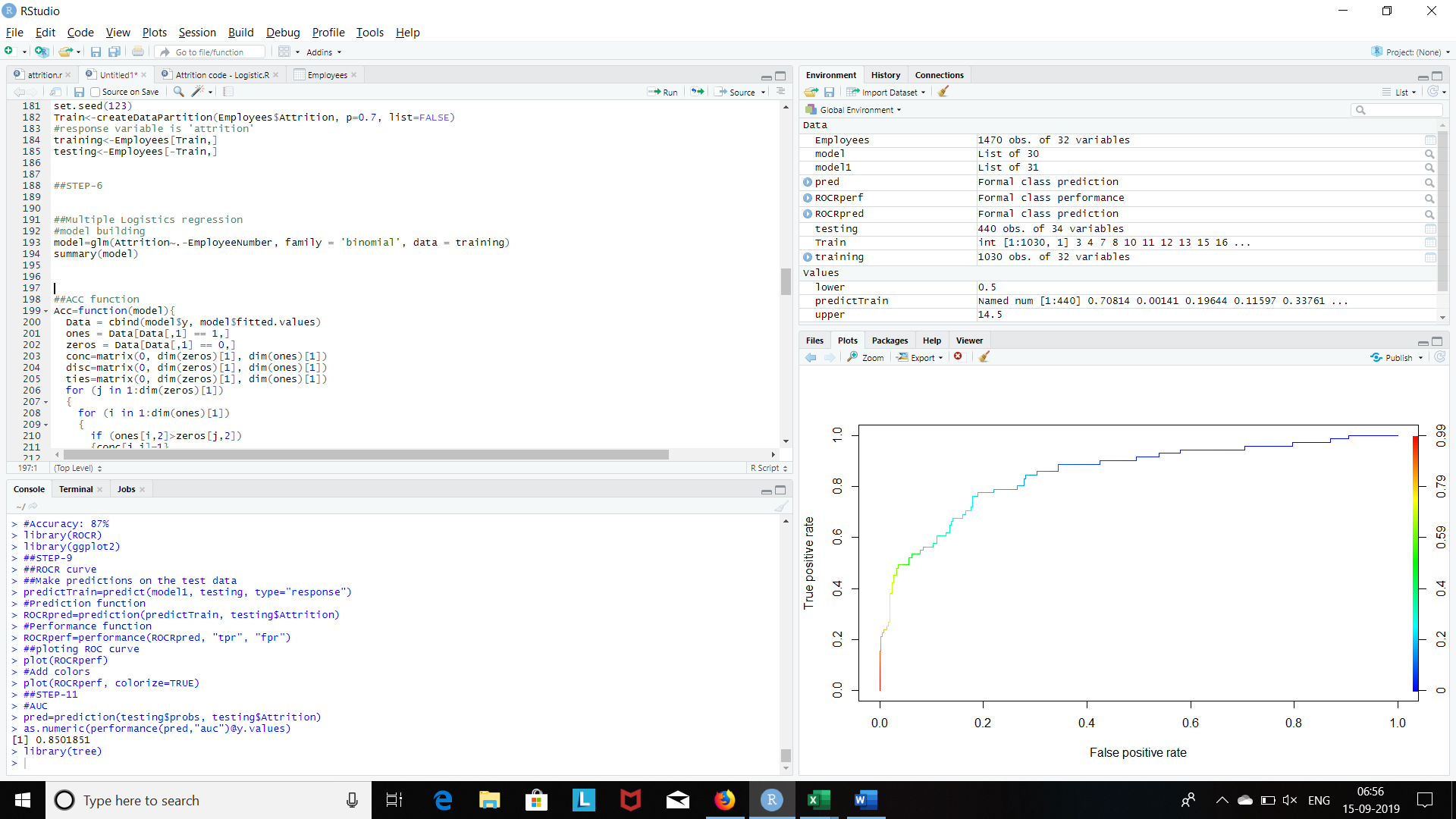
**f7: PerformanceRating**

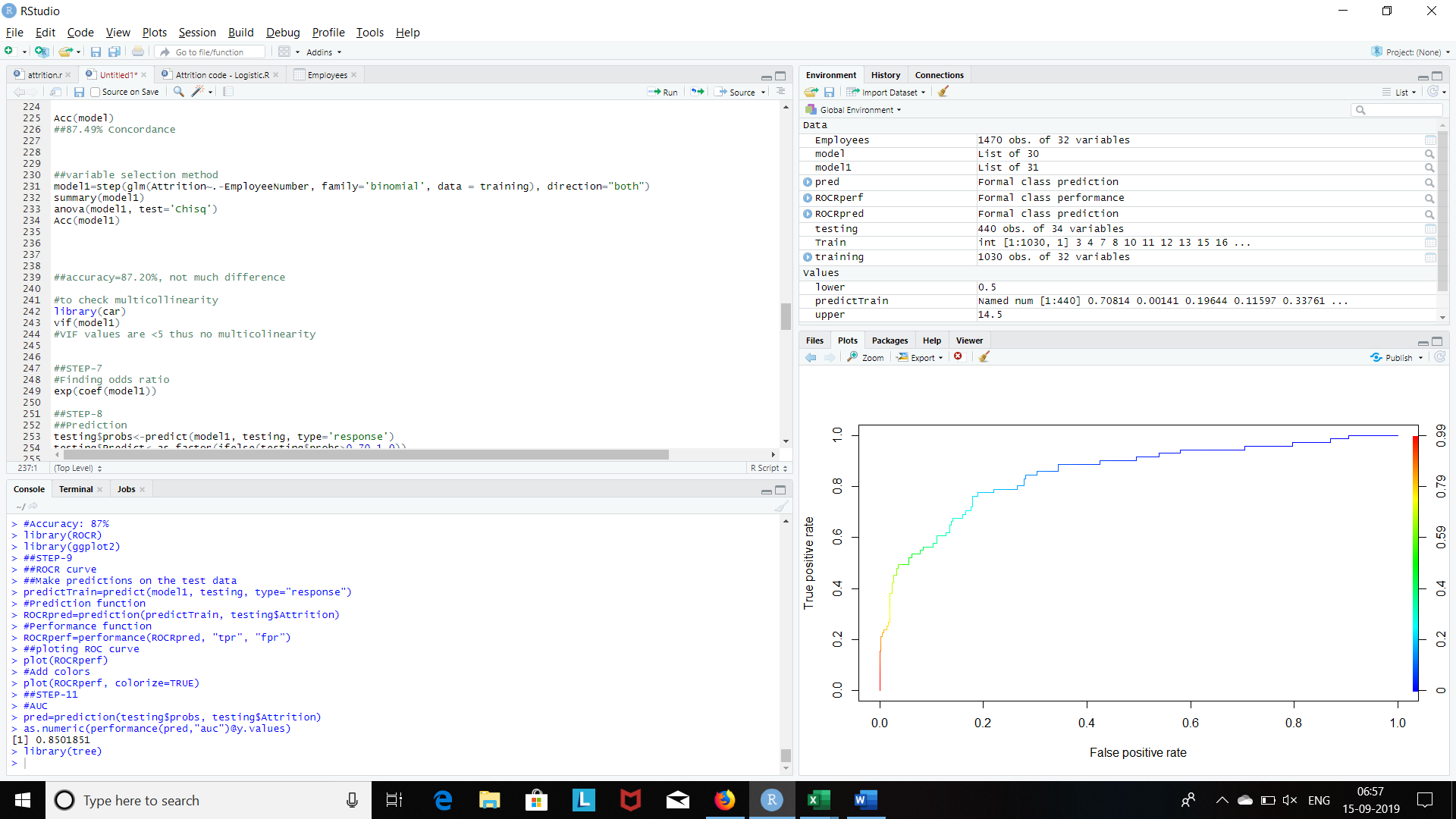
**And 23 more…**

**Steps: -**

* Start with data partition, create Training and testing data.
* Build models
* Compare AIC – lower the better
* Check multicollinearity – VIF values are <5 thus no multicollinearity
* Determining odds ratio
* Check model accuracy using ‘ACC’ function – 89.76 Concordance
* Run model on test data, check accuracy basis confusion matrix  
  - Accuracy: 88.41  
  - Sensitivity: 89.16  
  - Specificity: 79.41
* Check AUC – 85.01  
  (ROC curve i.e. Receiver Operating Characteristic)

