Employee Attrition Analysis

Research Question use data analytics to analyze an IBM employee dataset to determine the variables that affect attrition

EDA

Dataset

```
# load the data
data <- read.csv("C:/Users/mfbro/Downloads/WA_Fn-UseC_-HR-Employee-Attrition.csv", header=TRUE, sep=",")
data$Age <- data$ï..Age
data <- data[-1]
```

Packages

```
# install .packages("corrplot")
# install.packages("mctest")
# install.packages("car")
# install.packages("ROCR")
# install.packages("rpart")
# install.packages("randomForest")
# install.packages("caret")
#install.packages("DMwR")
# install.packages("mlbench")
```

Summary Stats

```
str(data)
```

```
## 'data.frame':    1470 obs. of  35 variables:
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 3 2 3 2 3 3
##  $ DailyRate               : int  1102 279 1373 1392 591 1005 1324 1358 216 1299 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ DistanceFromHome        : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ Education               : int  2 1 2 4 1 2 3 1 3 3 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ EmployeeCount           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeNumber          : int  1 2 4 5 7 8 10 11 12 13 ...
##  $ EnvironmentSatisfaction : int  2 3 4 4 1 4 3 4 4 3 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...
##  $ HourlyRate              : int  94 61 92 56 40 79 81 67 44 94 ...
##  $ JobInvolvement          : int  3 2 2 3 3 3 4 3 2 3 ...
```

```
##  $ JobLevel                : int  2 2 1 1 1 1 1 1 3 2 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1
##  $ JobSatisfaction         : int  4 2 3 3 2 4 1 3 3 3 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...
##  $ MonthlyIncome           : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ MonthlyRate             : int  19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...
##  $ NumCompaniesWorked      : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ Over18                  : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OverTime                : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 2 1 1 1 ...
##  $ PercentSalaryHike       : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ PerformanceRating       : int  3 4 3 3 3 3 4 4 4 3 ...
##  $ RelationshipSatisfaction: int  1 4 2 3 4 3 1 2 2 2 ...
##  $ StandardHours           : int  80 80 80 80 80 80 80 80 80 80 ...
##  $ StockOptionLevel        : int  0 1 0 0 1 0 3 1 0 2 ...
##  $ TotalWorkingYears       : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear   : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ WorkLifeBalance         : int  1 3 3 3 3 2 2 3 3 2 ...
##  $ YearsAtCompany          : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsInCurrentRole      : int  4 7 0 7 2 7 0 0 7 7 ...
##  $ YearsSinceLastPromotion : int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager    : int  5 7 0 0 2 6 0 0 8 7 ...
##  $ Age                     : int  41 49 37 33 27 32 59 30 38 36 ...
```

Structure

```r
str(data)
```

```
## 'data.frame':    1470 obs. of  35 variables:
##  $ Attrition               : Factor w/ 2 levels "No","Yes": 2 1 2 1 1 1 1 1 1 1 ...
##  $ BusinessTravel          : Factor w/ 3 levels "Non-Travel","Travel_Frequently",..: 3 2 3 2 3 2 3 3
##  $ DailyRate               : int  1102 279 1373 1392 591 1005 1324 1358 216 1299 ...
##  $ Department              : Factor w/ 3 levels "Human Resources",..: 3 2 2 2 2 2 2 2 2 2 ...
##  $ DistanceFromHome        : int  1 8 2 3 2 2 3 24 23 27 ...
##  $ Education               : int  2 1 2 4 1 2 3 1 3 3 ...
##  $ EducationField          : Factor w/ 6 levels "Human Resources",..: 2 2 5 2 4 2 4 2 2 4 ...
##  $ EmployeeCount           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ EmployeeNumber          : int  1 2 4 5 7 8 10 11 12 13 ...
##  $ EnvironmentSatisfaction : int  2 3 4 4 1 4 3 4 4 3 ...
##  $ Gender                  : Factor w/ 2 levels "Female","Male": 1 2 2 1 2 2 1 2 2 2 ...
##  $ HourlyRate              : int  94 61 92 56 40 79 81 67 44 94 ...
##  $ JobInvolvement          : int  3 2 2 3 3 3 4 3 2 3 ...
##  $ JobLevel                : int  2 2 1 1 1 1 1 1 3 2 ...
##  $ JobRole                 : Factor w/ 9 levels "Healthcare Representative",..: 8 7 3 7 3 3 3 3 5 1
##  $ JobSatisfaction         : int  4 2 3 3 2 4 1 3 3 3 ...
##  $ MaritalStatus           : Factor w/ 3 levels "Divorced","Married",..: 3 2 3 2 2 3 2 1 3 2 ...
##  $ MonthlyIncome           : int  5993 5130 2090 2909 3468 3068 2670 2693 9526 5237 ...
##  $ MonthlyRate             : int  19479 24907 2396 23159 16632 11864 9964 13335 8787 16577 ...
##  $ NumCompaniesWorked      : int  8 1 6 1 9 0 4 1 0 6 ...
##  $ Over18                  : Factor w/ 1 level "Y": 1 1 1 1 1 1 1 1 1 1 ...
##  $ OverTime                : Factor w/ 2 levels "No","Yes": 2 1 2 2 1 1 2 1 1 1 ...
##  $ PercentSalaryHike       : int  11 23 15 11 12 13 20 22 21 13 ...
##  $ PerformanceRating       : int  3 4 3 3 3 3 4 4 4 3 ...
##  $ RelationshipSatisfaction: int  1 4 2 3 4 3 1 2 2 2 ...
```

```
##  $ StandardHours           : int  80 80 80 80 80 80 80 80 80 80 ...
##  $ StockOptionLevel        : int  0 1 0 0 1 0 3 1 0 2 ...
##  $ TotalWorkingYears       : int  8 10 7 8 6 8 12 1 10 17 ...
##  $ TrainingTimesLastYear   : int  0 3 3 3 3 2 3 2 2 3 ...
##  $ WorkLifeBalance         : int  1 3 3 3 3 2 2 3 3 2 ...
##  $ YearsAtCompany          : int  6 10 0 8 2 7 1 1 9 7 ...
##  $ YearsInCurrentRole      : int  4 7 0 7 2 7 0 0 7 7 ...
##  $ YearsSinceLastPromotion : int  0 1 0 3 2 3 0 0 1 7 ...
##  $ YearsWithCurrManager    : int  5 7 0 0 2 6 0 0 8 7 ...
##  $ Age                     : int  41 49 37 33 27 32 59 30 38 36 ...
```

Missing data

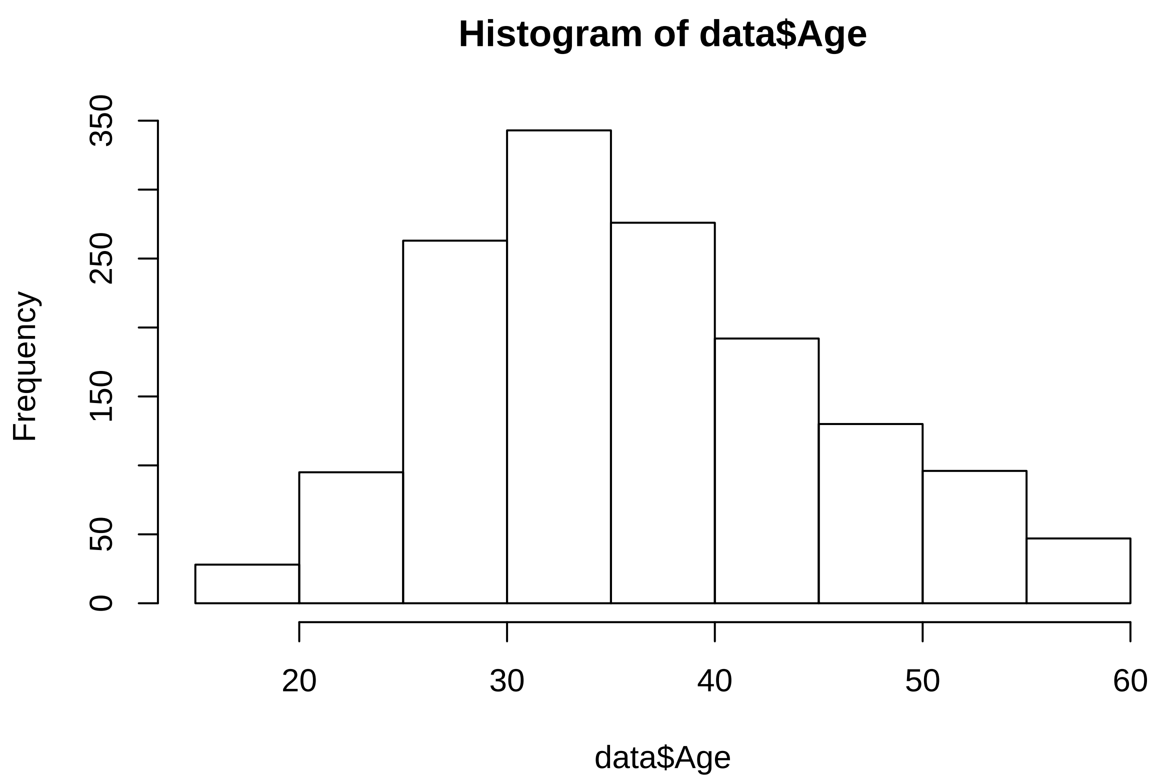
```
## [1] 0
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

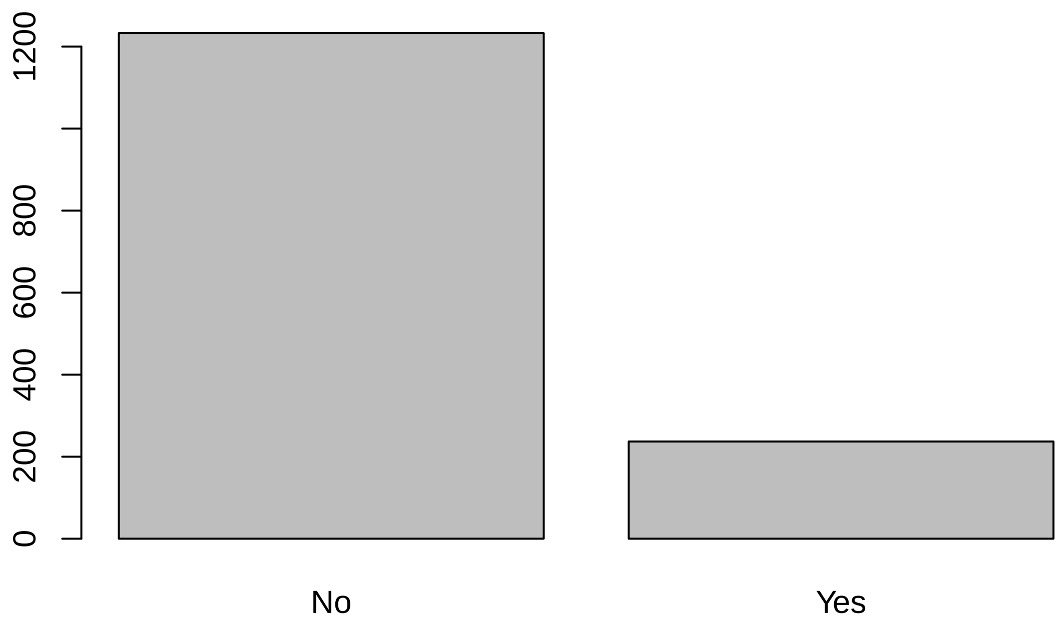
Analysis of Variables

Analyze each variable

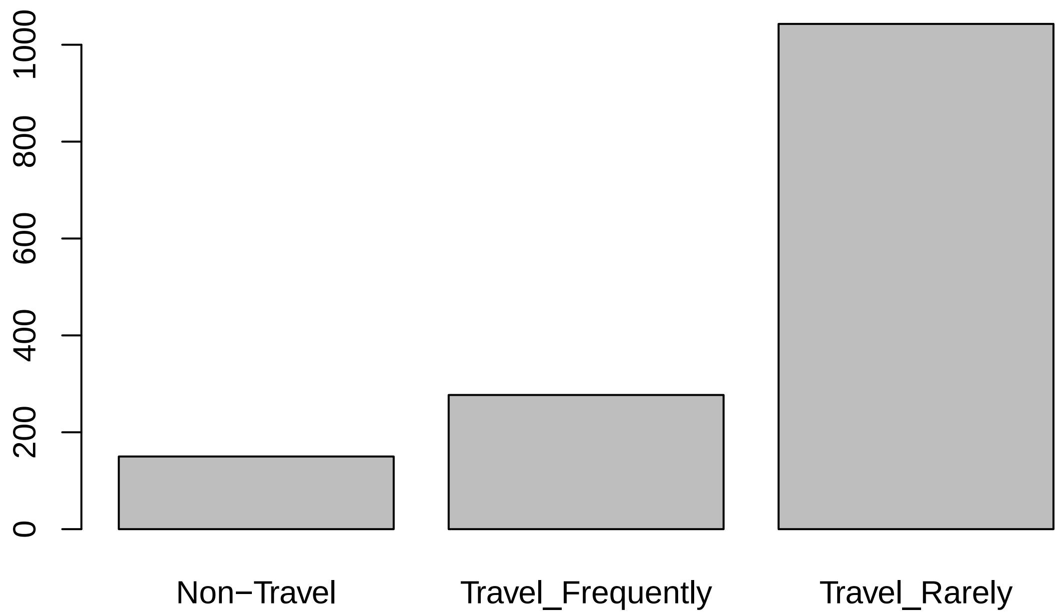
```r
hist(data$Age)
```



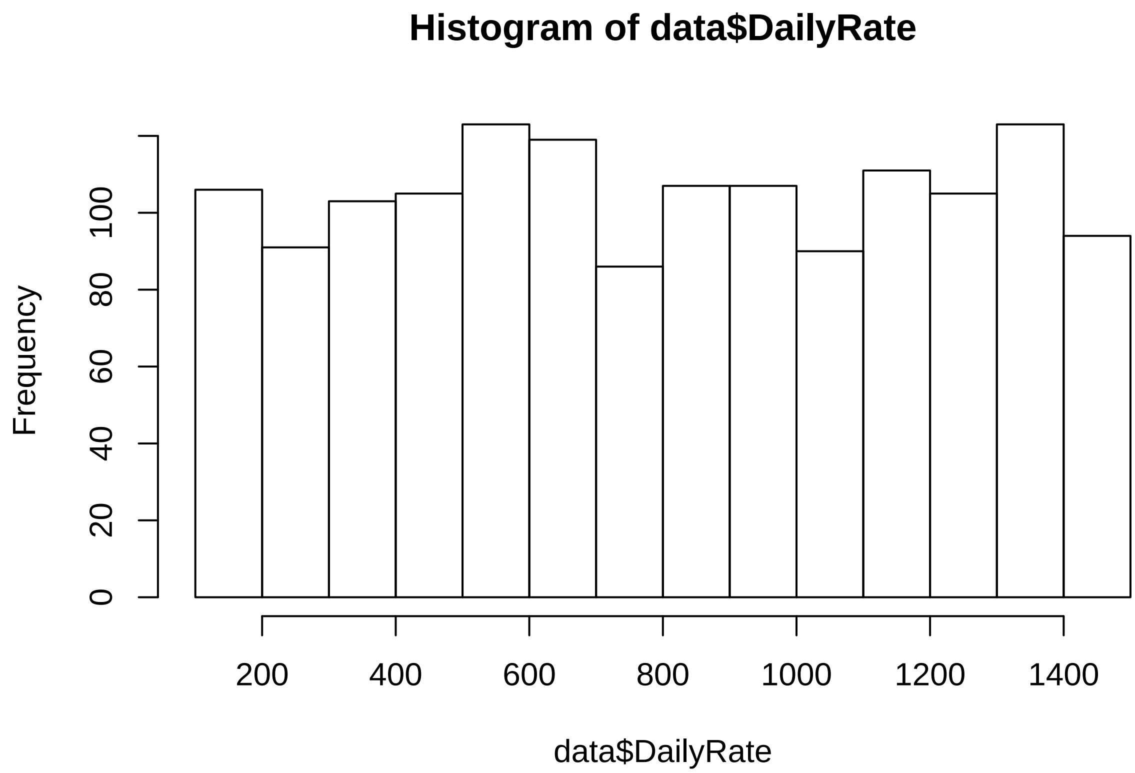
```r
plot(data$Attrition)
```



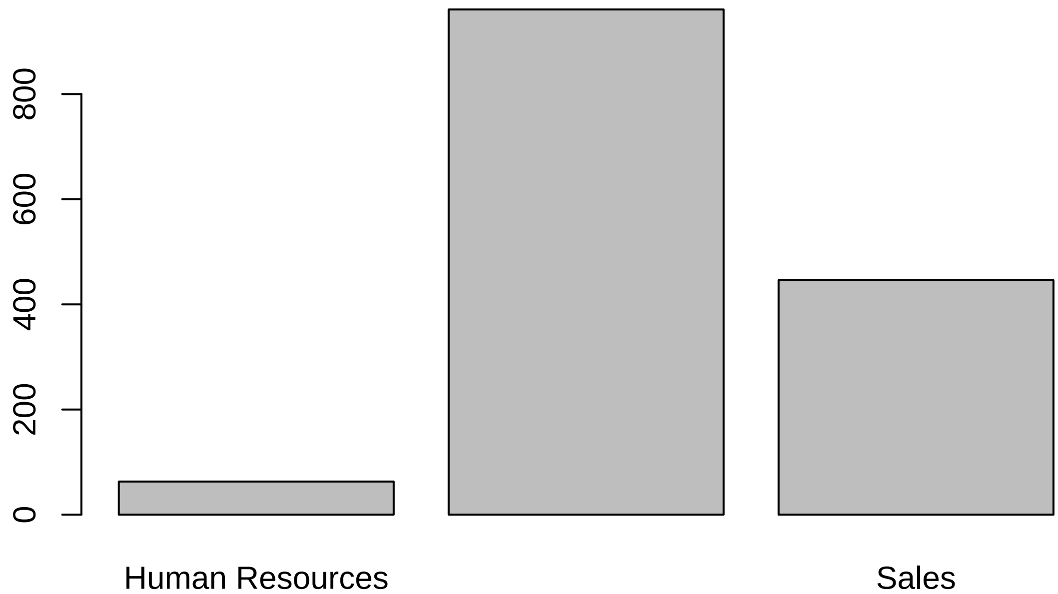