**Building the Model**



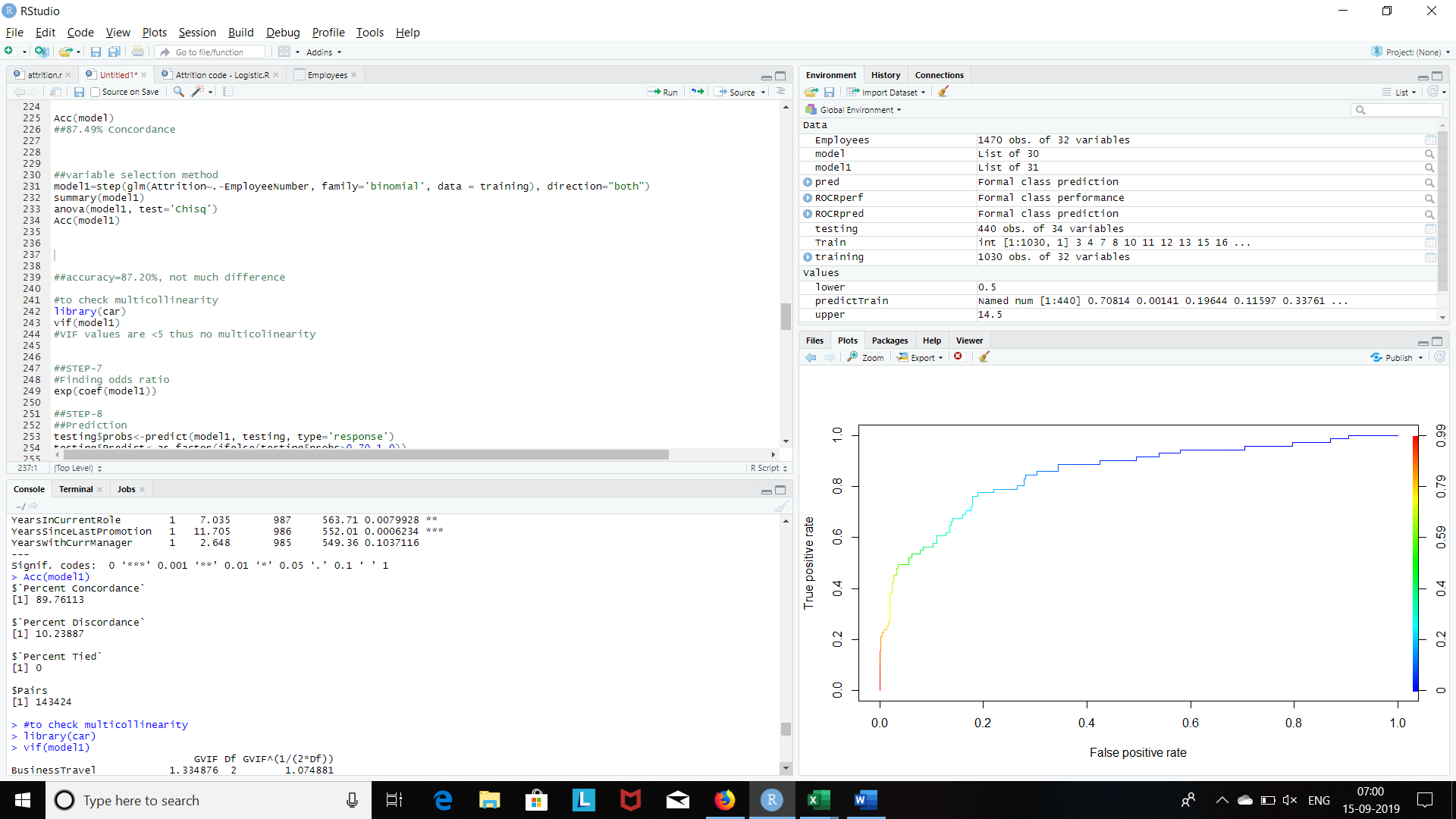


**VARIATION INFLATION FACTOR(VIF): VIF** stands for Variation Inflation Factor. During regression analysis, **VIF** assesses whether factors are correlated to each other (multicollinearity), which could affect p-values and the model isn't going to be as reliable.

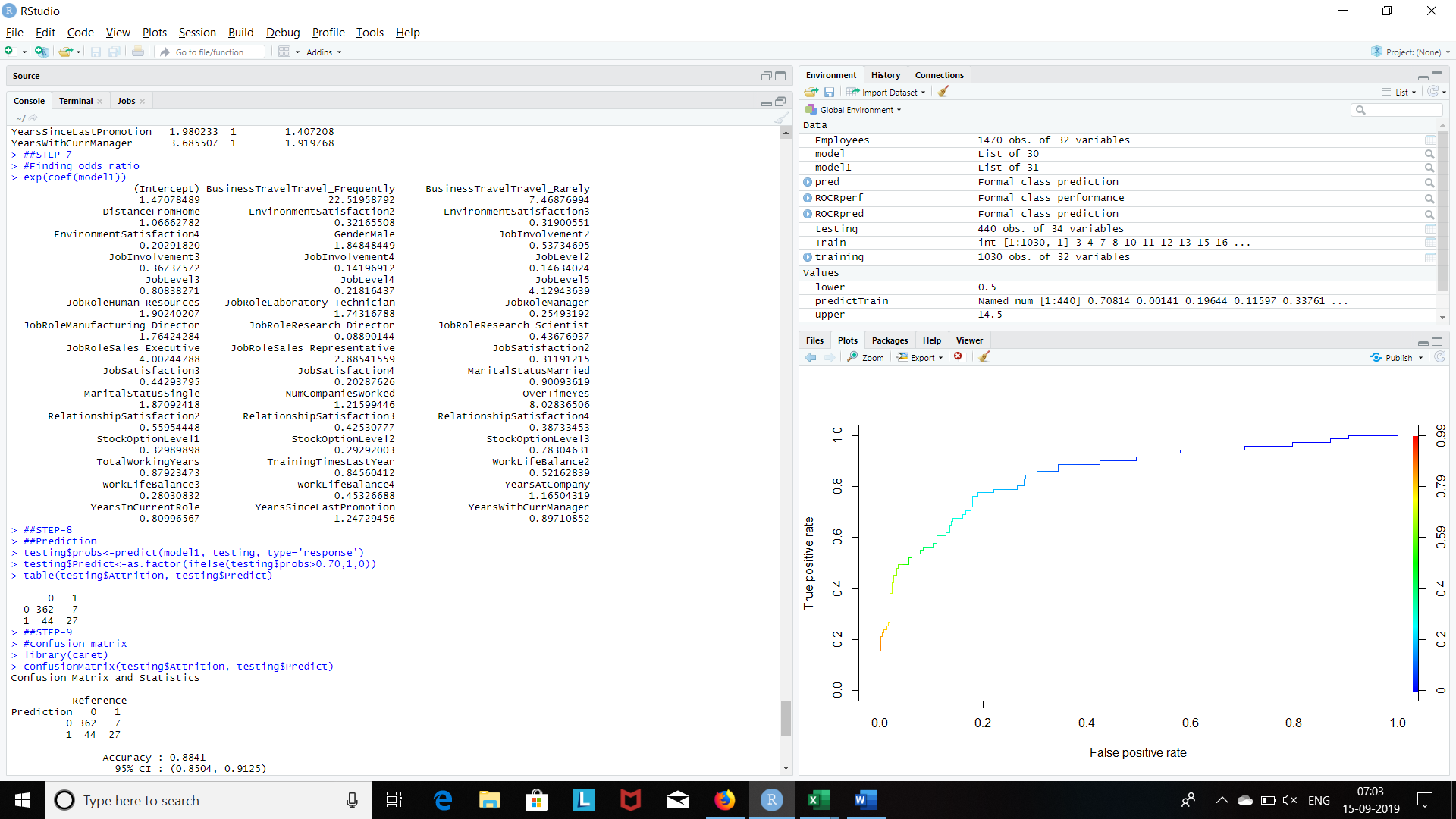


**Concordance and Discordance for Training Model: -**

**Criteria: High Concordance, Low Discordance and low Tied Pair, Better the model**



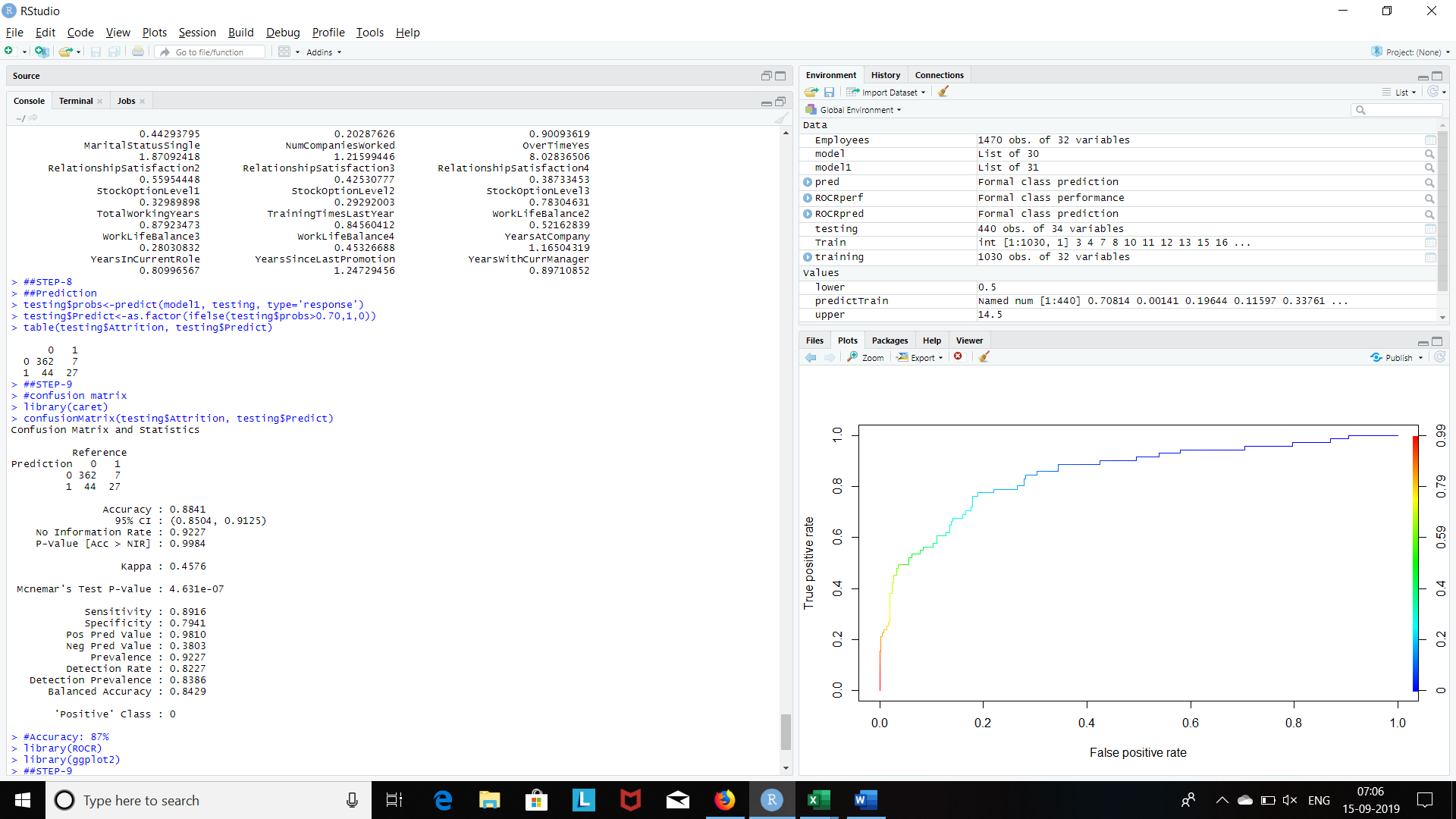
**ODDS RATIO:** The odds ratio measures the strength of association between a predictor and the response variable of interest.



* **Interpretation:** For Female V/S Male: The chances of Males Attrition are 1.85 times higher than that of Females Attrition.