```r
plot(data$BusinessTravel)
```



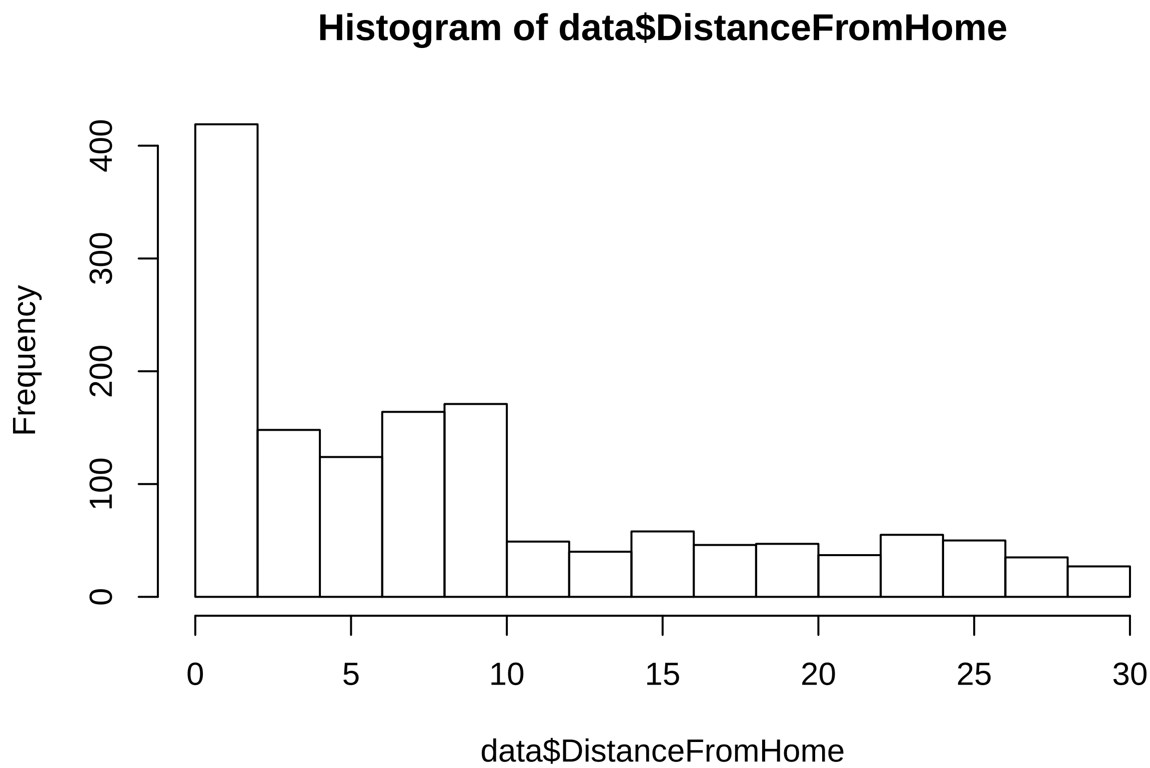

```
hist(data$DailyRate)
```



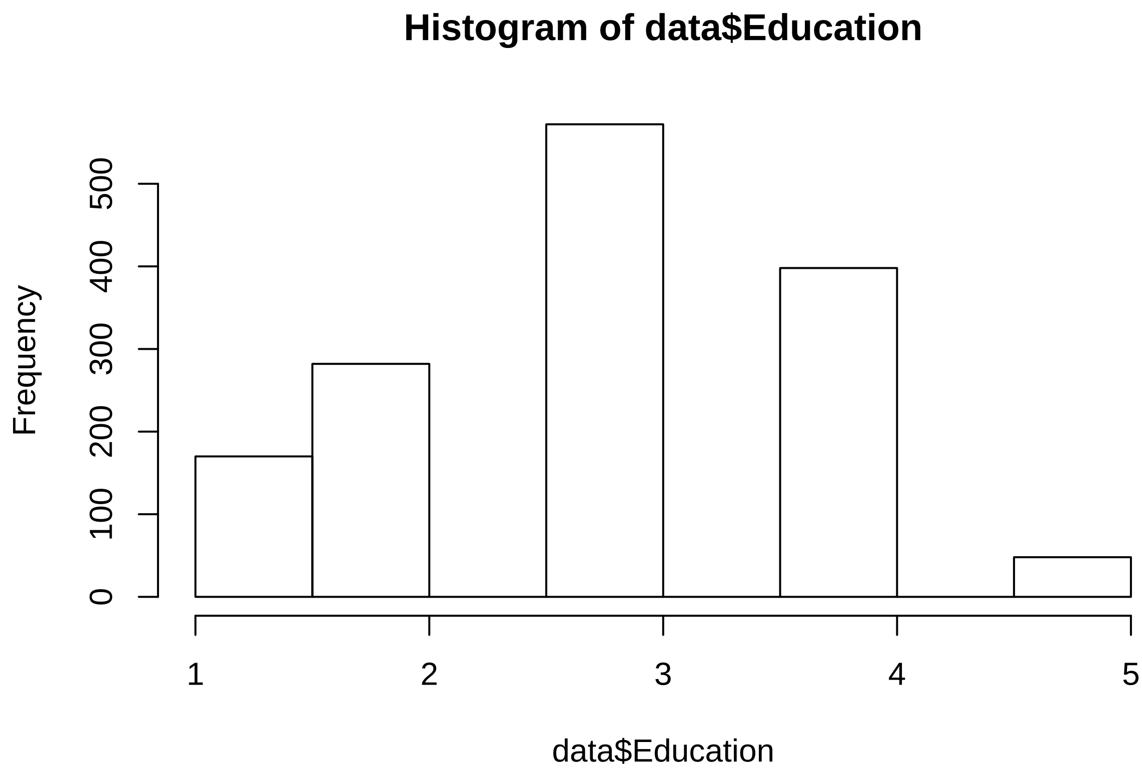

```r
plot(data$Department)
```



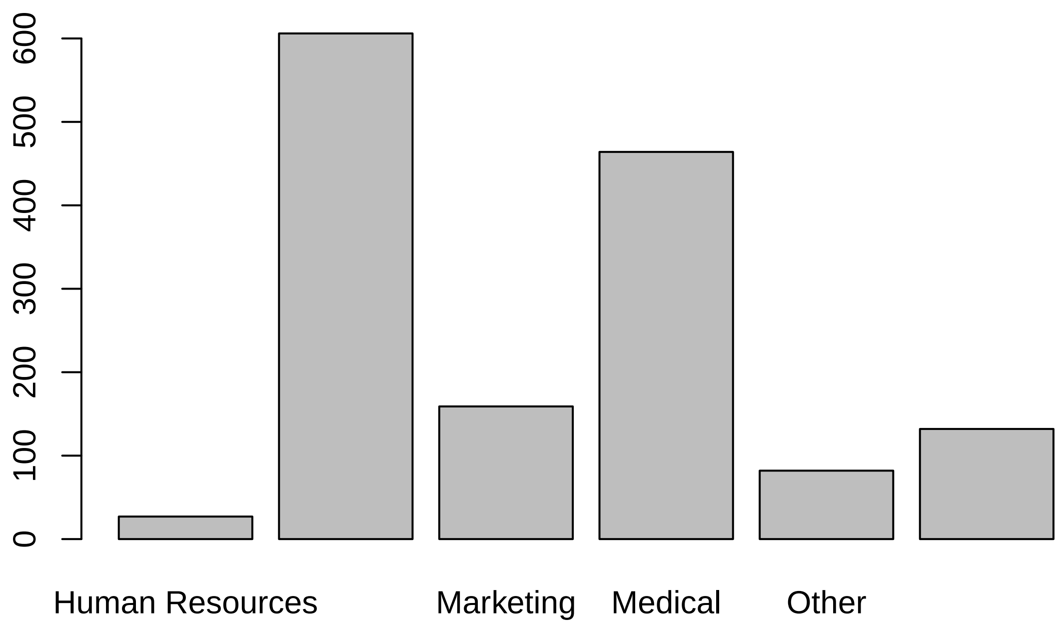

```
hist(data$DistanceFromHome)
```



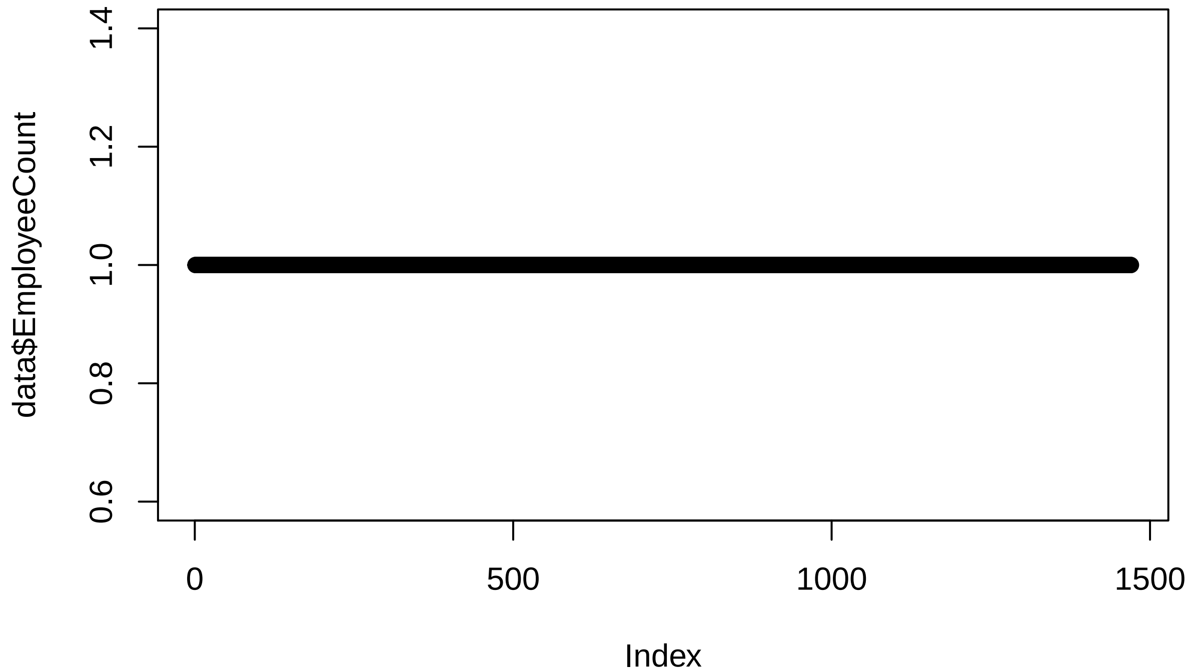

```
hist(data$Education)
```



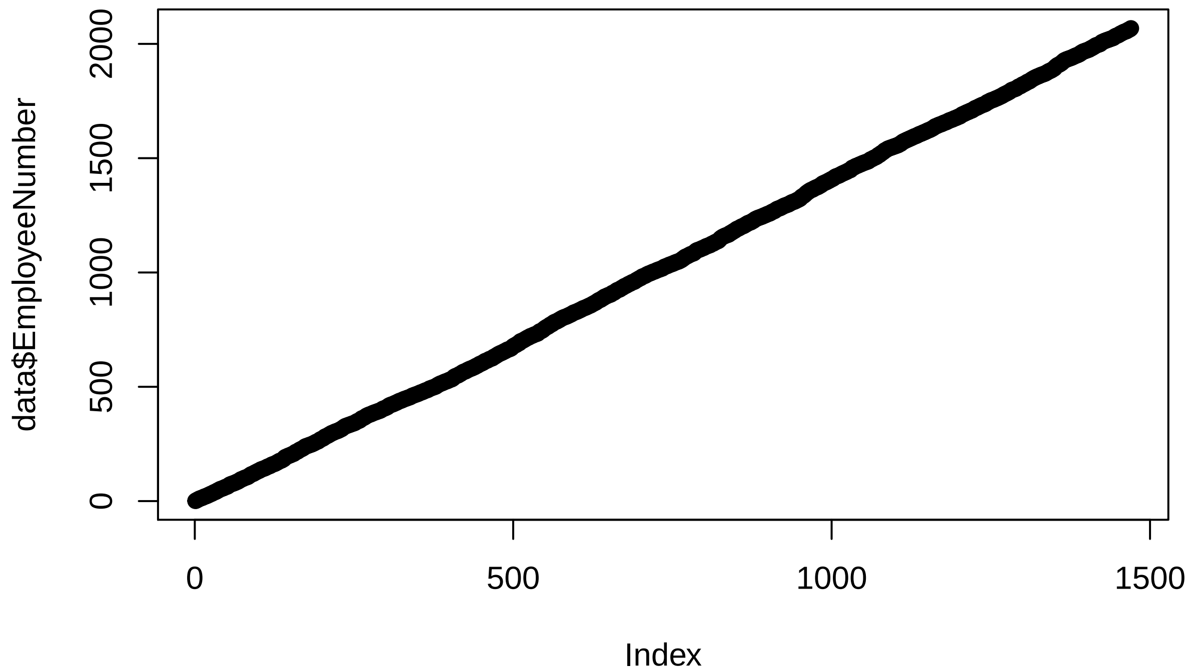

```
plot(data$EducationField)
```



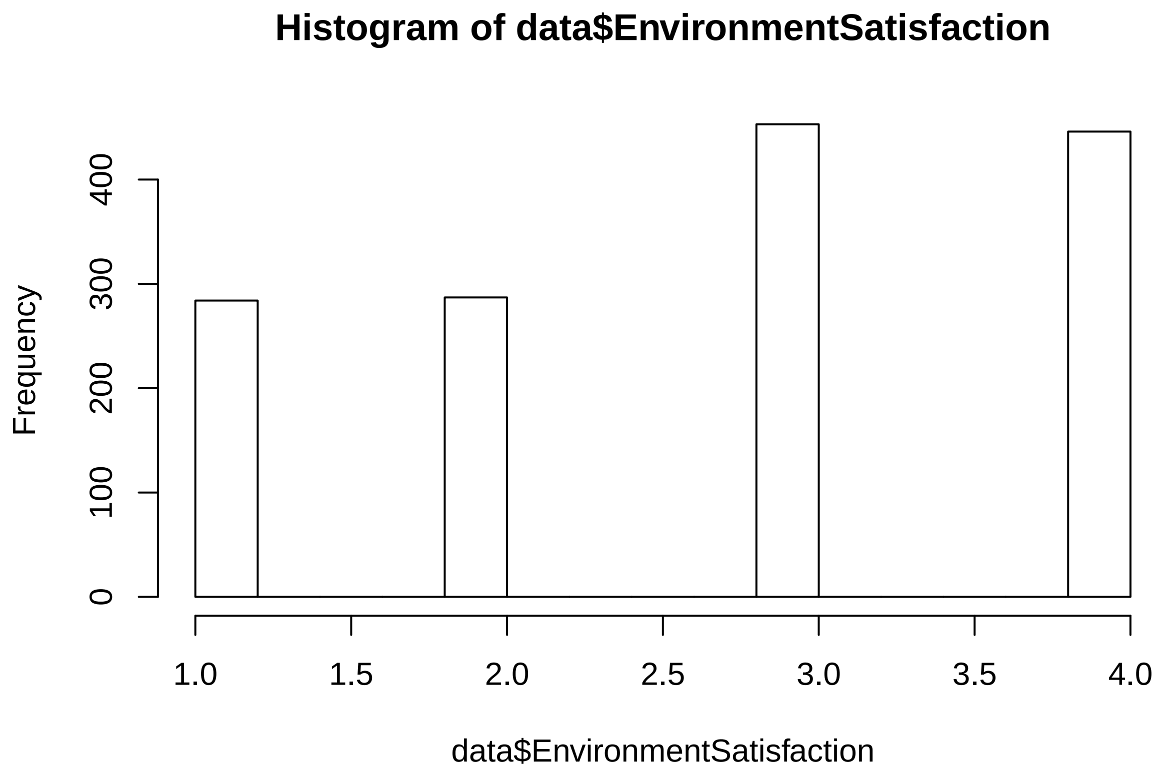

```r
plot(data$EmployeeCount)
```



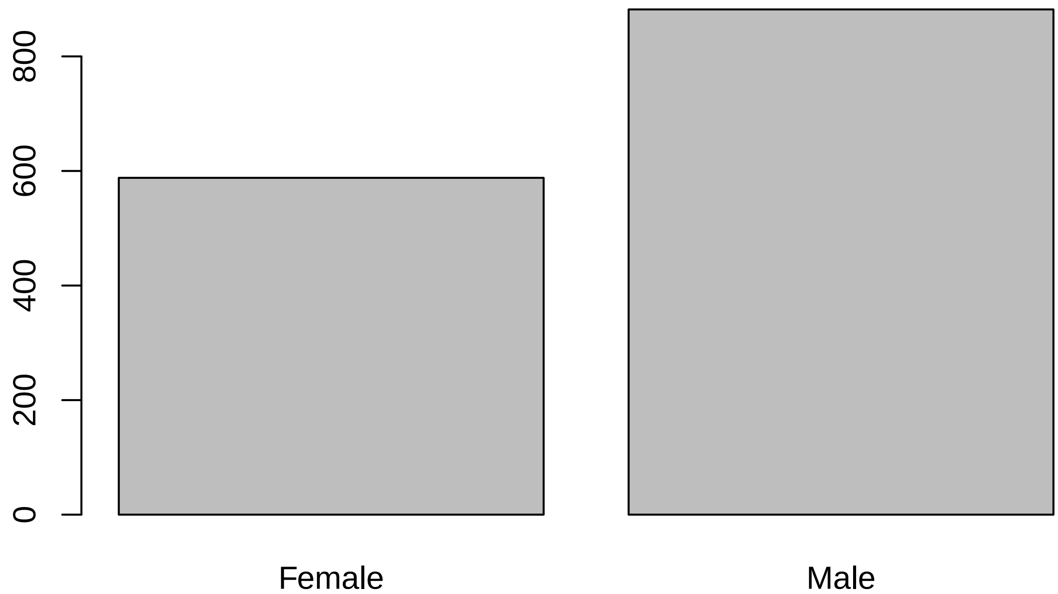

```r
plot(data$EmployeeNumber)
```



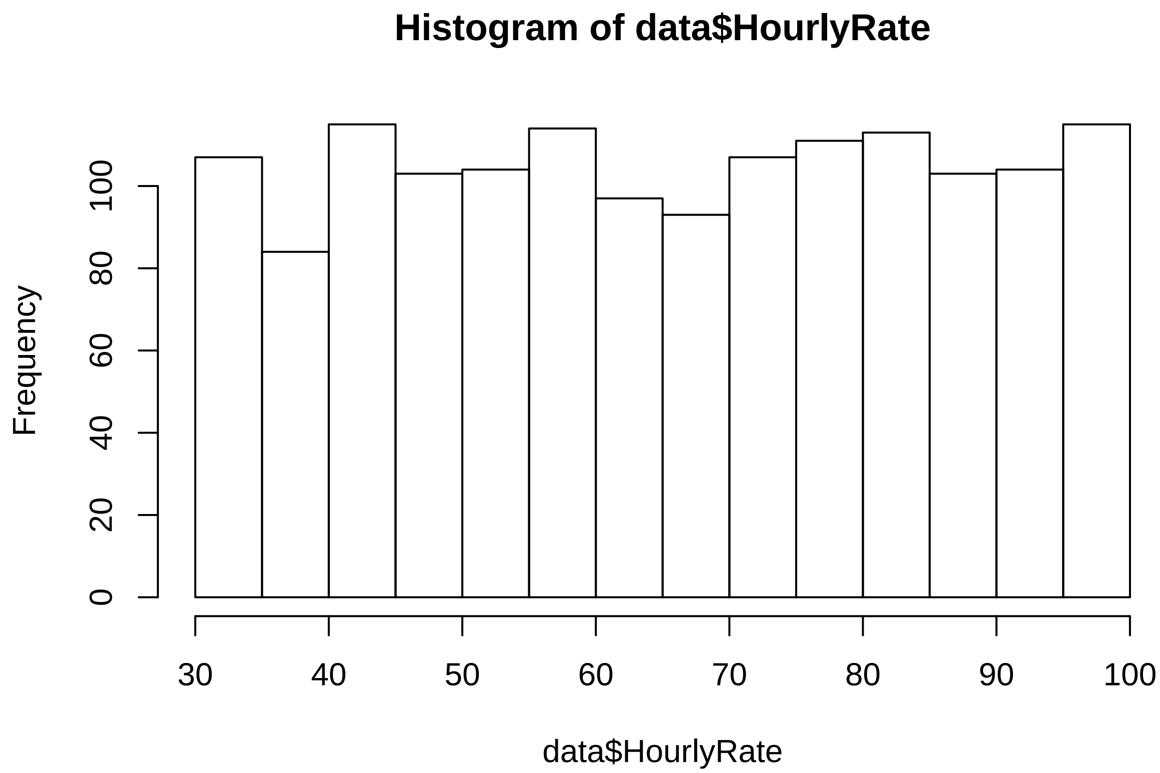

```r
hist(data$EnvironmentSatisfaction)
```



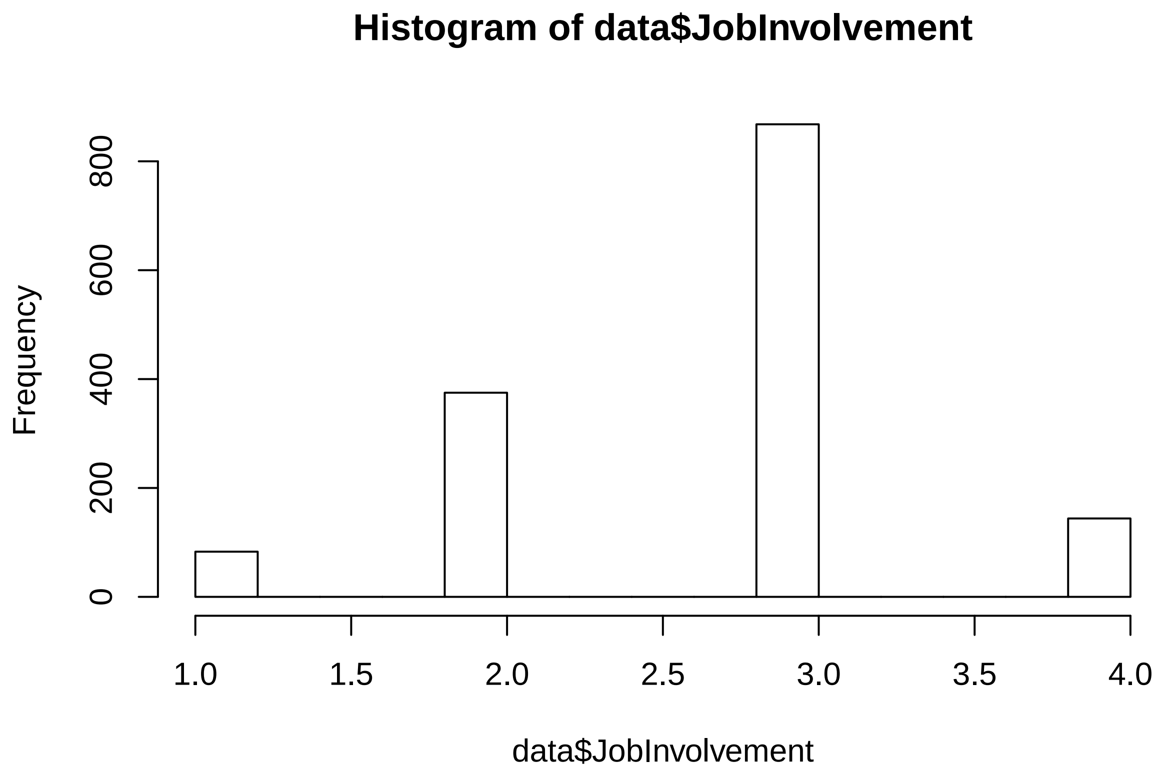