**ACCURACY OF MODEL**

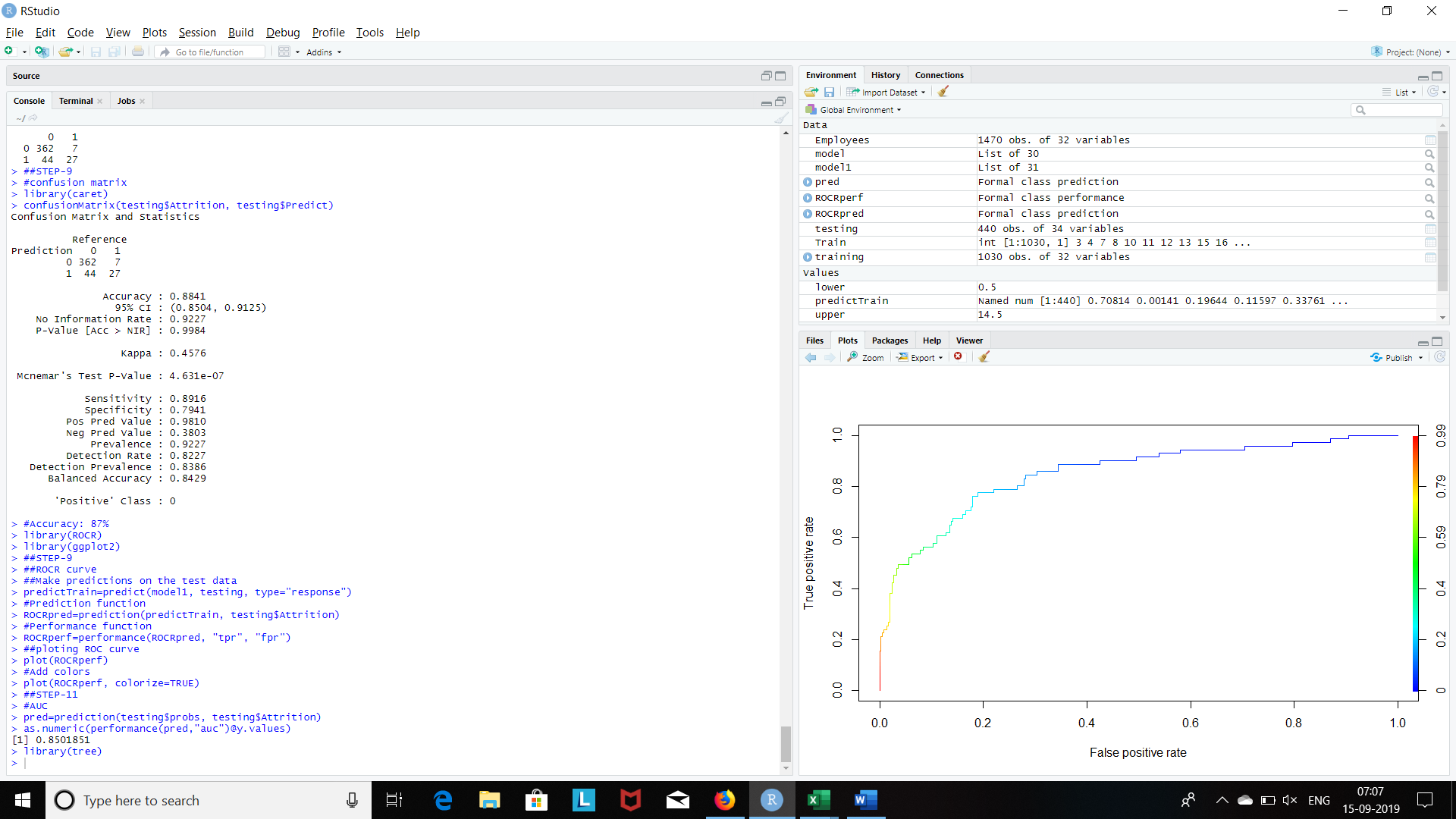
**Confusion Matrix and Statistics**



**Sensitivity: - TPR = TP/TP+FN**

**Specificity: - TNR = TN/TN+FP**

**ROC and AUC of MODEL: 85.02**



**Decision Tree**

Tree based learning algorithms are considered to be one of the best and mostly used supervised learning methods.

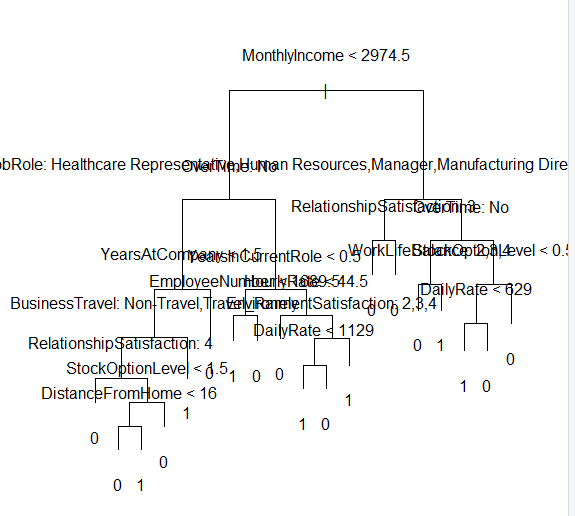
Decision tree is used in both the scenario regression and classification, depending on the data type of “Y” variable.