```r
plot(data$Gender)
```



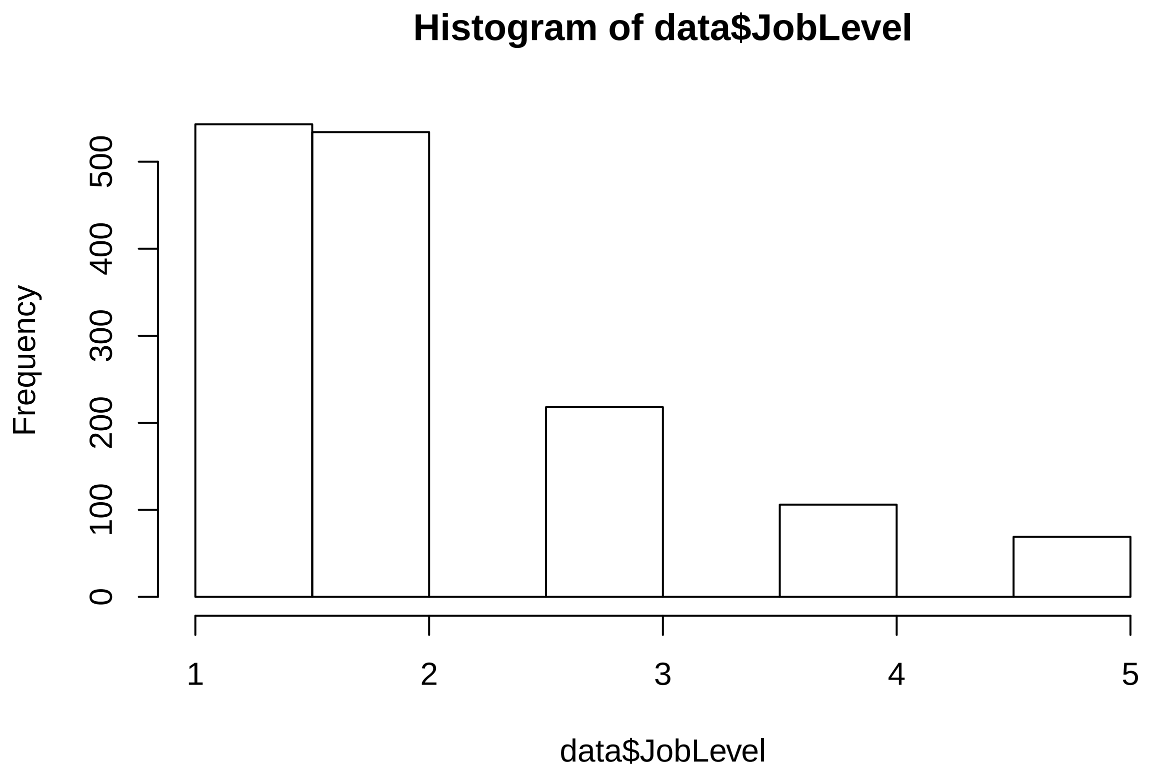

```r
hist(data$HourlyRate)
```



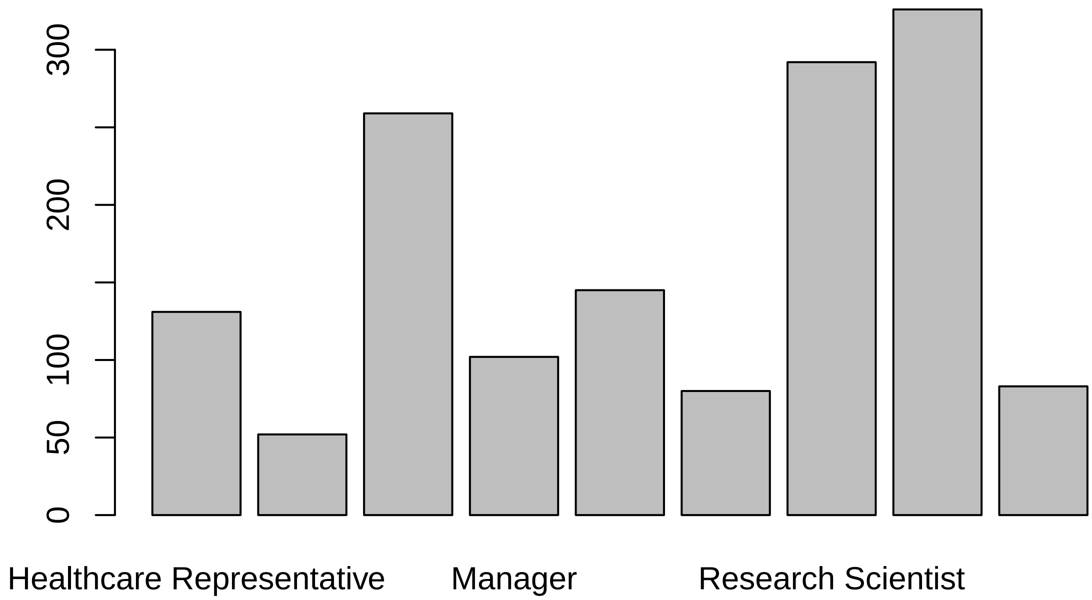

```
hist(data$JobInvolvement)
```



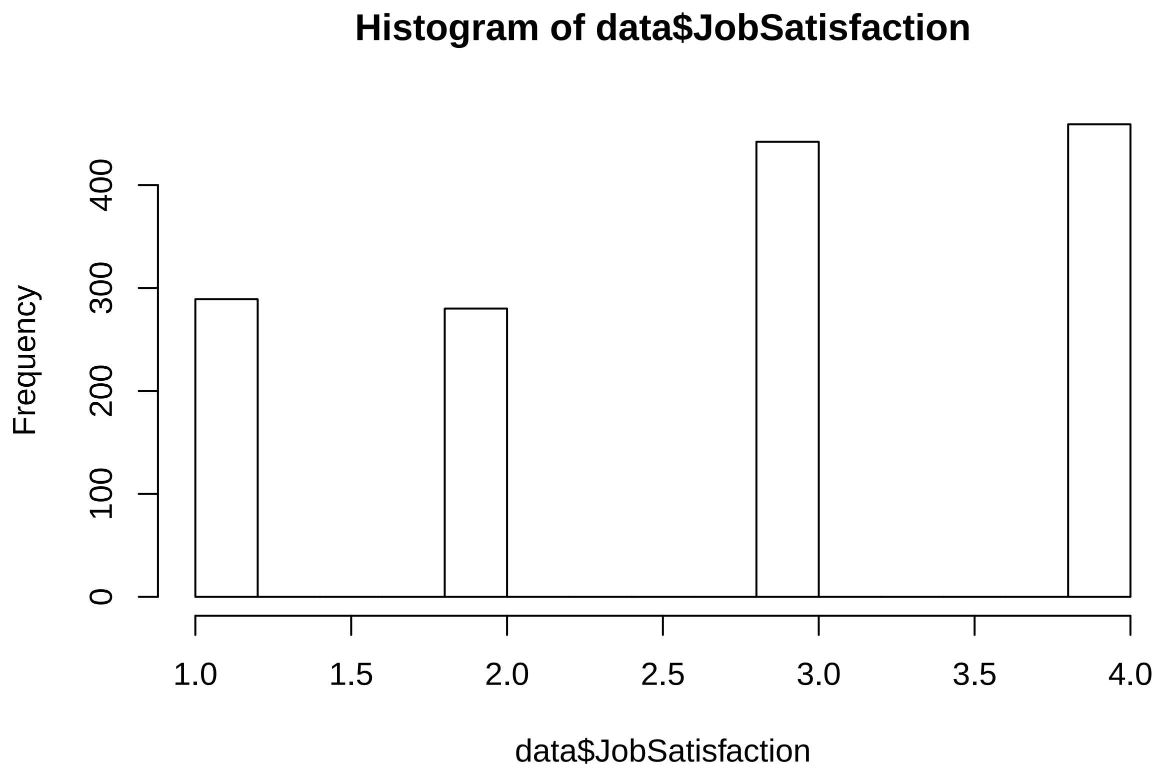

```
hist(data$JobLevel)
```



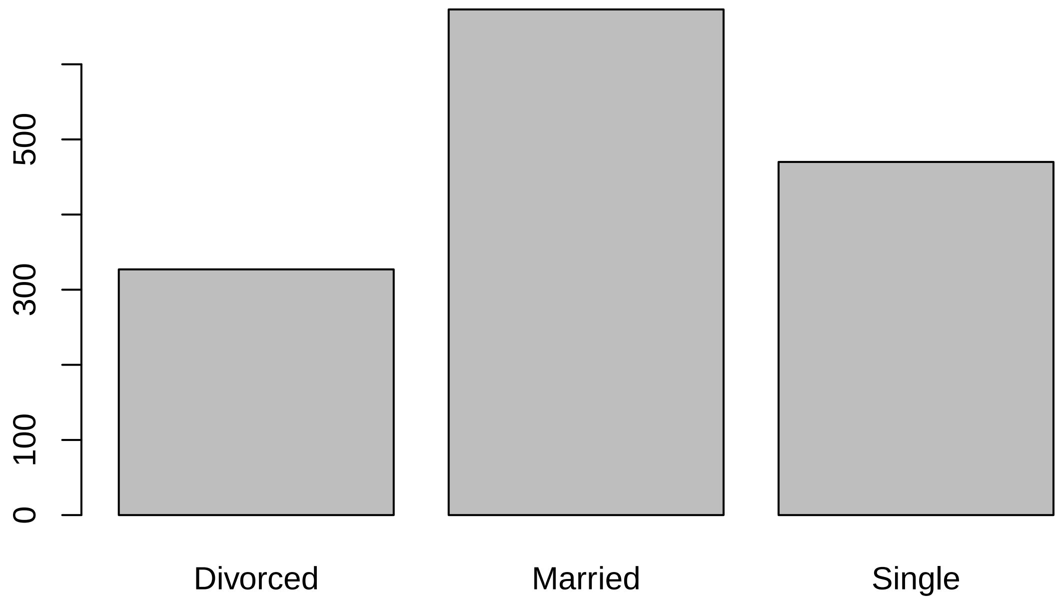

```
plot(data$JobRole)
```



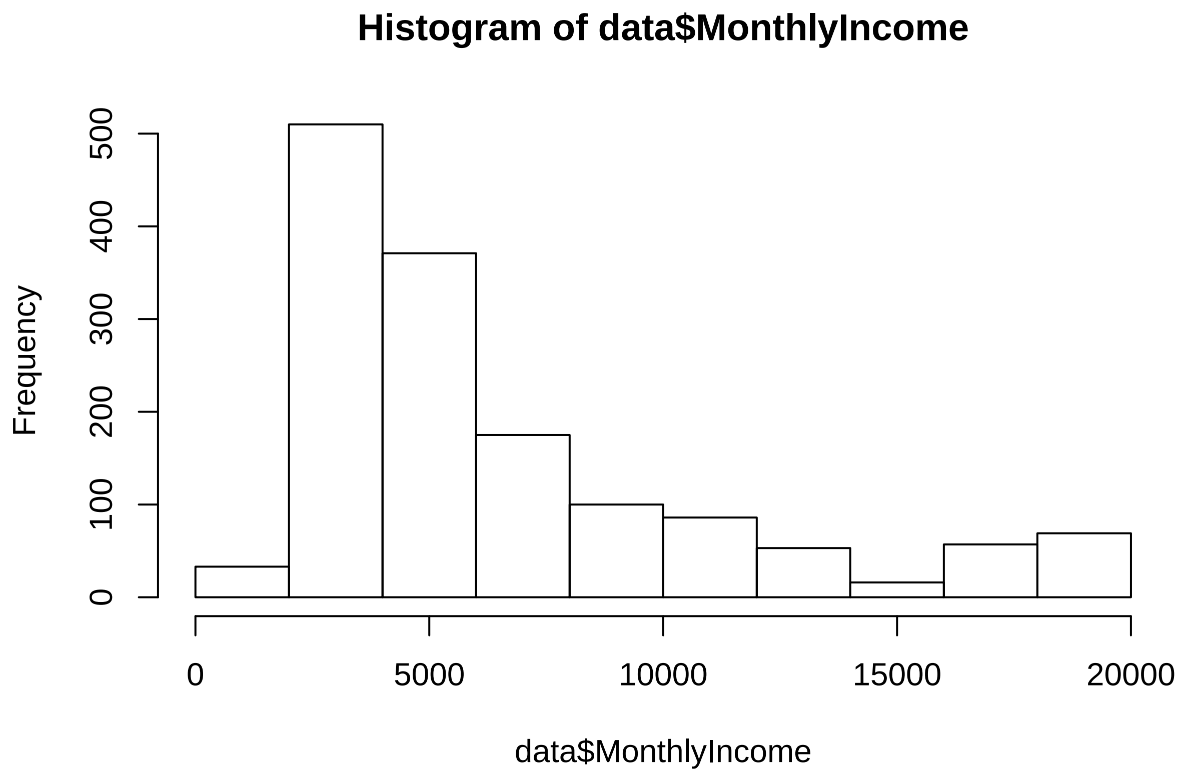

```
hist(data$JobSatisfaction)
```



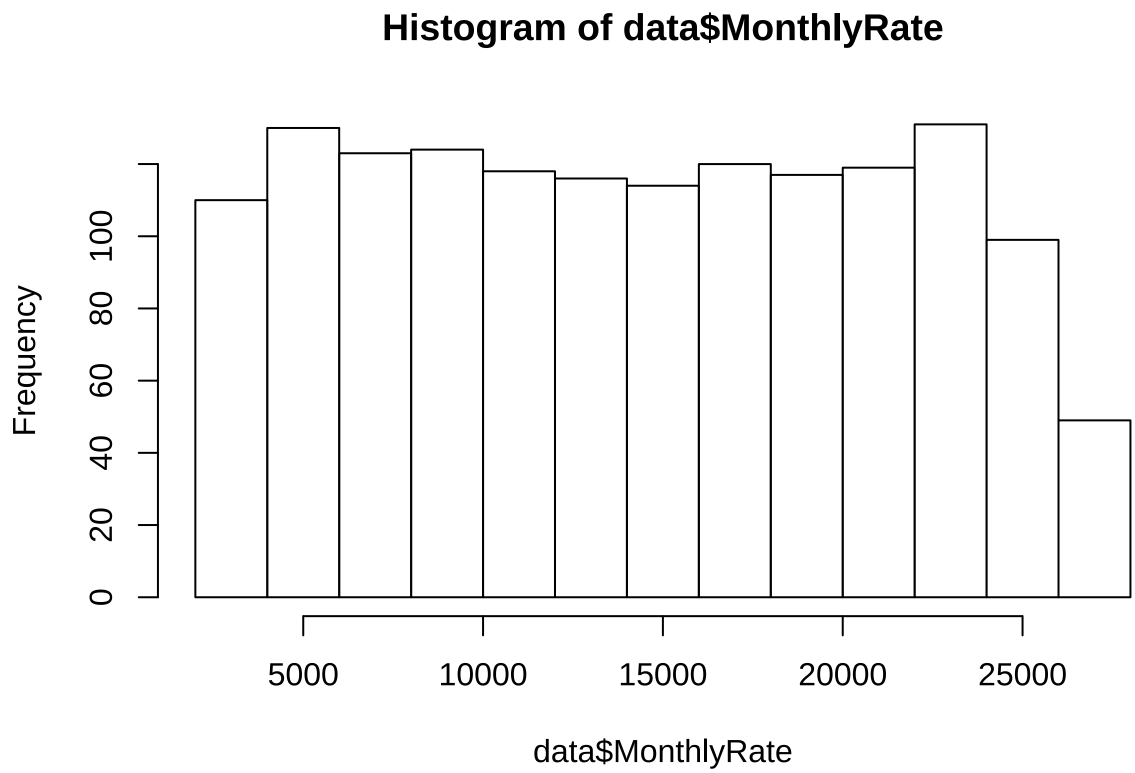

```r
plot(data$MaritalStatus)
```



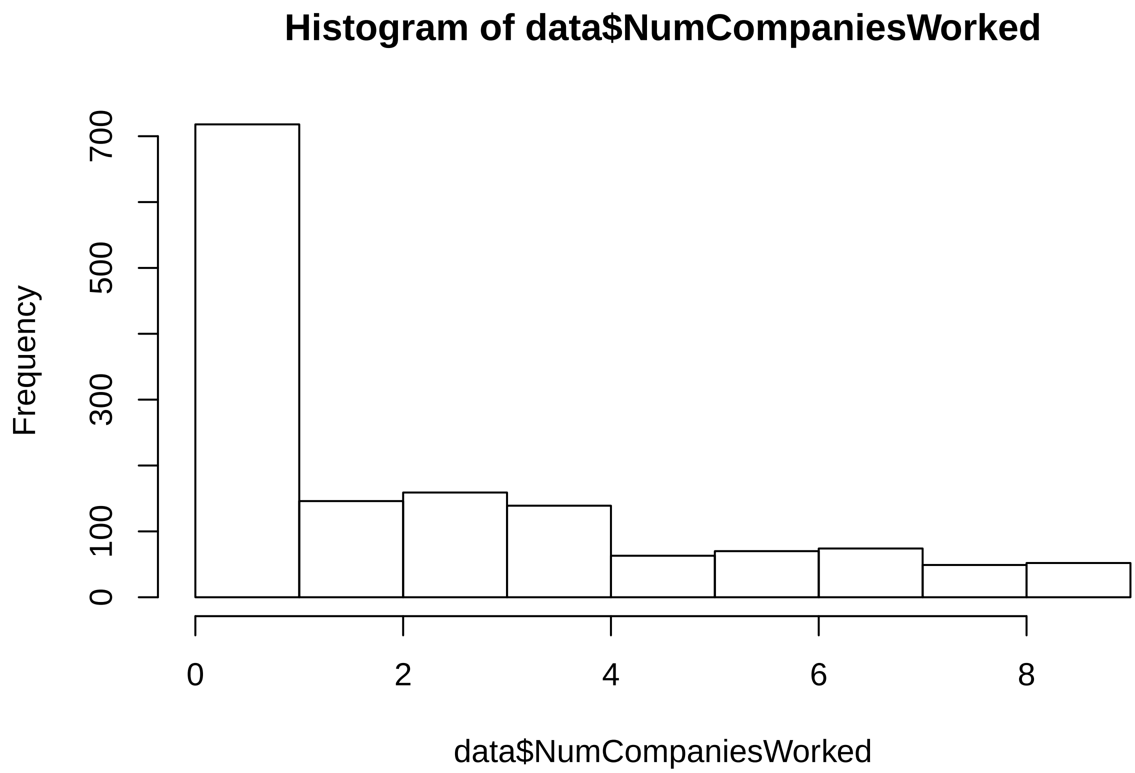

```r
hist(data$MonthlyIncome)
```



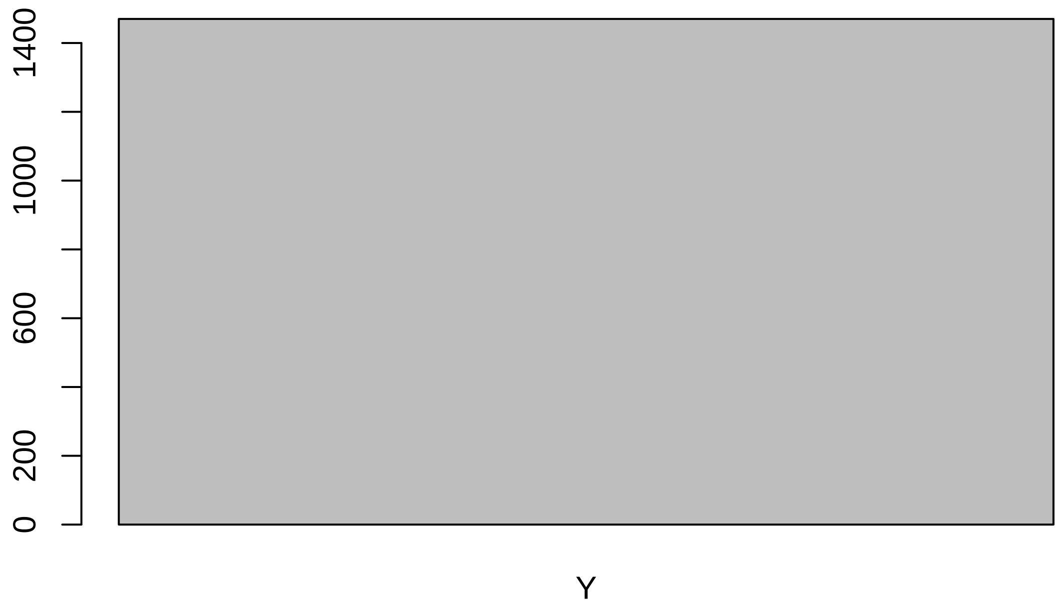

```
hist(data$MonthlyRate)
```



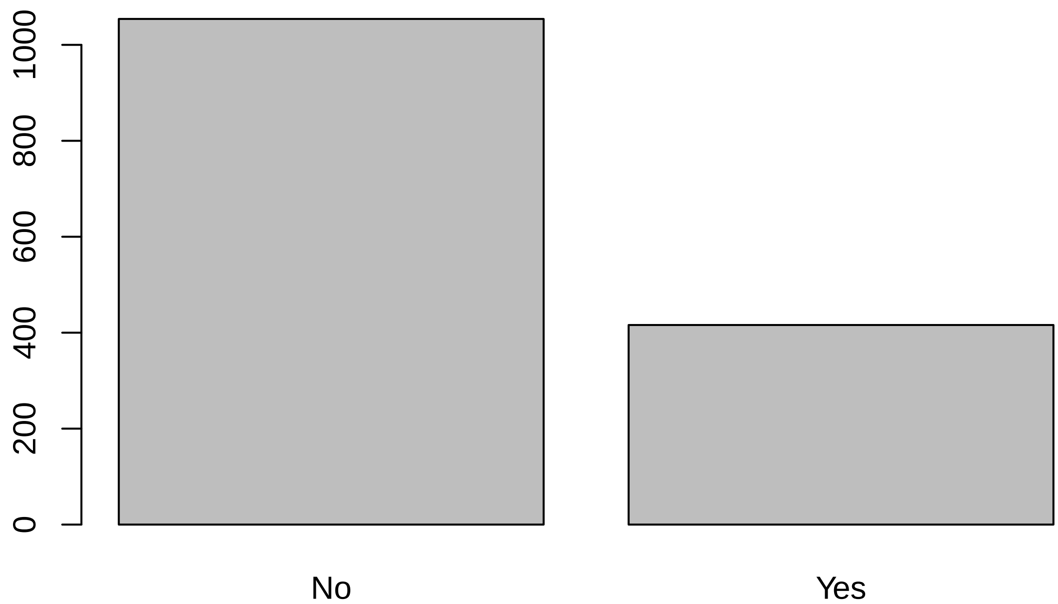

```
hist(data$NumCompaniesWorked)
```



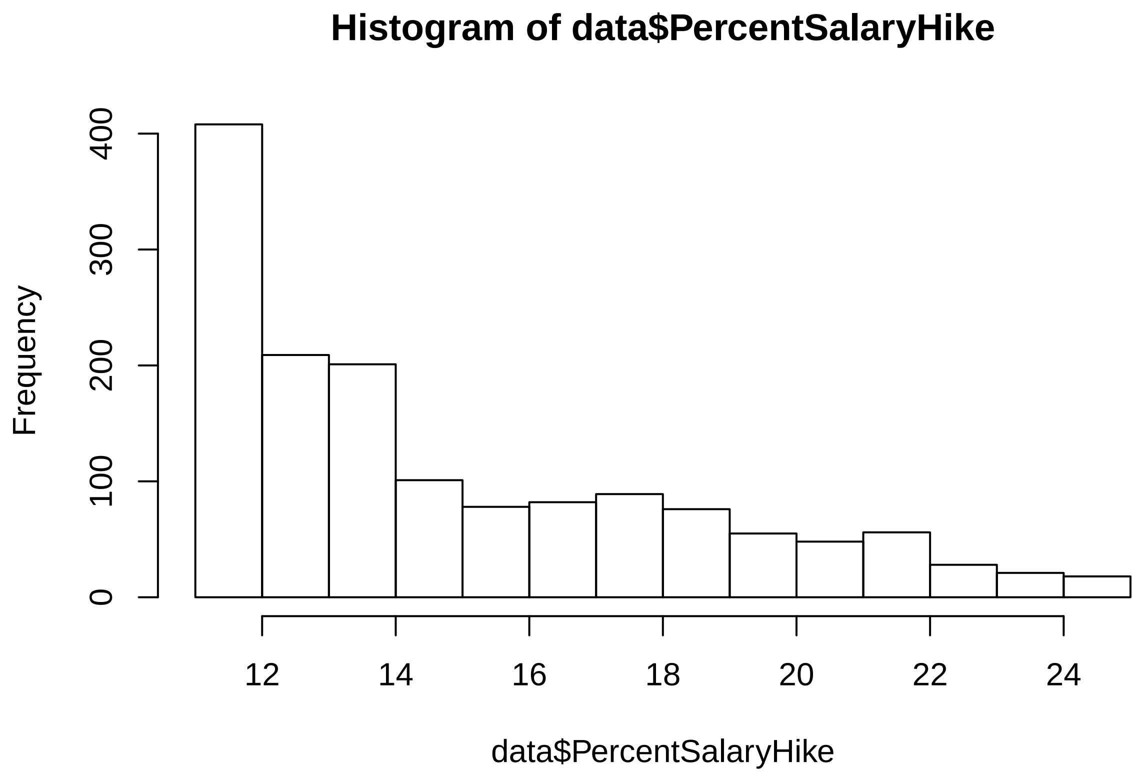