In this technique we split the populations or sample into two or more homogeneous sets based on the most significant splitter.

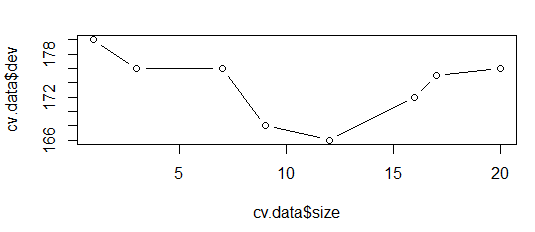
There are Four methods to identify most significant splitter.

1. Entropy
2. Information Gain
3. Gini Index
4. Chisquare Method (CHAID)

**Plotting the Decision Tree**



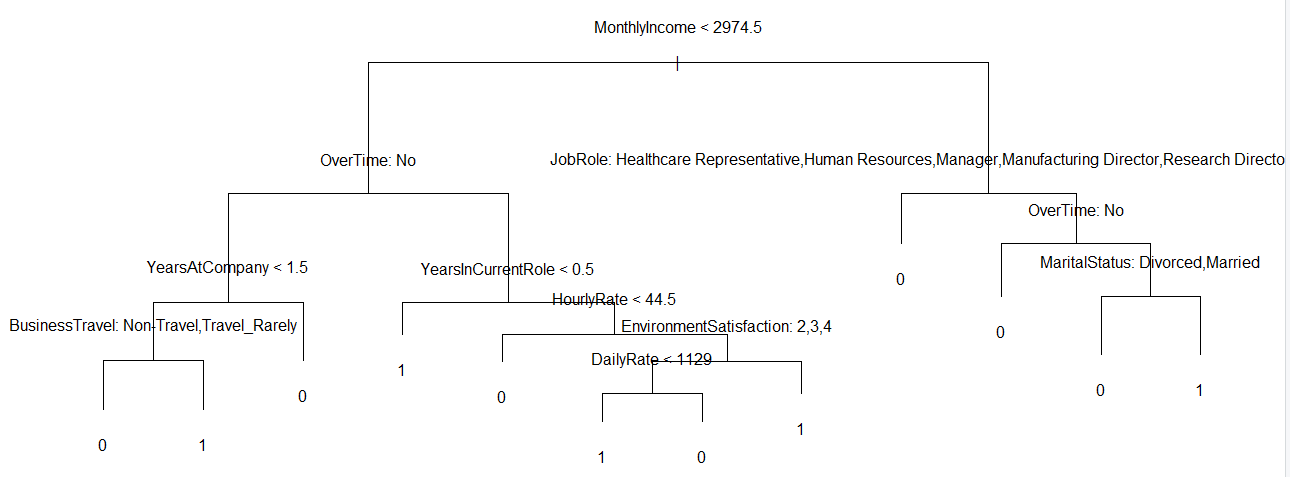
**Hence the data nodes are more, before prediction first we have to prune the data based on below graph.**