```
plot(data$Over18)
```



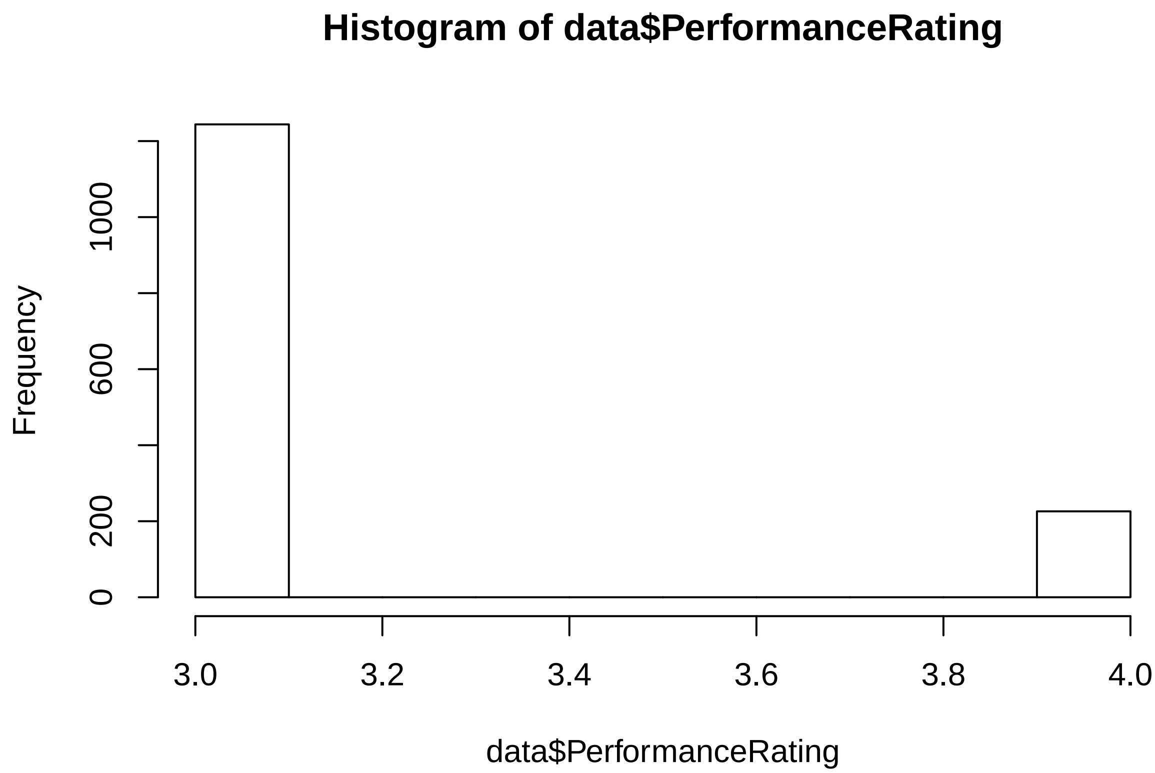

```r
plot(data$OverTime)
```



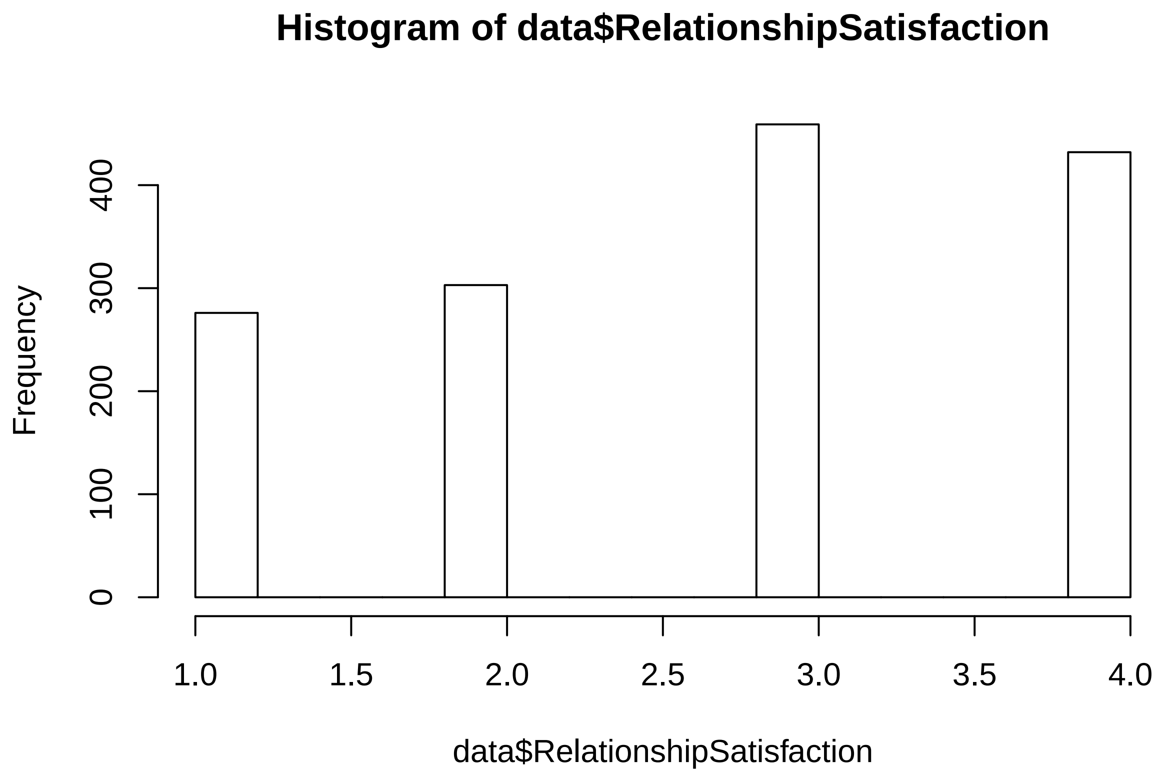

```r
hist(data$PercentSalaryHike)
```



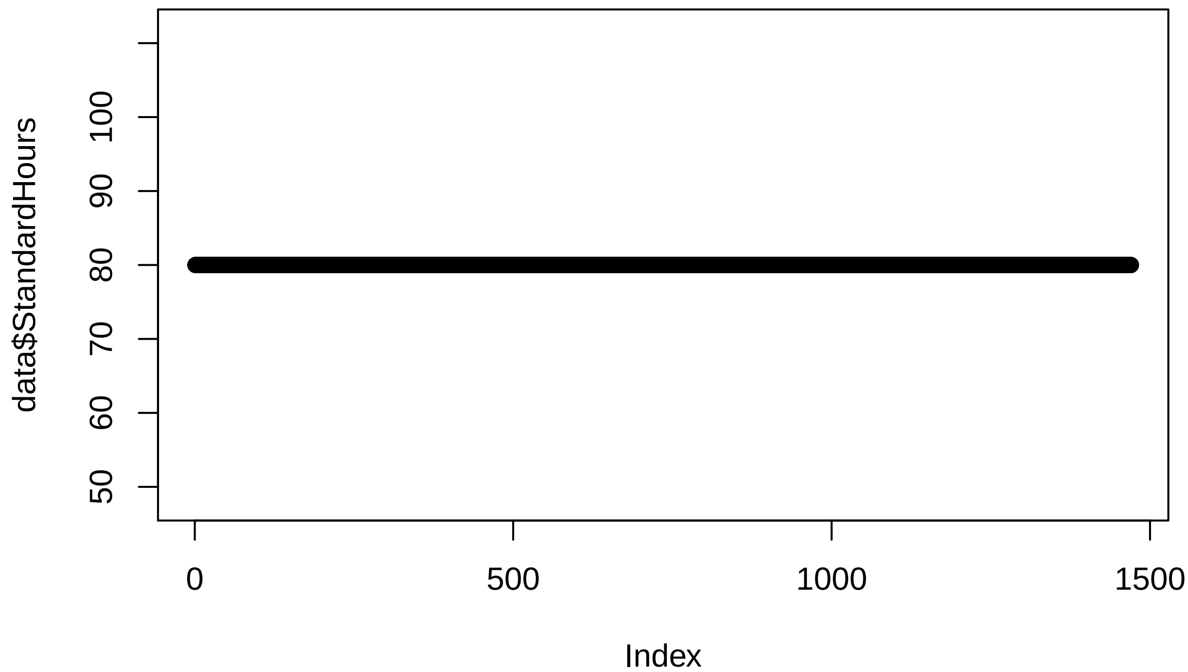

```r
hist(data$PerformanceRating)
```



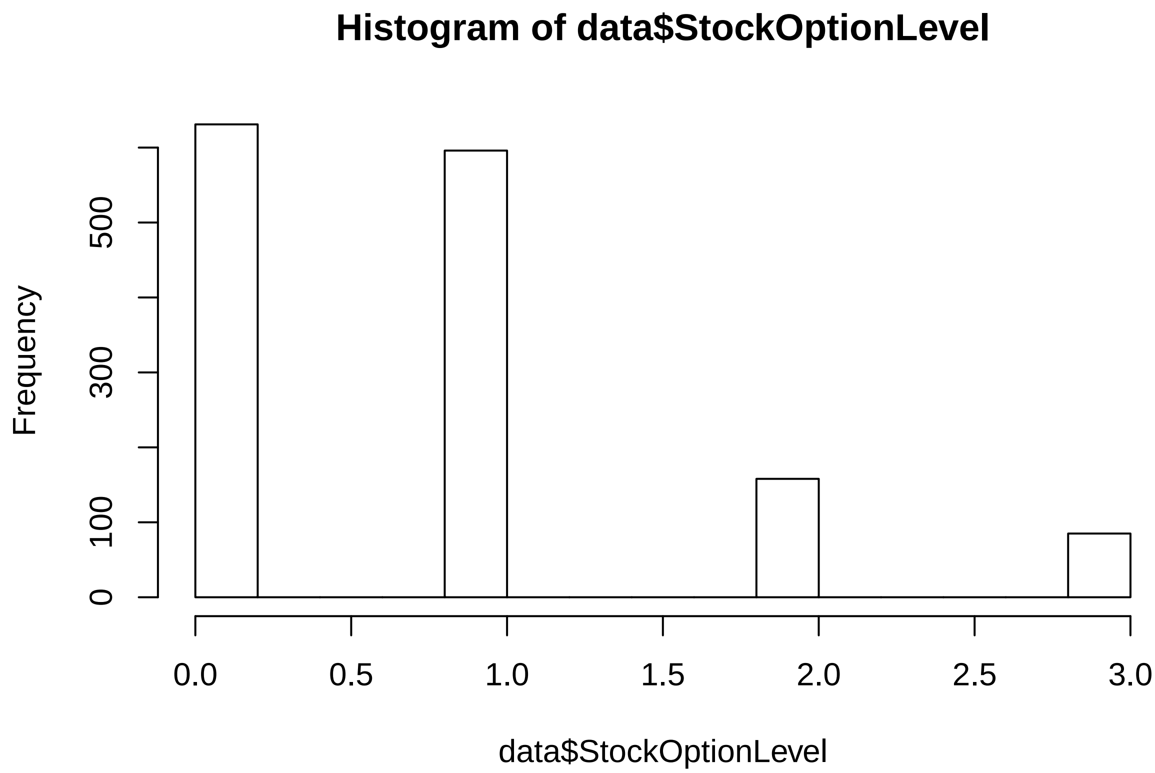

```
hist(data$RelationshipSatisfaction)
```



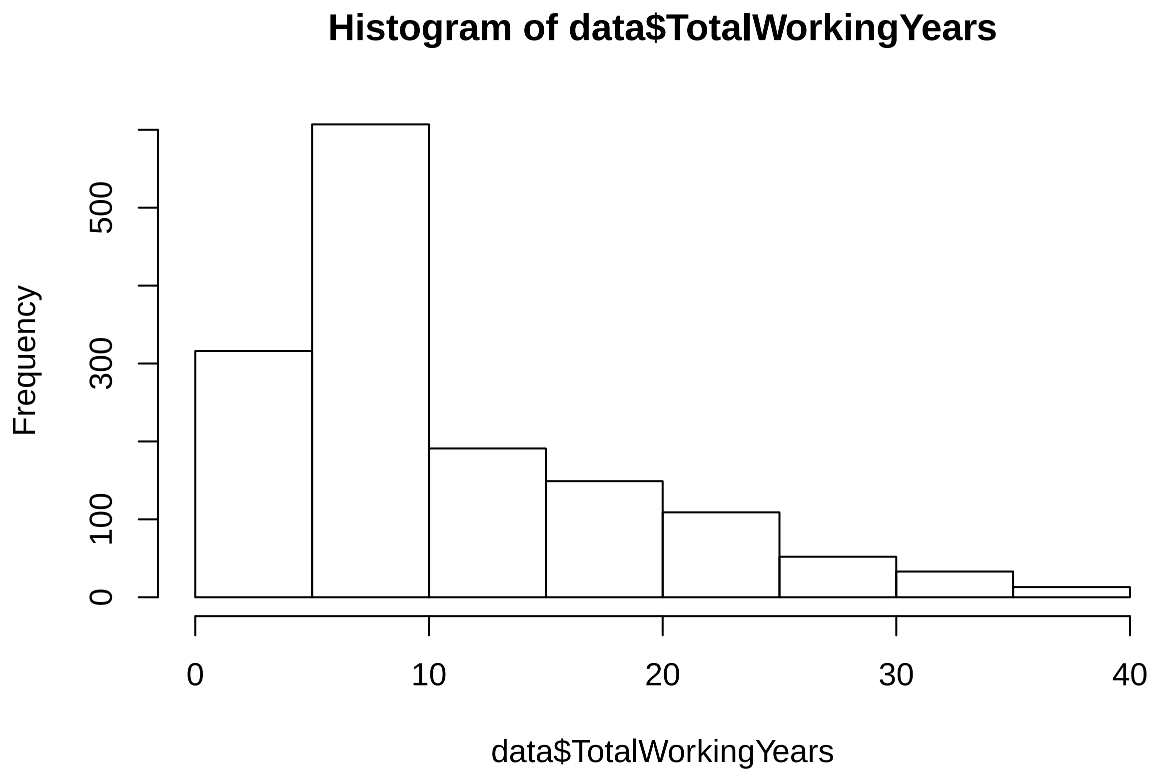