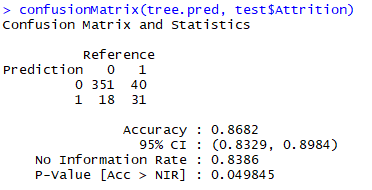


**prune.data = prune.misclass (tree.model, best = 12)**

**#misclass is used when y is classification**

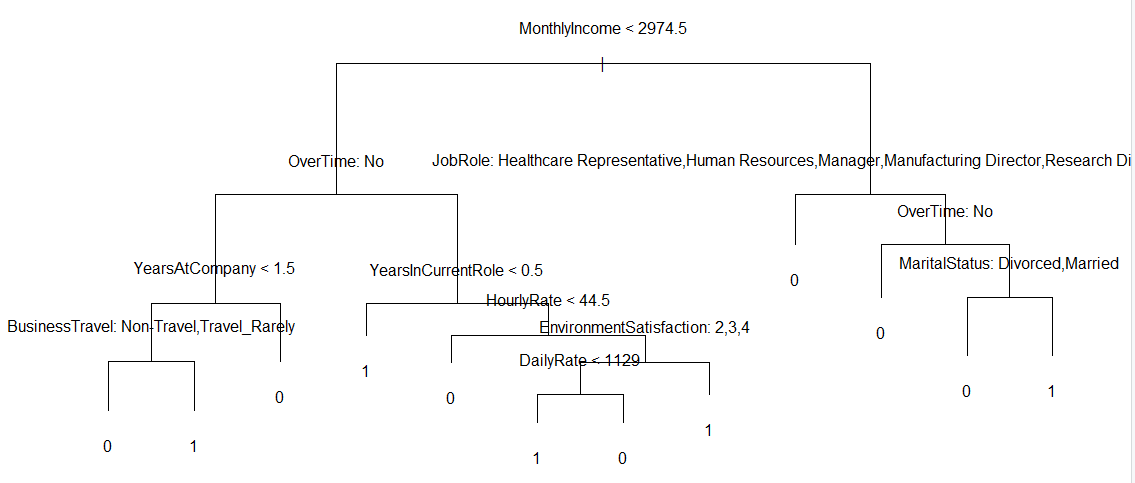
**Post pruning the Deceision tree have less nodes**





**Pruning with 11 Nodes**

**prune.data1 = prune.misclass (tree.model, best = 11)**



**SVM**

**SVM (Support Vector Machine)**

* SVM is a supervised learning algorithm. Which can be used for both classification and regression challenges
* In this algorithm we can plot each data item as a point in n dimensional space
* (where n is number of feature you have)
* Then we can perform classification by finding the hyperplane that differentiate two classes

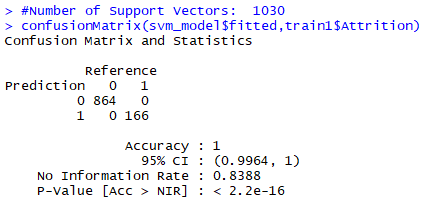
**Support vector –**

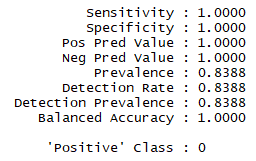
* Support vectors are the data point nearest to hyperplane

**Margin –**

* Distance between support vector and hyperplane from both side
* We select the hyperplane which have maximum margin so that out data should be properly classified.

**#building model1 without cost**





**#building model2 with cost**

svm\_model2 <- svm(Attrition~., data=train1, cost=0.01, scale = FALSE)

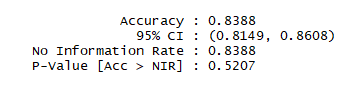
summary(svm\_model2)

#Number of Support Vectors: 1030

confusionMatrix(svm\_model2$fitted,train1$Attrition)

#Accuracy = 0.8388

**Accuracy when cost = 0.01**



**#building model3 with Auto selection of cost**

**It is best for the model**

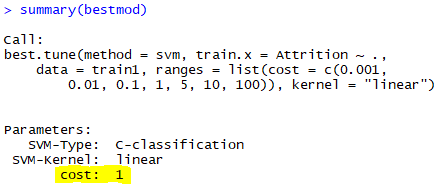
**tune.out<-tune(svm,Attrition~.,data=train1, kernel="linear",**

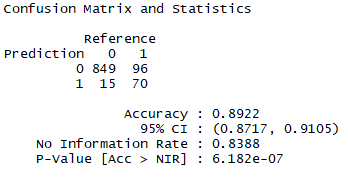
**ranges = list(cost=c(0.001,0.01,0.1,1,5,10,100)))**

**summary(tune.out)**

**bestmod=tune.out$best.model**

**summary(bestmod)**





**#Accuracy = 0.8922**

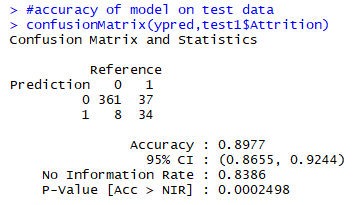
**#prediction on testing data**

**ypred=predict(bestmod,test1)**

**table(predict=ypred,truth=test1$Attrition)**

**#accuracy of model on test data**

**confusionMatrix(ypred,test1$Attrition)**



**#Accuracy = 0.8977**