```r
plot(data$StandardHours)
```



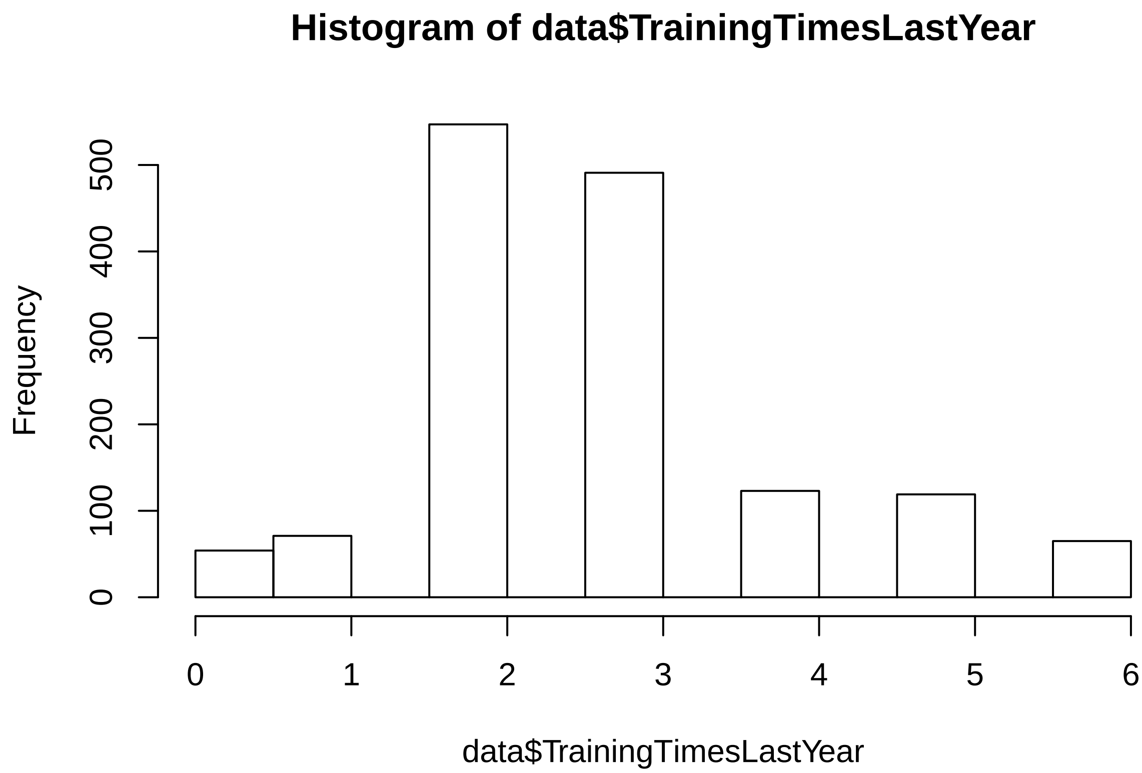

```r
hist(data$StockOptionLevel)
```



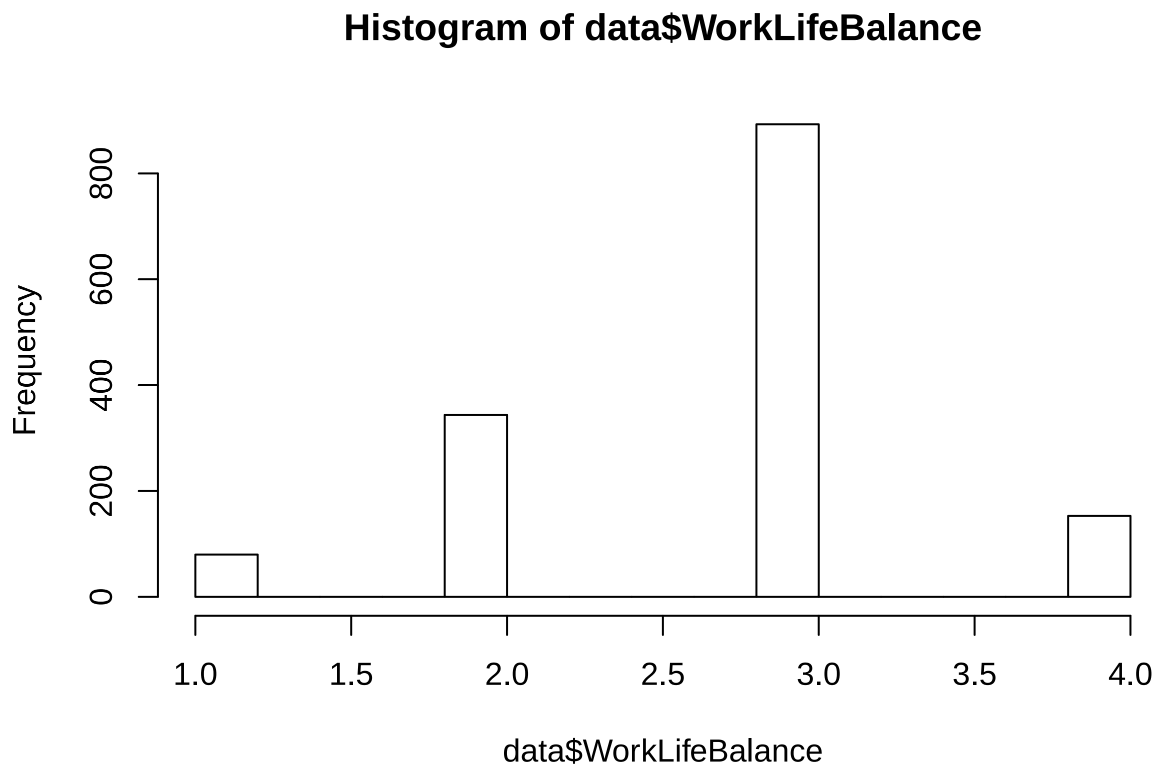

```
hist(data$TotalWorkingYears)
```



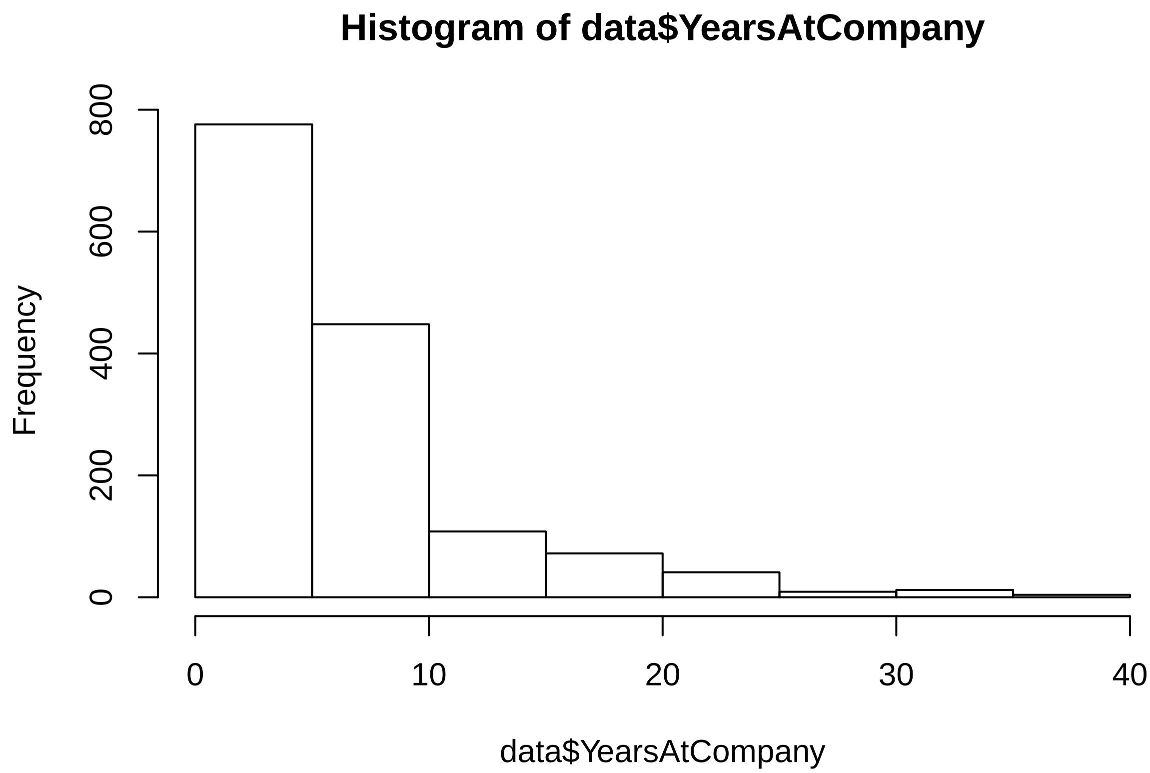

```r
hist(data$TrainingTimesLastYear)
```



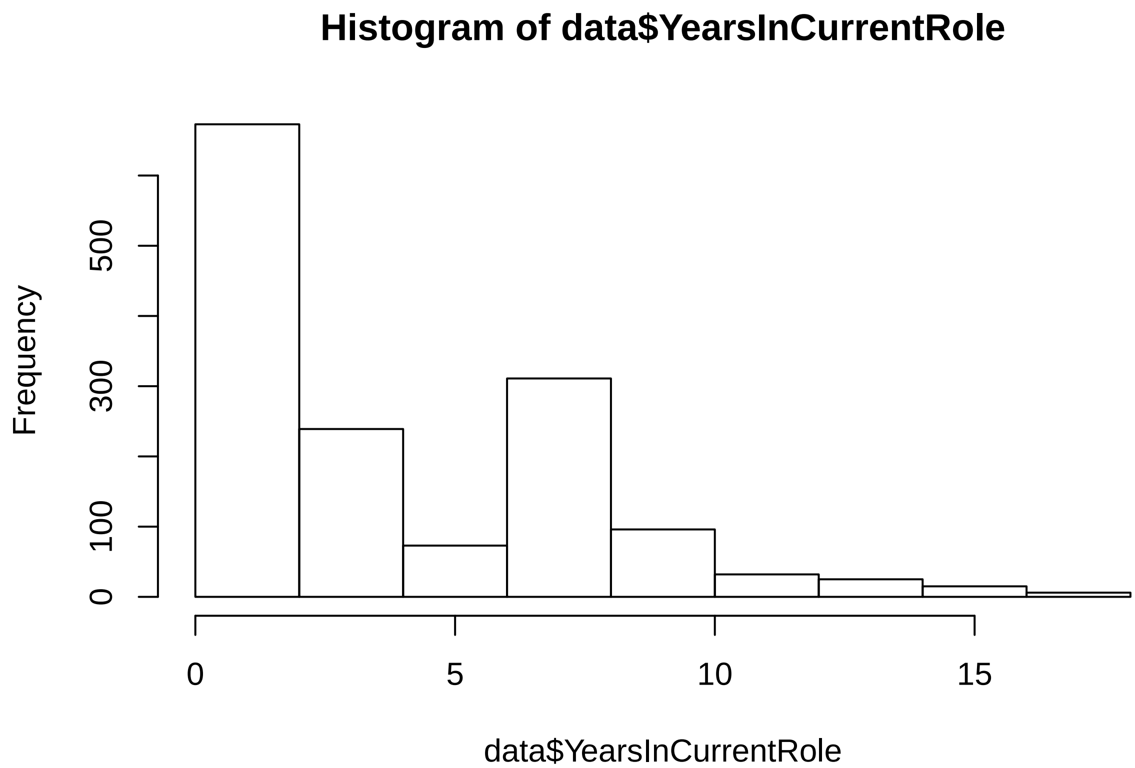

```r
hist(data$WorkLifeBalance)
```



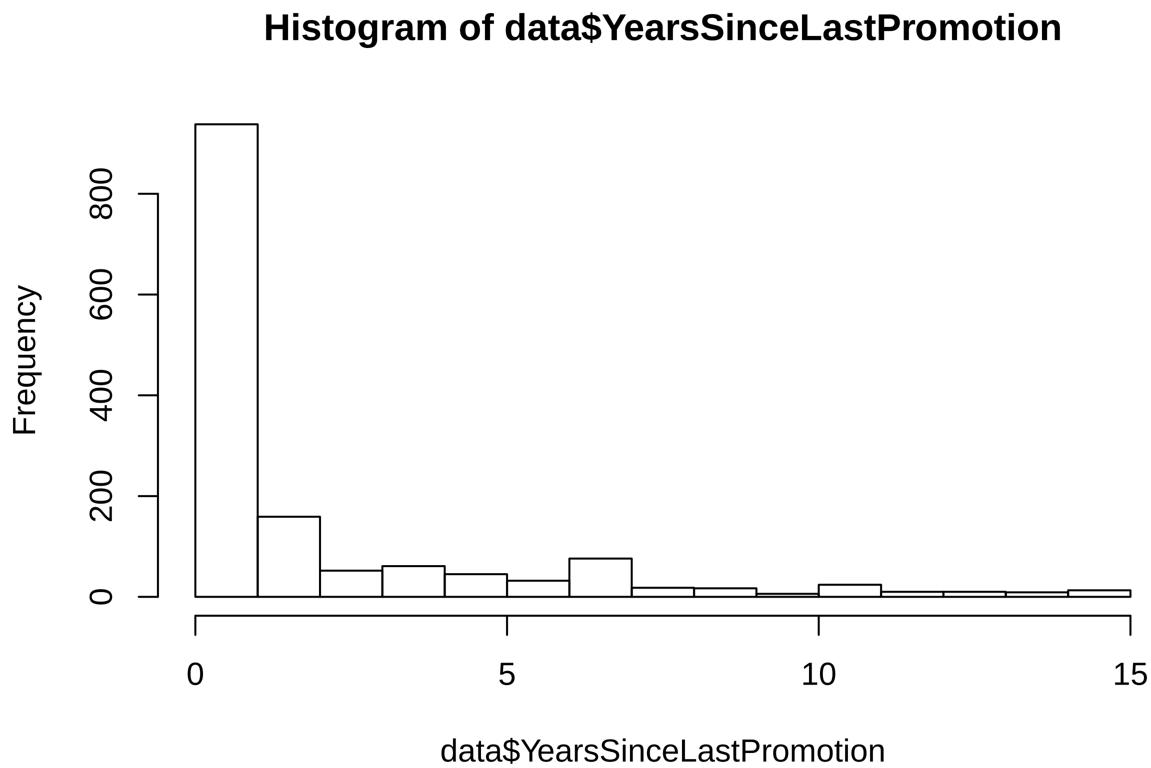

```
hist(data$YearsAtCompany)
```



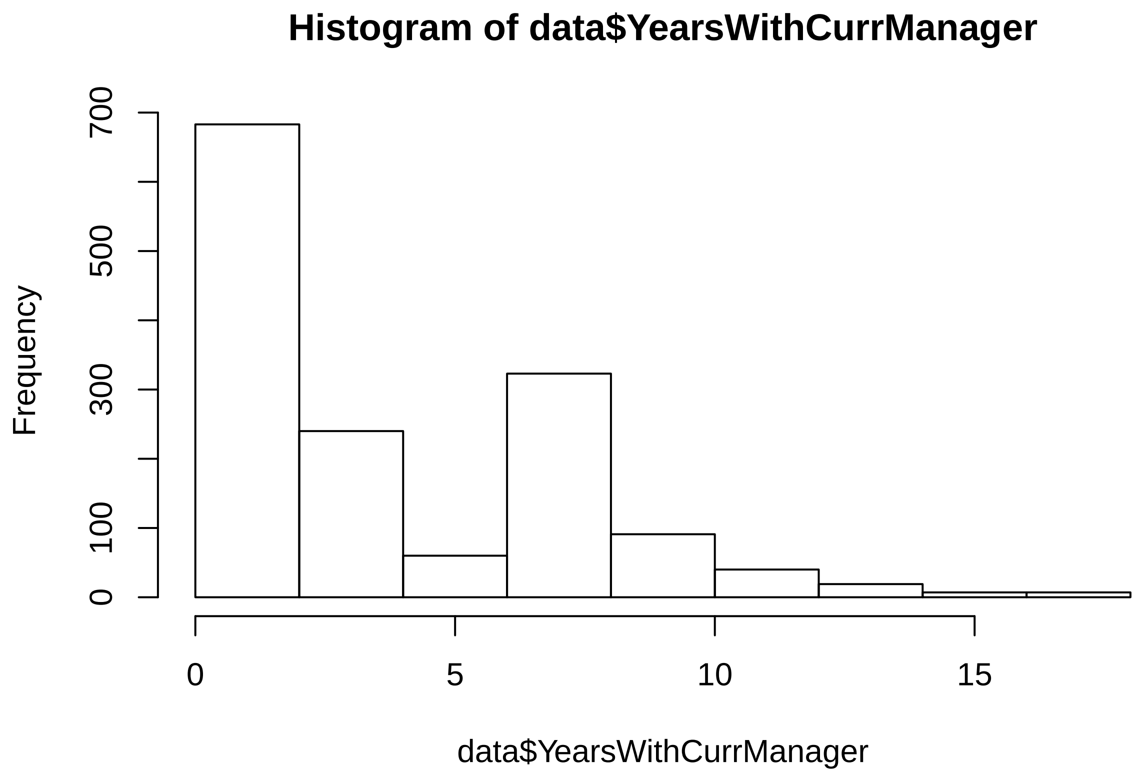

```
hist(data$YearsInCurrentRole)
```




```r
hist(data$YearsSinceLastPromotion)
```




```r
hist(data$YearsWithCurrManager)
```



Results

Dependent variable Attrition - attrition rate is 16%

Independent variables Age - employee are young (mean age is 36 years)

Business Travel - 80% of employees travel rarely or not at all

Daily rate - distribution is flat

Department - 65% of employees work in R&D, 30% in Sales

Distance from home - most employees work < 10km from home

Education - 65% of employees have bachelor degree or better

Education Field - 80% of employees have a technical degree

EmployeeCount - only one value, will be removed

EmployeeNumber- internal number, will be removed

Environment Satisfaction - most employees have a high environment satisfaction

Gender - 60% men, 40% women

Hourly Rate - distribution is flat

Job Involvement - most employees have a high job involvement

Job level - the job level for most employees is fairly low

Job Role - 17% management, 22% sales, technical 60%

Job Satisfaction - most employees have a high job satisfaction

Maritial status - 22% divorced, 46% married, 31% single

Monthly Income - distribution is skewed, most employees have a low monthly income (median = 4900 vs mean = 6500)

Monthly rate - distribution is flat

Num Companies Worked - most employees have only worked for a few companies (median = 2)

Over18 - only one value, will be removed

Overtime - 30% of employees get overtime, 70% do not get overtime

Percent salary Hike - most employees got the average salary

Performance Rating - distribution is skewed, there are only 3 and 4 ratings

Relationship Satisfaction - most employees are satisfied with relationship

StandardHours - only one value, will be removed

Stock Options Levels - 80% of employees get zero or few stock options

Total Working Years - most employees have about 10 years work experience

Training Times - most employees were trained 2 - 3 times in the year

Work Life Balance - most employees are happy with work life balance

Years at Company - 80% of employees have been with the company less than 10 years

Years in Role - most employees have been in their role less than 5 years

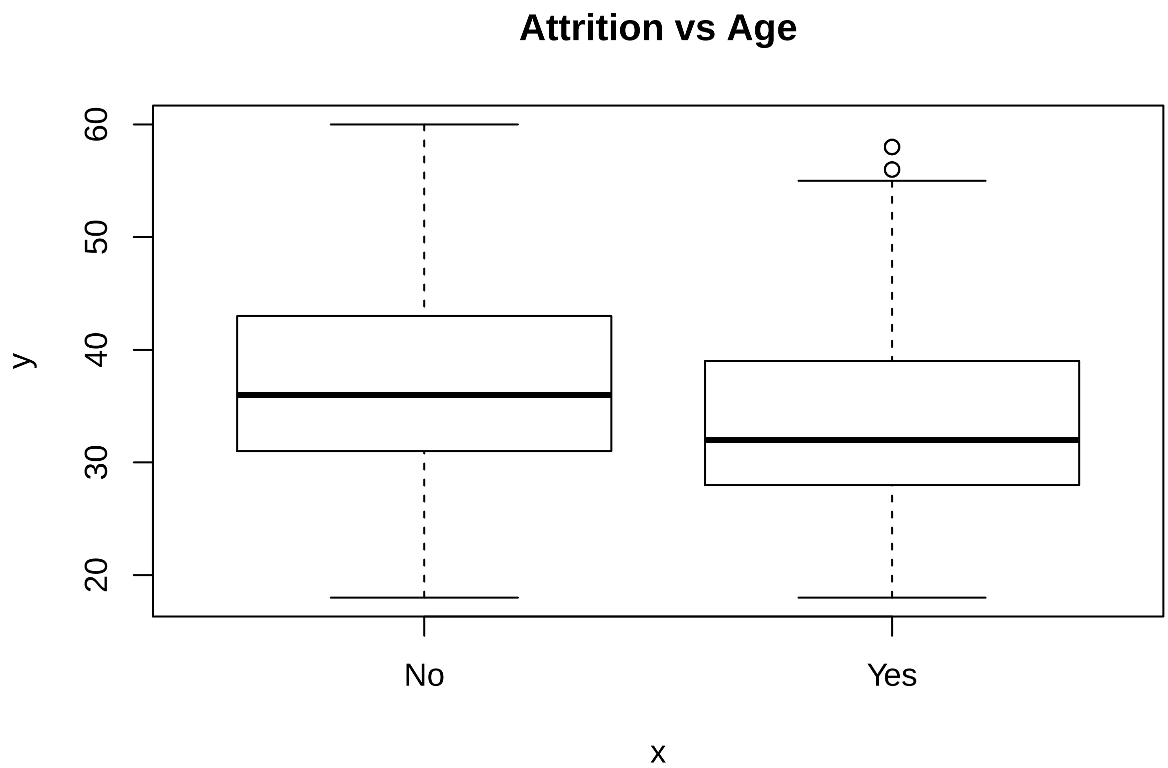
Years since promotion - most employees have been promoted within last years

Years with manager - most employees have been with their manager less than 5 years

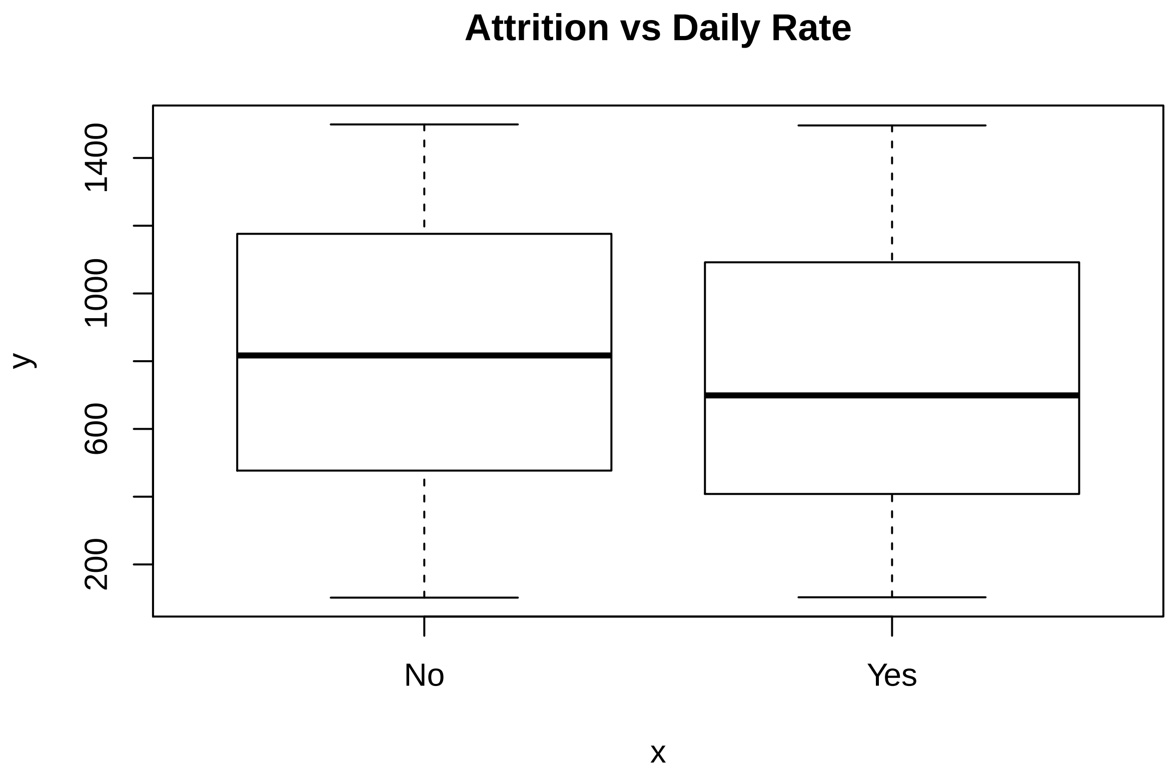
Analyze dependent vs independent variable

Numerical variable against dependent variable (Bar Plots)

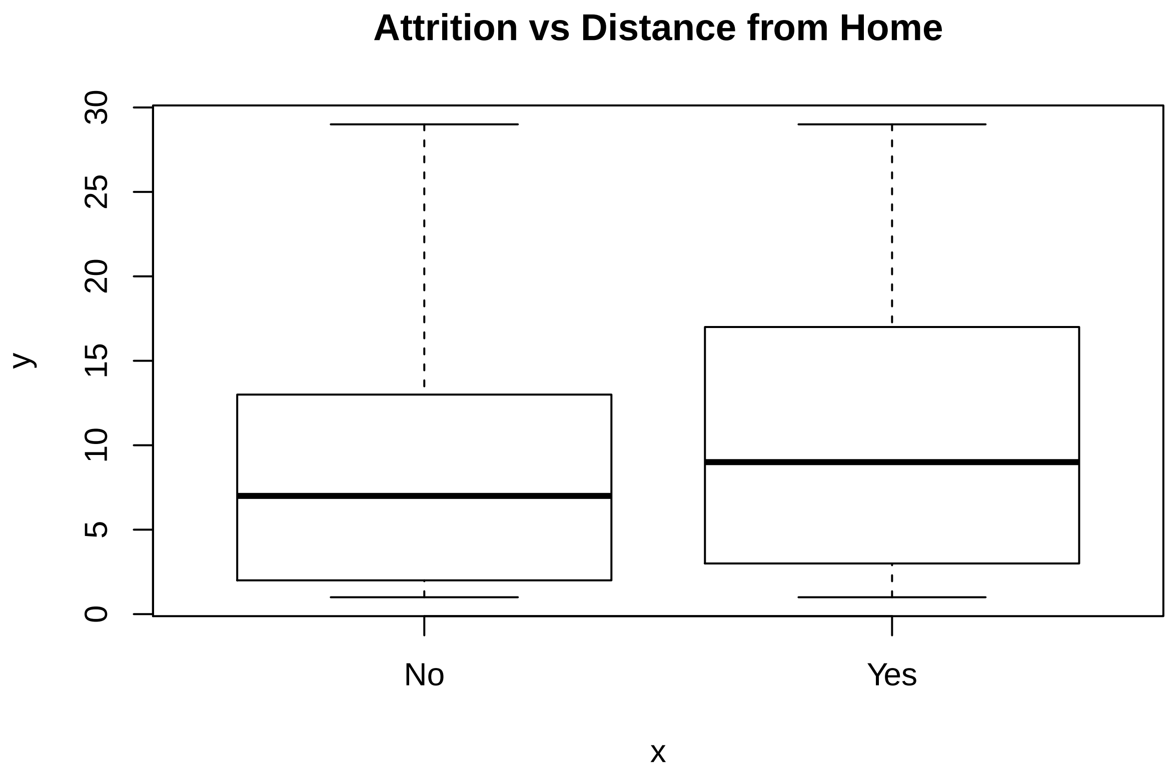
```
plot(x=data$Attrition, y=data$Age, main ="Attrition vs Age")
```




```
plot(x=data$Attrition, y=data$DailyRate, main ="Attrition vs Daily Rate")
```



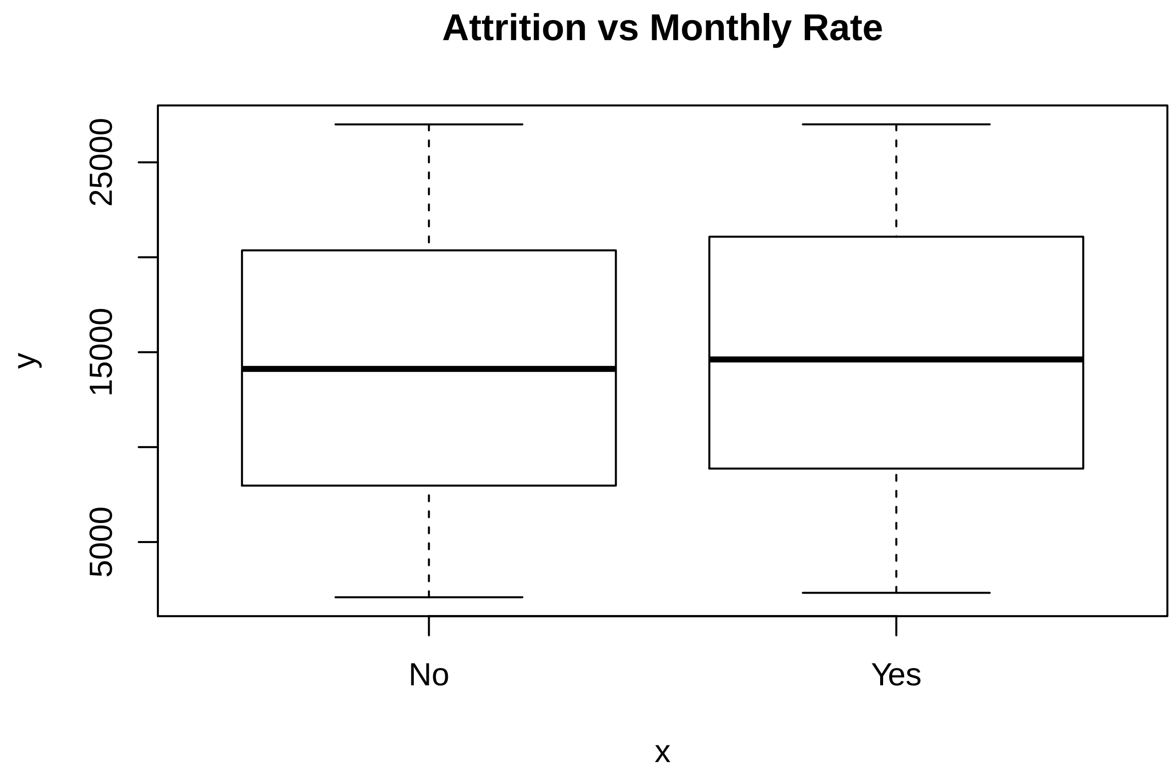

```r
plot(x=data$Attrition, y=data$DistanceFromHome, main ="Attrition vs Distance from Home")
```



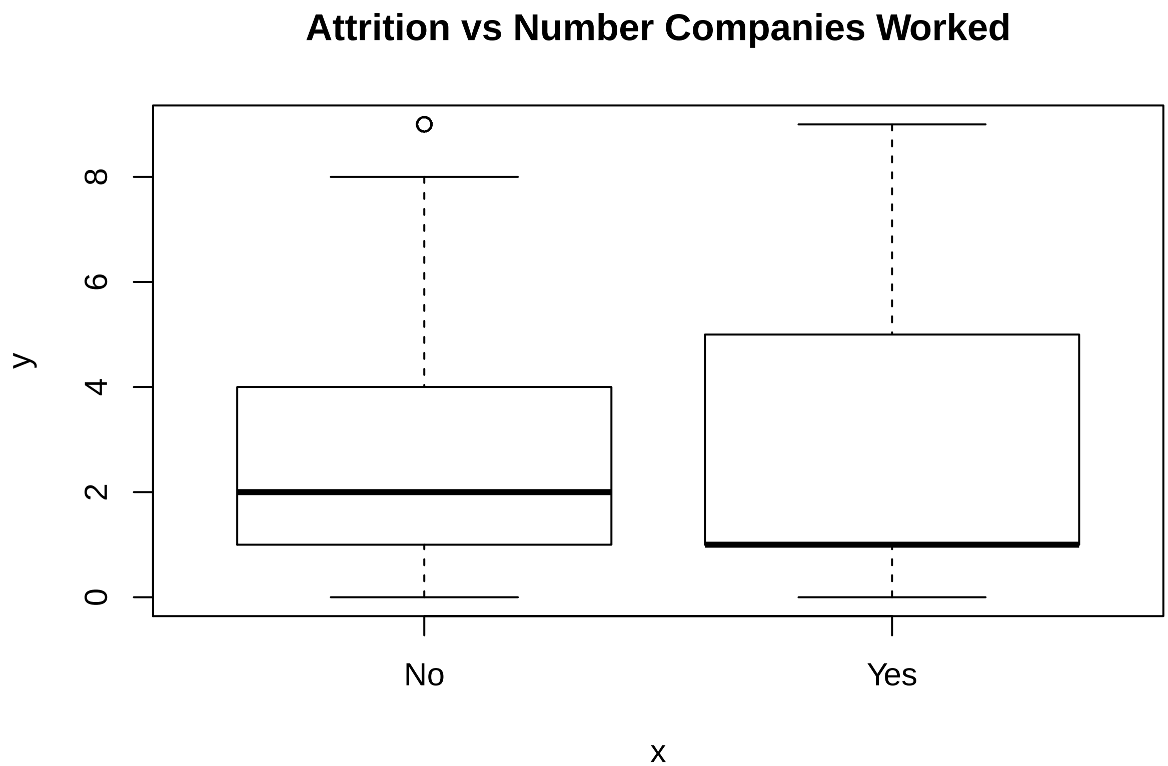

```r
plot(x=data$Attrition, y=data$HourlyRate, main = "Attrition vs Hourly Rate")
```


```r
plot(x=data$Attrition, y=data$MonthlyIncome, main="Attrition vs Monthly Income")
```

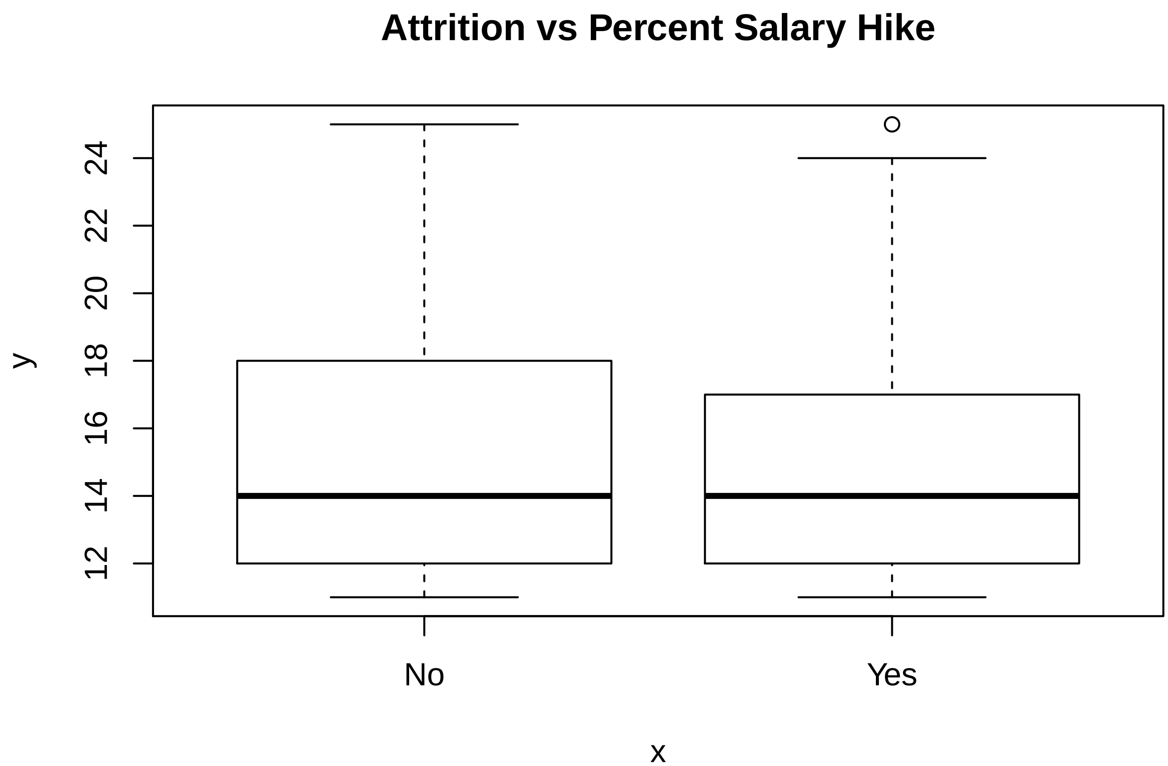

```r
plot(x=data$Attrition, y=data$MonthlyRate, main="Attrition vs Monthly Rate")
```




```r
plot(x=data$Attrition, y=data$NumCompaniesWorked, main="Attrition vs Number Companies Worked")
```

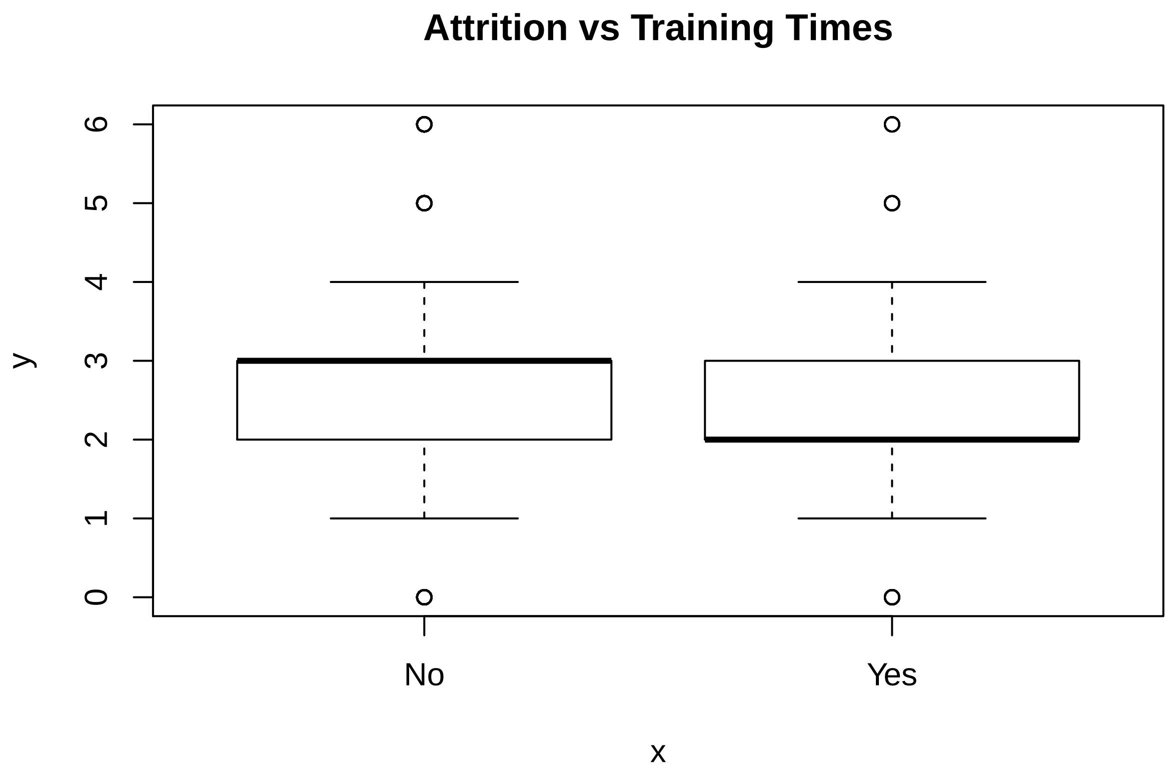



```r
plot(x=data$Attrition, y=data$PercentSalaryHike, main="Attrition vs Percent Salary Hike")
```

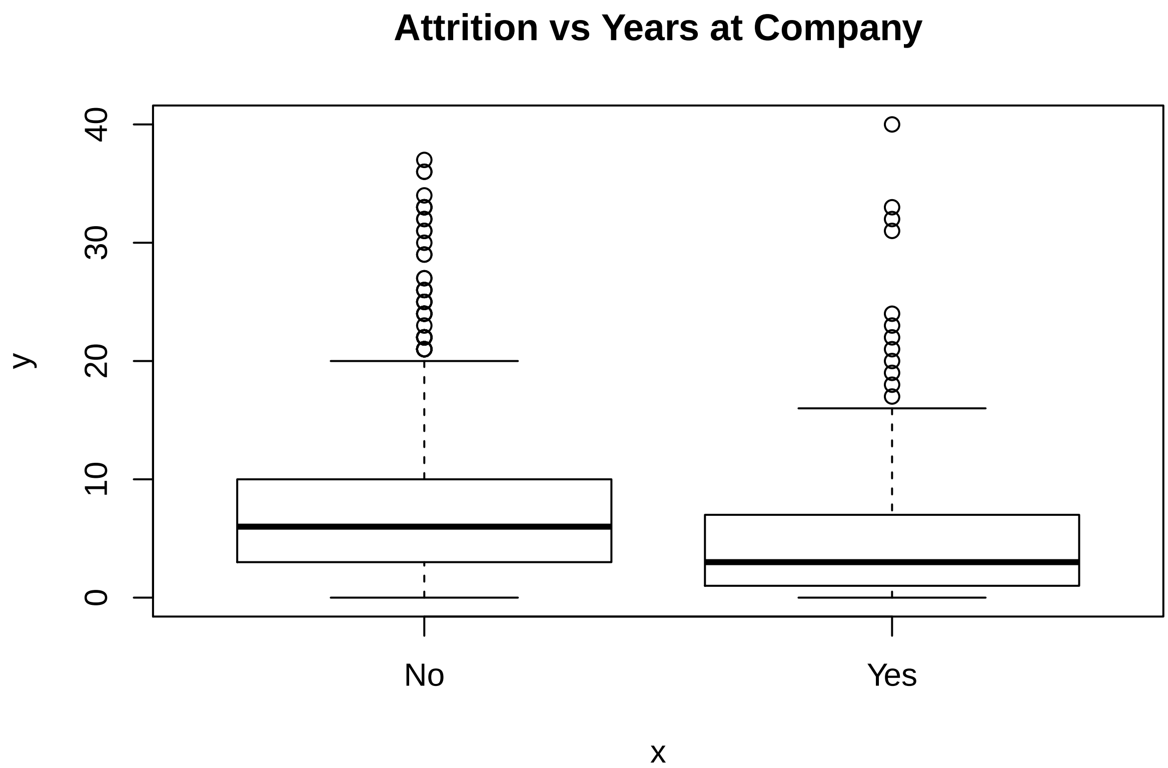



```r
plot(x=data$Attrition, y=data$TotalWorkingYears, main="Attrition vs Total Working Years")
```

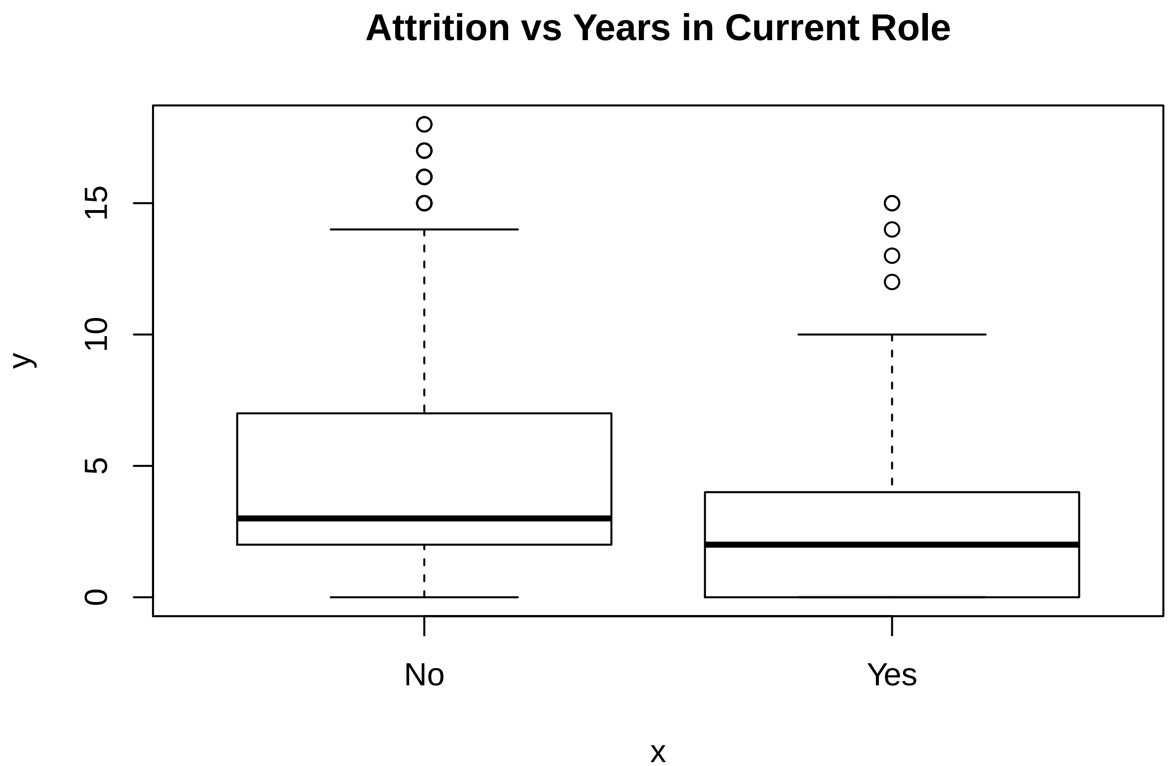

```
plot(x=data$Attrition, y=data$TrainingTimesLastYear, main="Attrition vs Training Times")
```



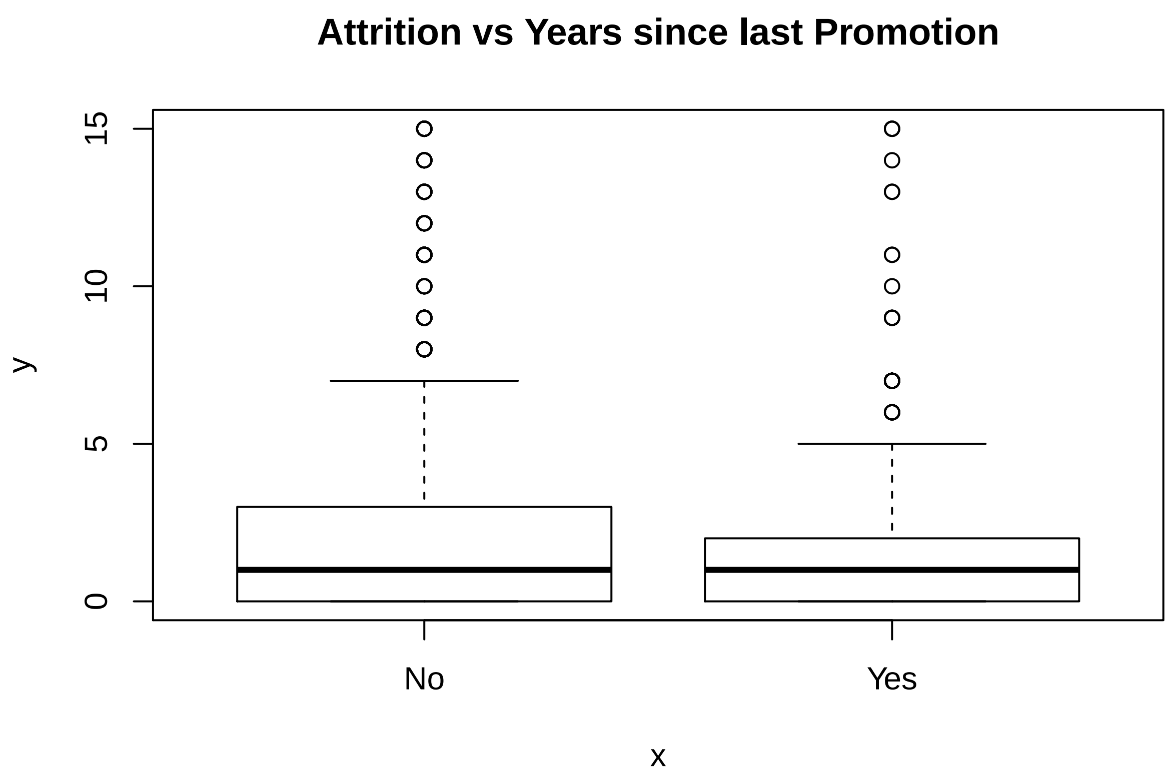

```
plot(x=data$Attrition, y=data$YearsAtCompany, main="Attrition vs Years at Company")
```



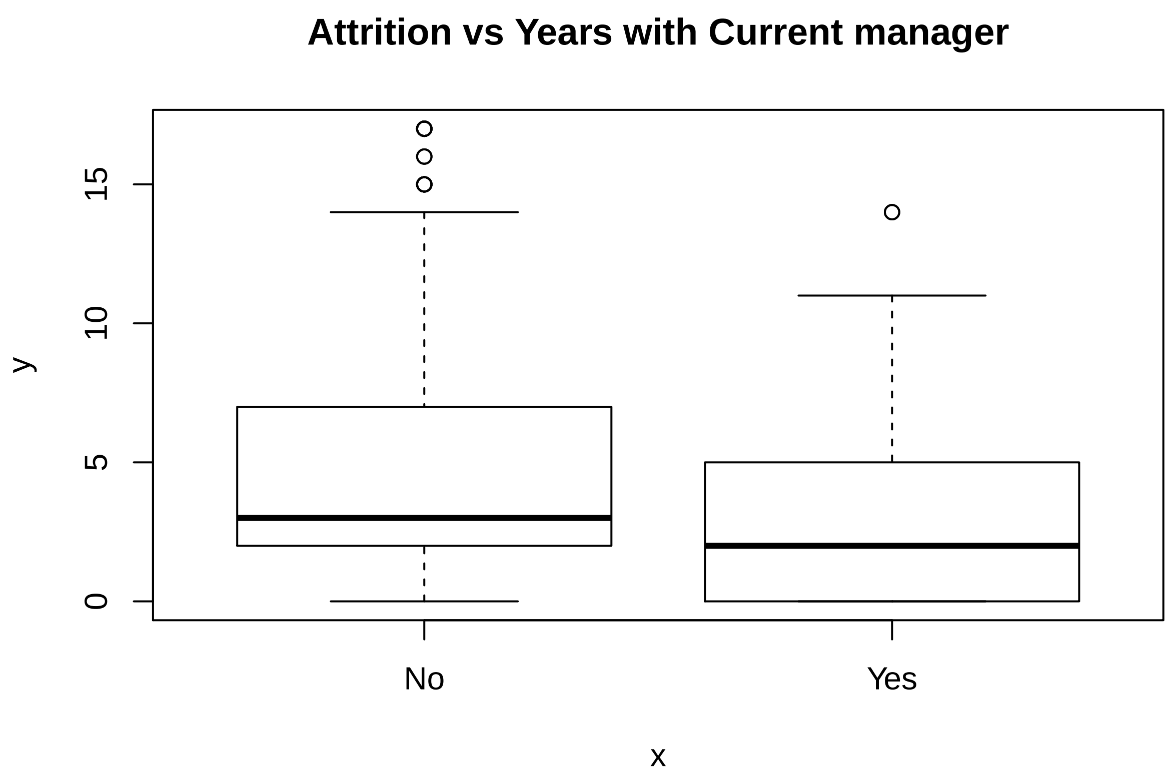

```r
plot(x=data$Attrition, y=data$YearsInCurrentRole, main="Attrition vs Years in Current Role")
```




```
plot(x=data$Attrition, y=data$YearsSinceLastPromotion, main="Attrition vs Years since last Promotion")
```

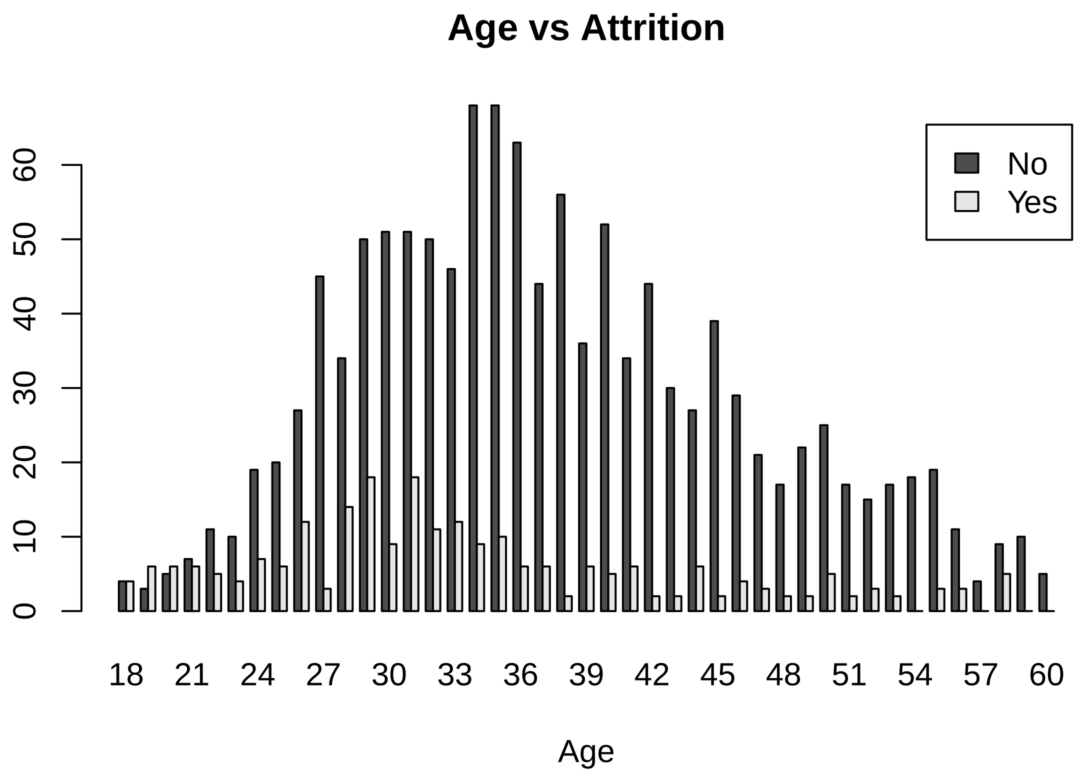



```r
plot(x=data$Attrition, y=data$YearsWithCurrManager, main="Attrition vs Years with Current manager")
```

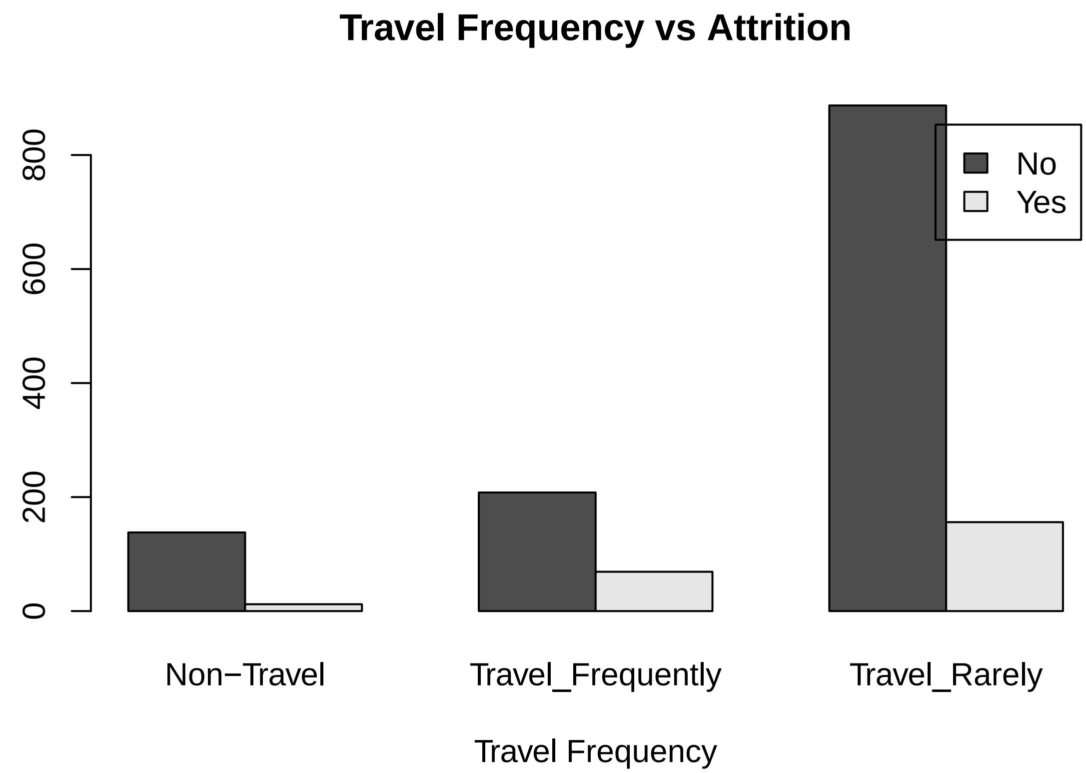


Independent Variable vs Dependent Variable (Bar Charts)

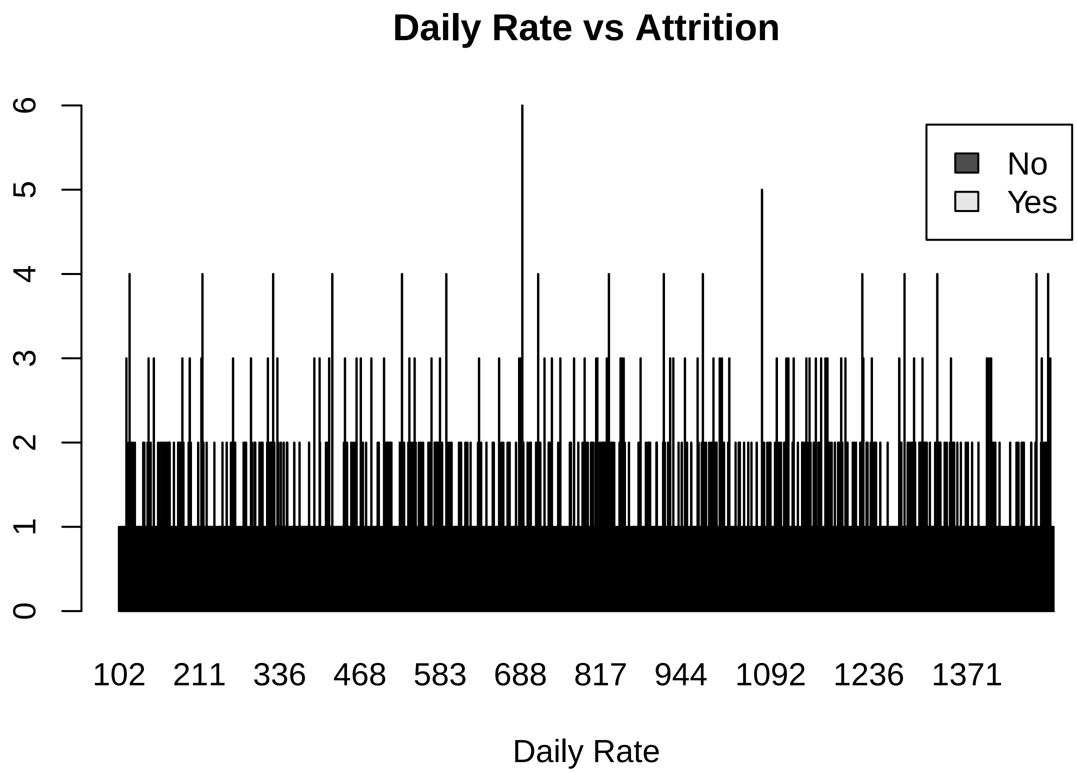
```
plot_age = table(data$Attrition, data$Age)
barplot(plot_age, main="Age vs Attrition", xlab="Age", legend=rownames(plot_age), beside = TRUE)
```



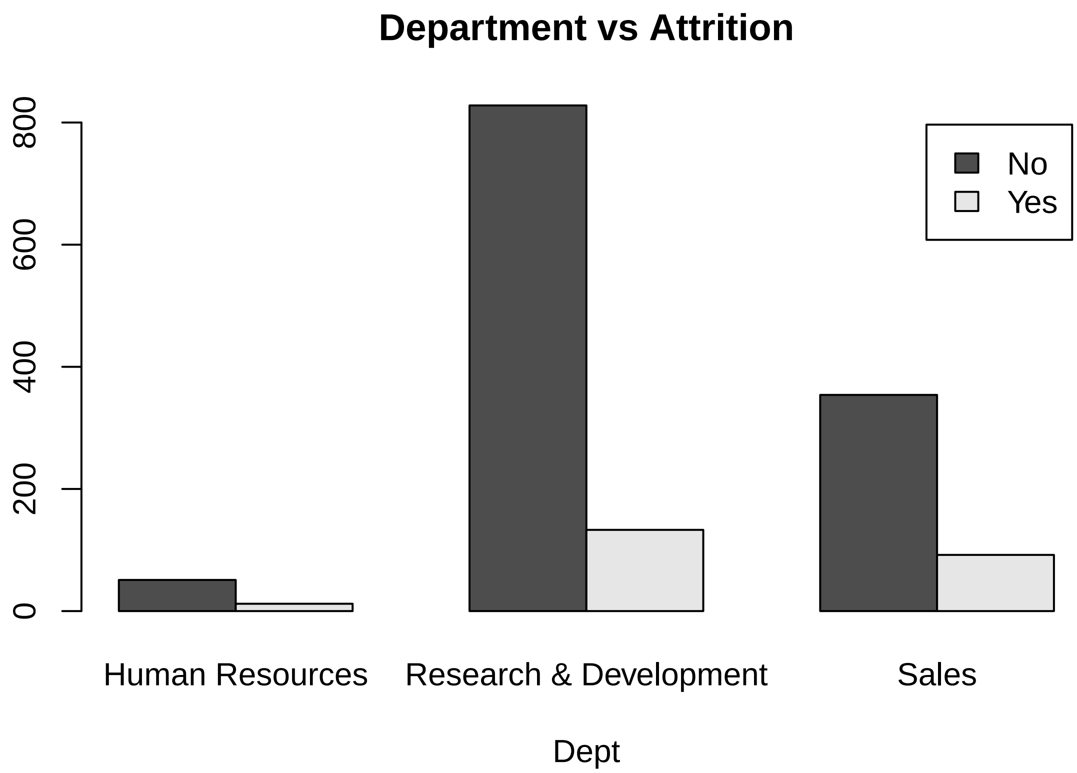

```
#
plot_travel = table(data$Attrition, data$BusinessTravel)
barplot(plot_travel, main = "Travel Frequency vs Attrition", xlab = "Travel Frequency", legend = rownam
```



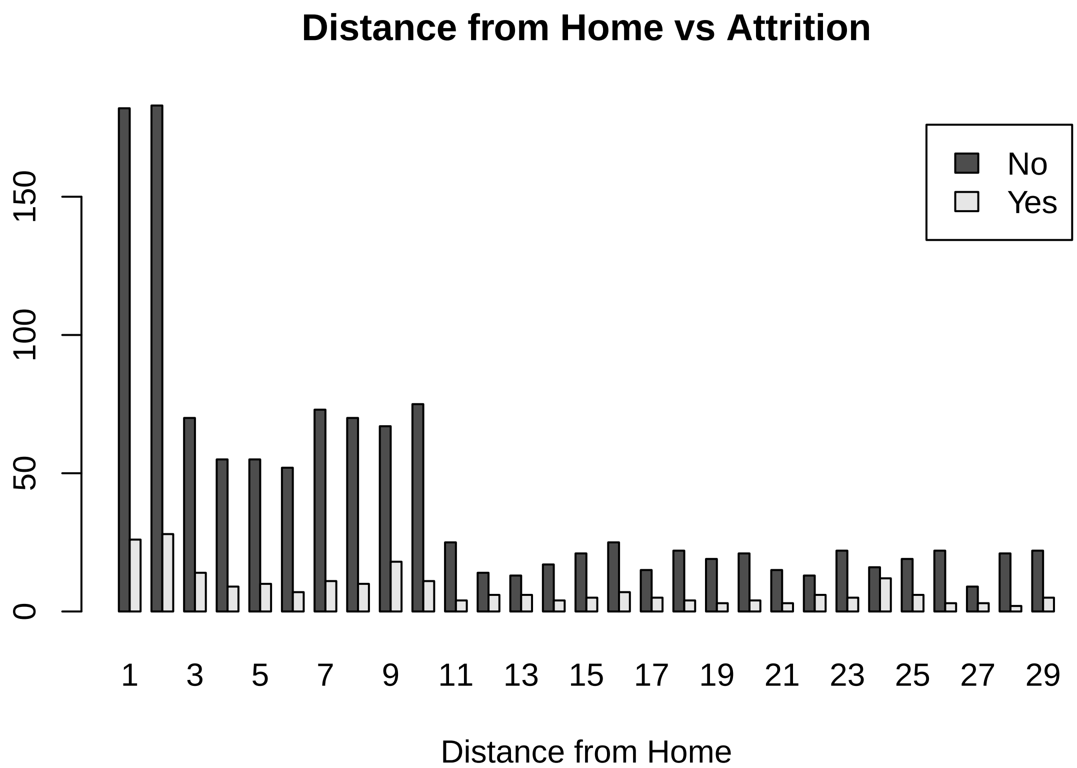

```r
#
plot_dailyrate = table(data$Attrition, data$DailyRate)
barplot(plot_dailyrate, main = "Daily Rate vs Attrition", xlab = "Daily Rate", legend=rownames(plot_dai
```



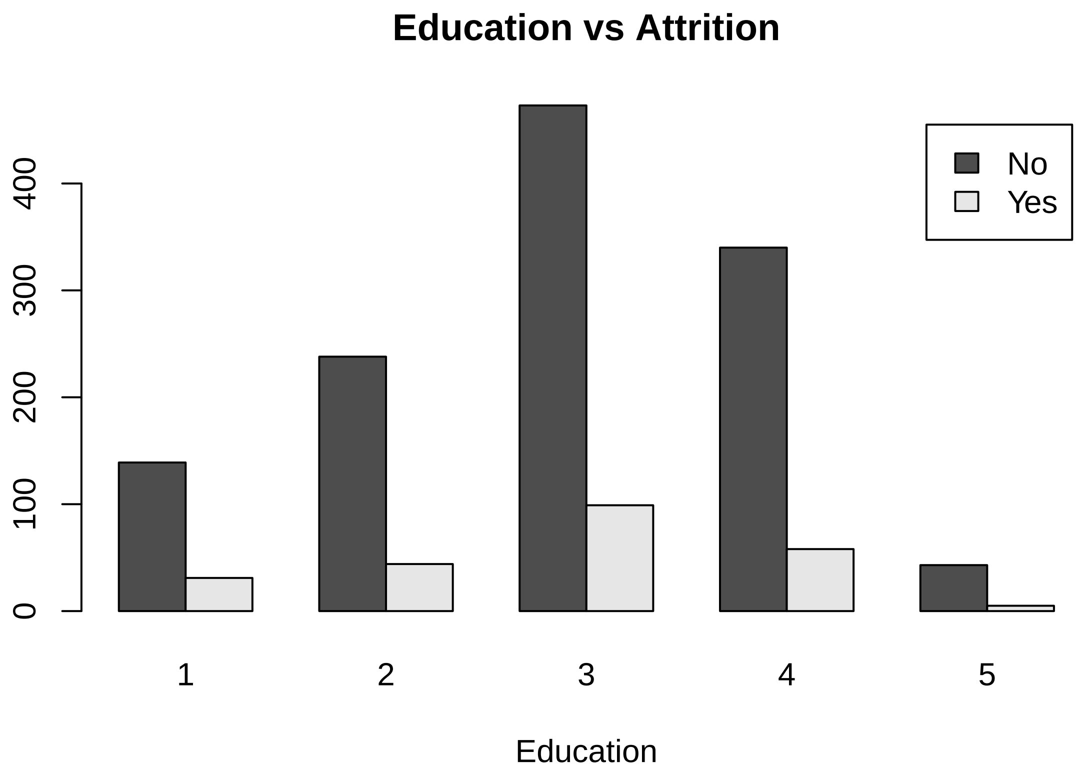

```
#
plot_dept = table(data$Attrition, data$Department)
barplot(plot_dept, main = "Department vs Attrition", xlab ="Dept", legend=rownames(plot_dept), beside =
```



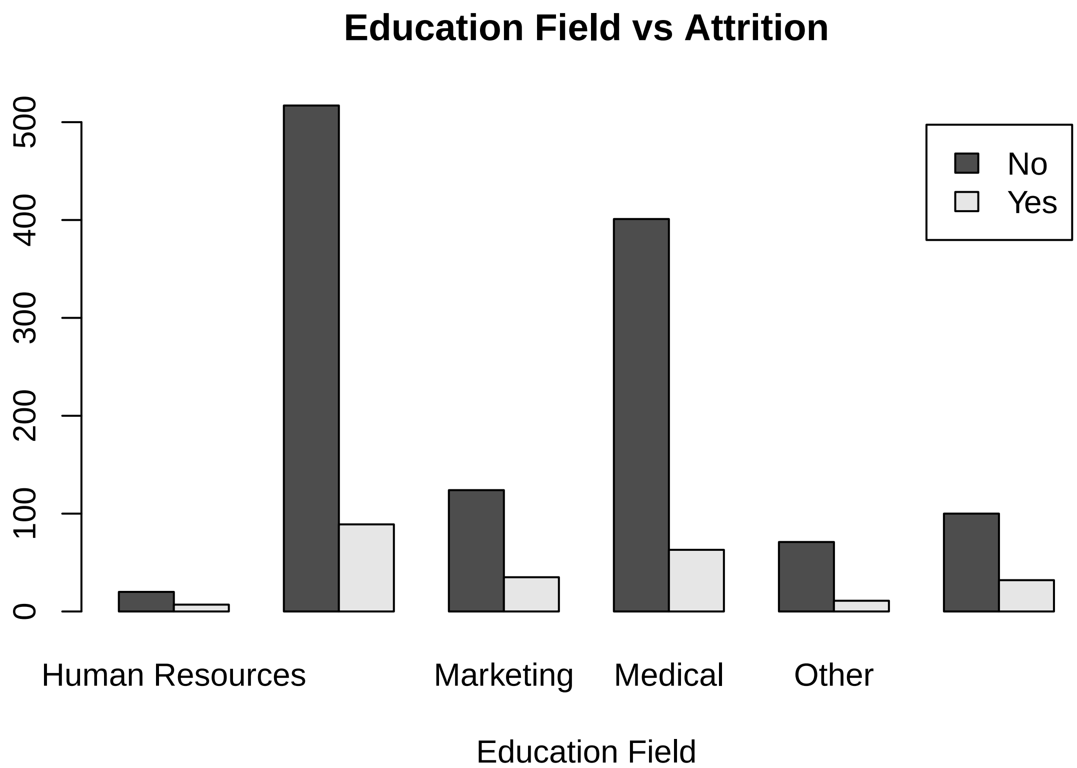

```
#
plot_distance = table(data$Attrition, data$DistanceFromHome)
barplot(plot_distance, main = "Distance from Home vs Attrition", xlab="Distance from Home", legend=rowna
```



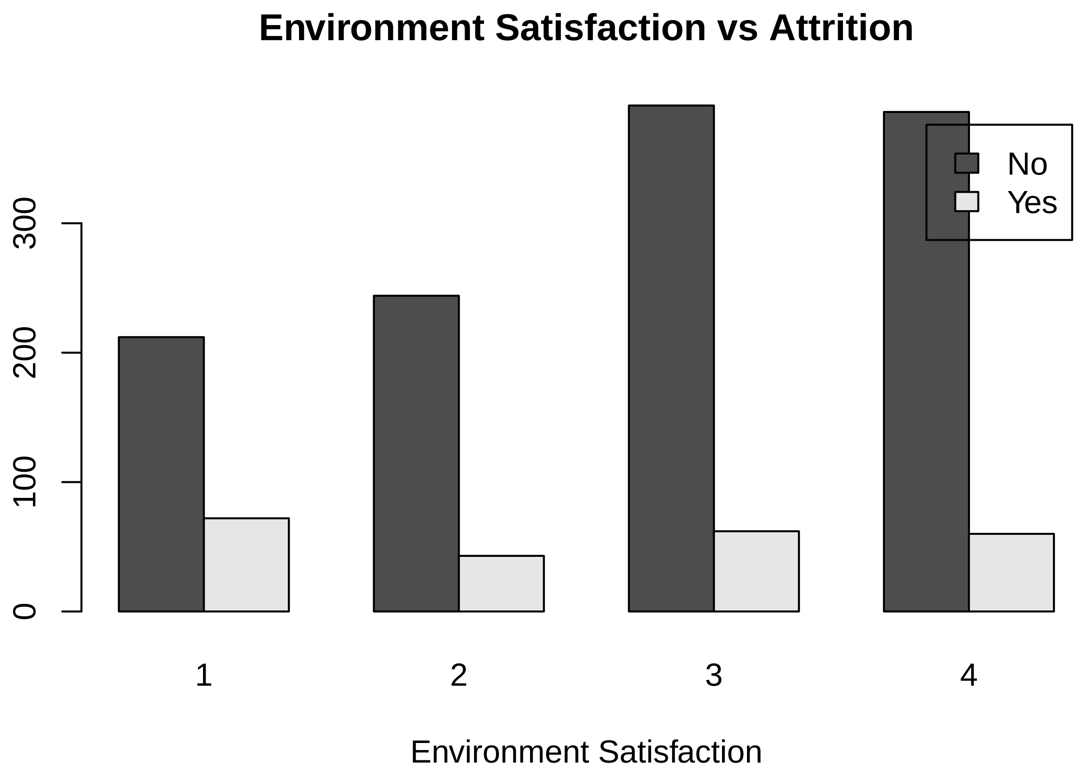

```
#
plot_education = table(data$Attrition, data$Education)
barplot(plot_education, main ="Education vs Attrition", xlab="Education", legend=rownames(plot_educatio
```



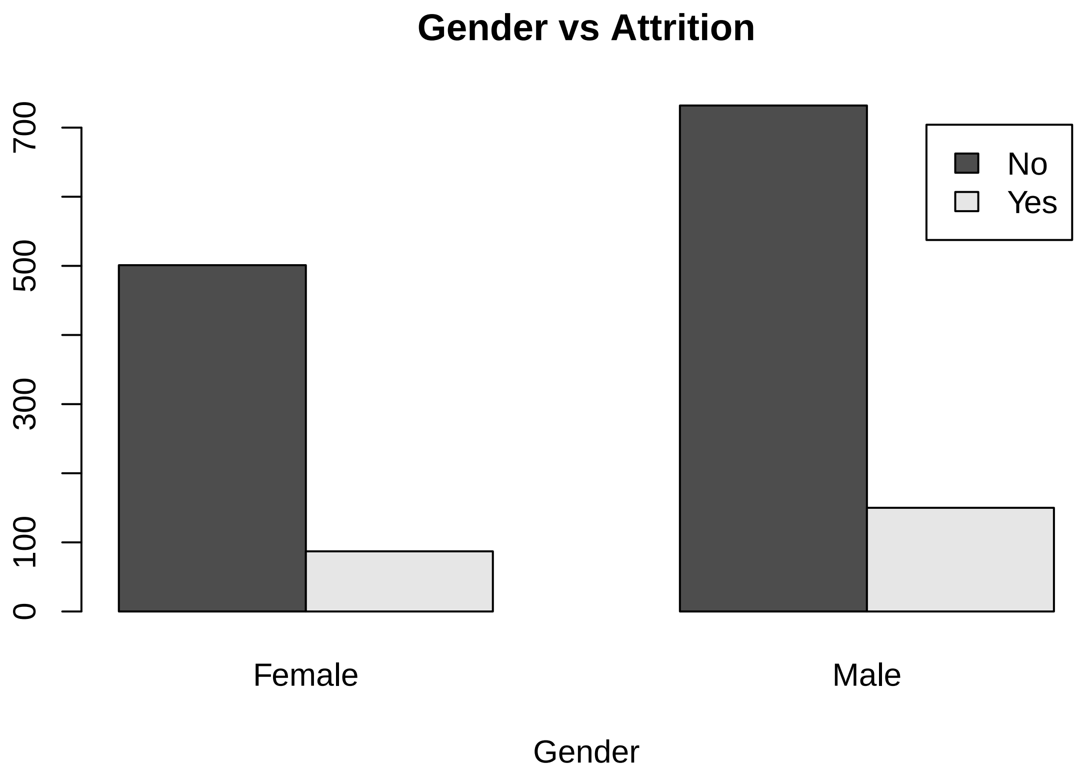

```
#
plot_field = table(data$Attrition, data$EducationField)
barplot(plot_field, main = "Education Field vs Attrition", xlab="Education Field", legend=rownames(plot_
```



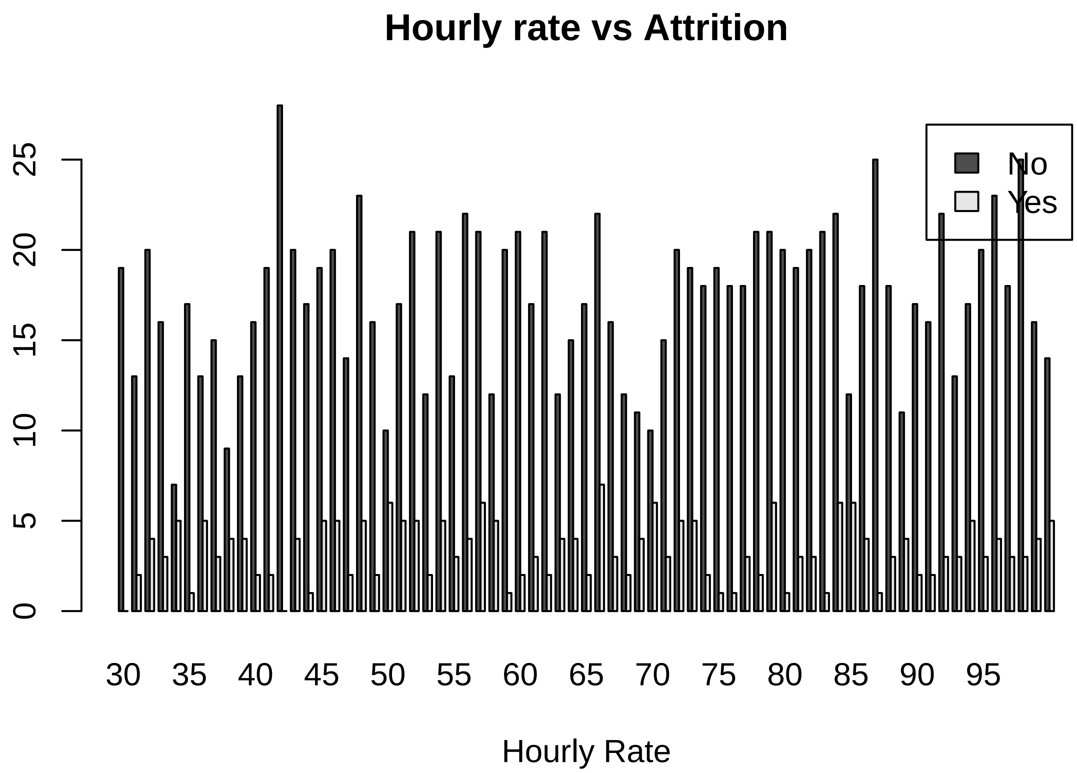

```
#
plot_envsat = table(data$Attrition, data$EnvironmentSatisfaction)
barplot(plot_envsat, main="Environment Satisfaction vs Attrition", xlab="Environment Satisfaction", lege
```



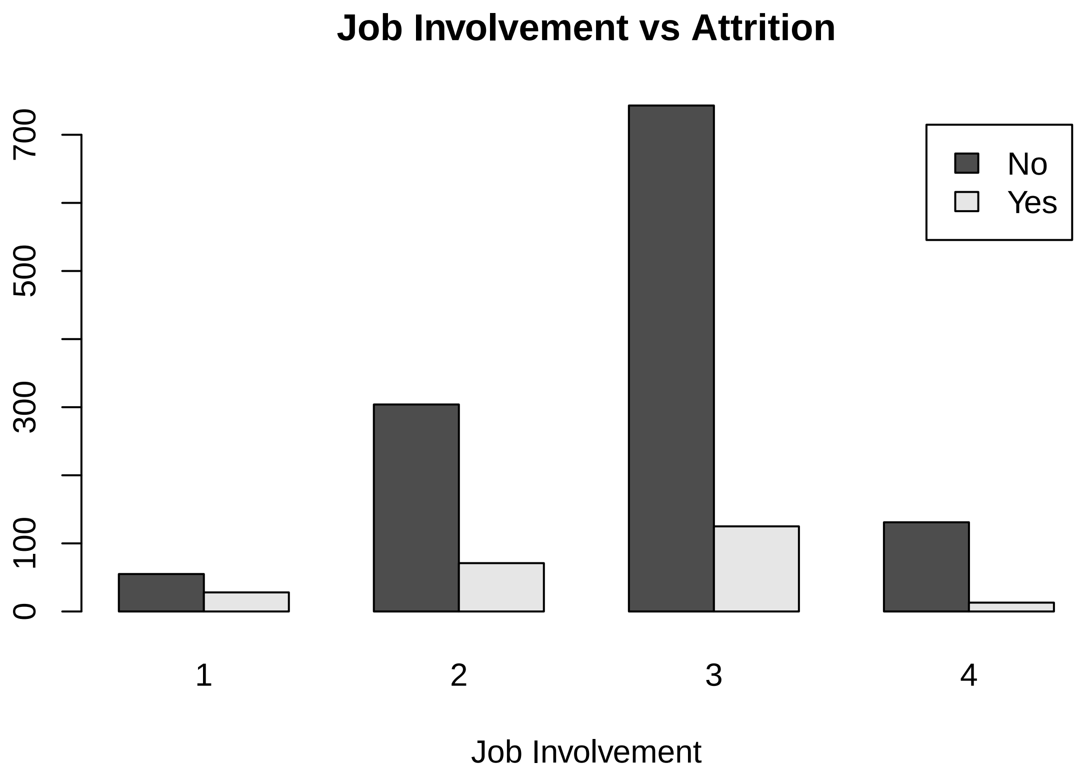

```
#
plot_gender = table(data$Attrition, data$Gender)
barplot(plot_gender, main="Gender vs Attrition", xlab = "Gender", legend=rownames(plot_gender), beside='
```



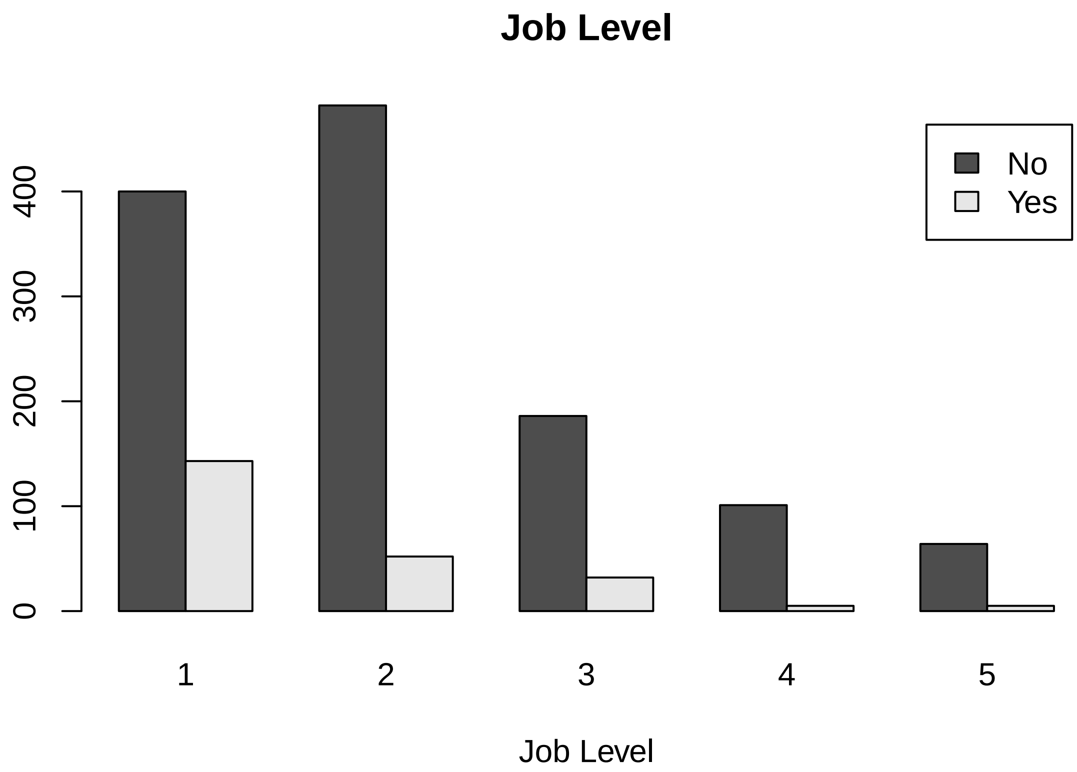

```
#
plot_hourlyrate = table(data$Attrition, data$HourlyRate)
barplot(plot_hourlyrate, main="Hourly rate vs Attrition", xlab="Hourly Rate", legend=rownames(plot_hour
```



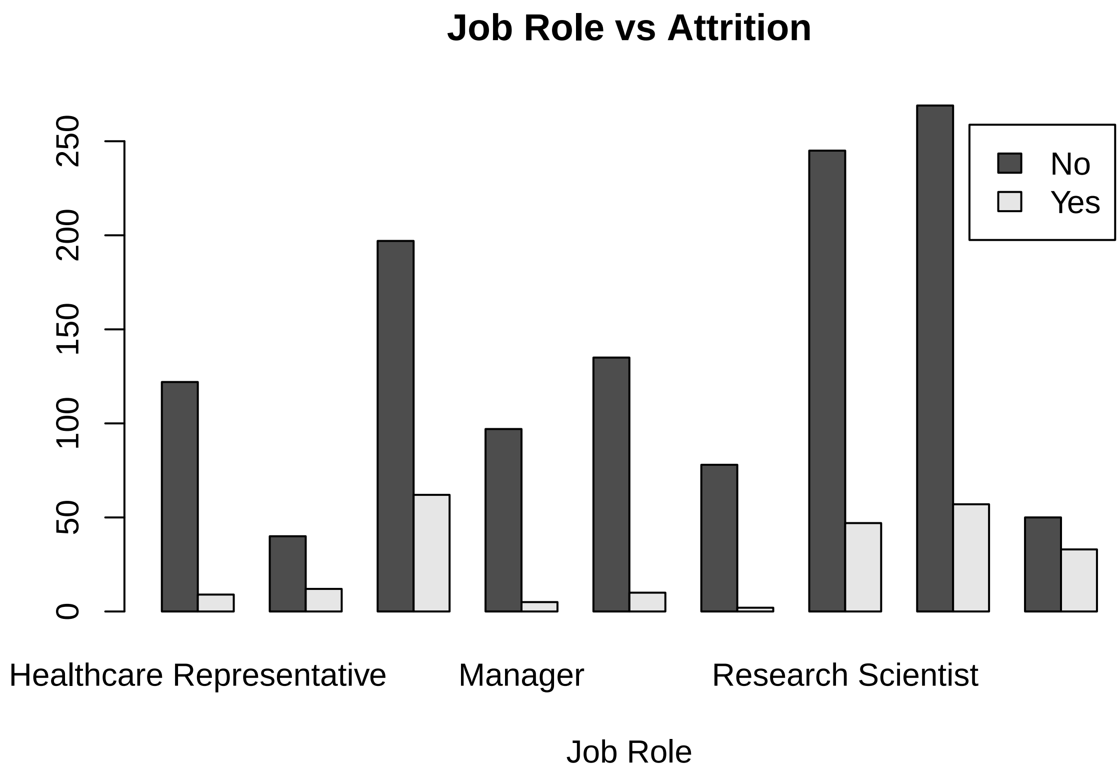

```
#
plot_involvement = table(data$Attrition, data$JobInvolvement)
barplot(plot_involvement, main="Job Involvement vs Attrition", xlab="Job Involvement", legend=rownames(p
```



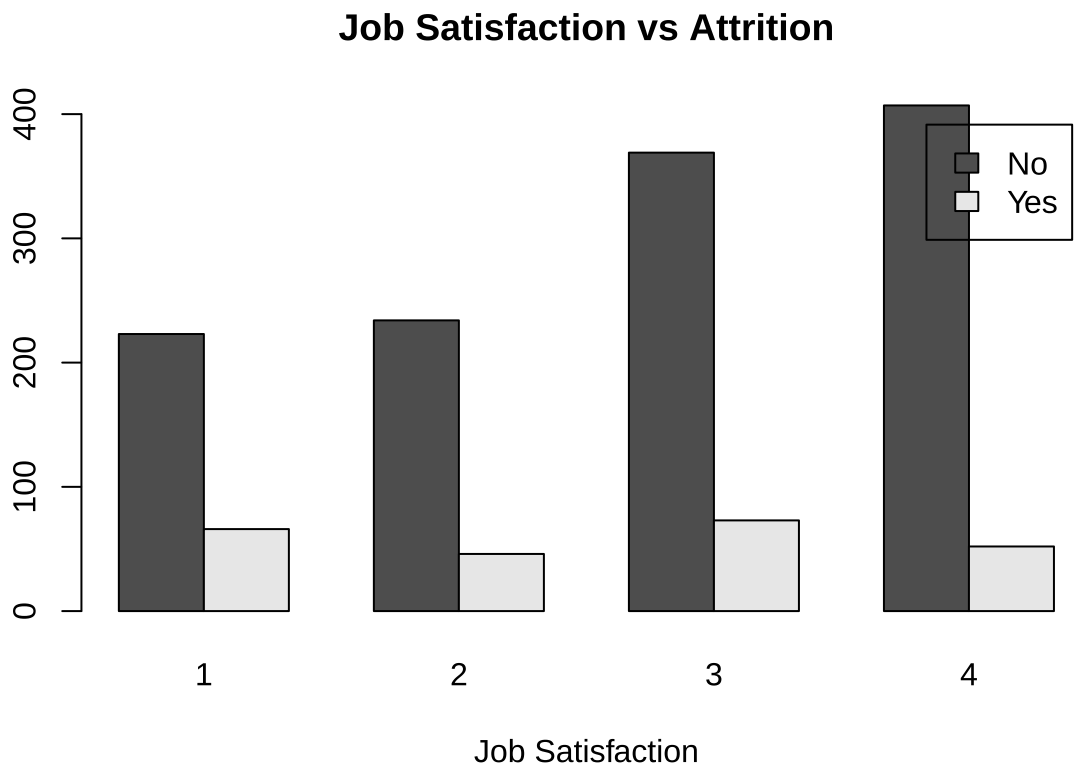

```
#
plot_joblevel = table(data$Attrition, data$JobLevel)
barplot(plot_joblevel, main="Job Level", xlab="Job Level", legend=rownames(plot_joblevel), beside = TRUI
```



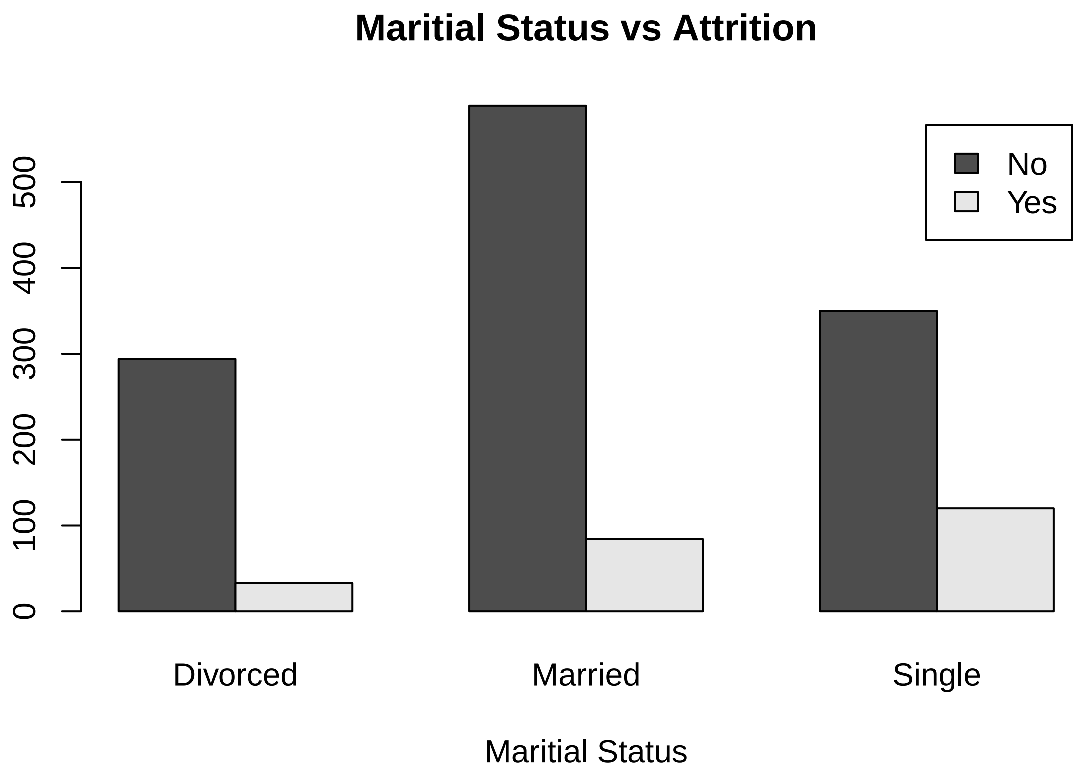

```
#
plot_jobrole = table(data$Attrition, data$JobRole)
barplot(plot_jobrole, main="Job Role vs Attrition", xlab = "Job Role", legend=rownames(plot_jobrole), be
```



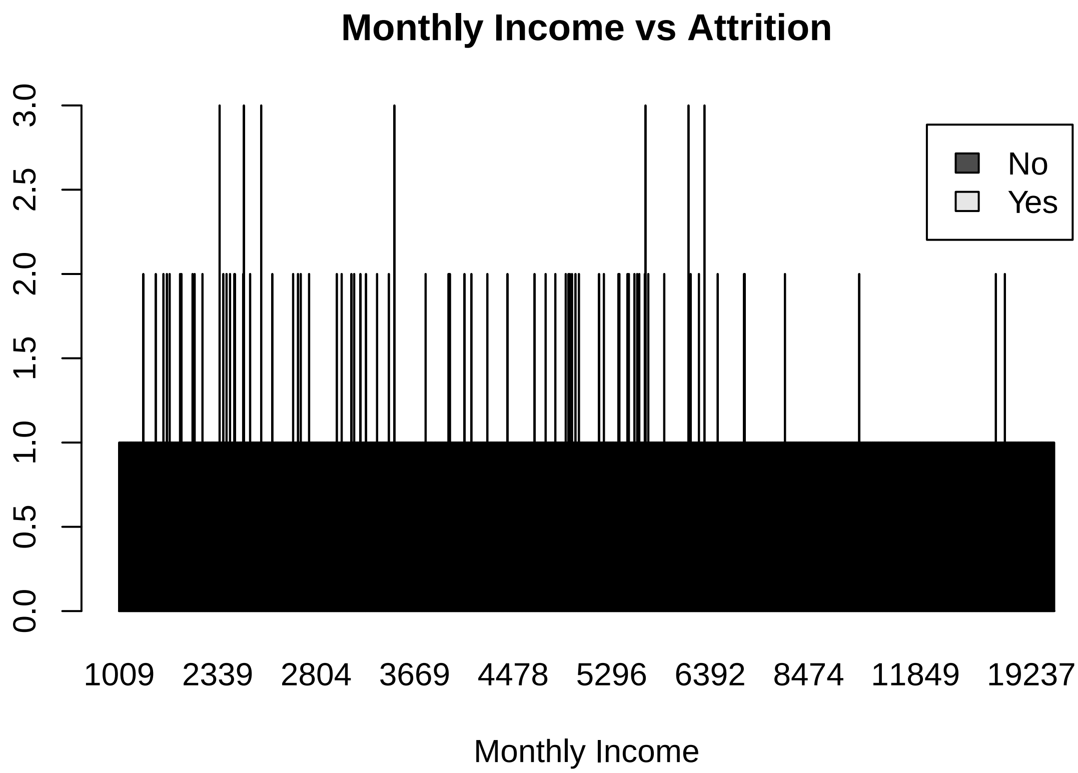

```
#
plot_jobsat = table(data$Attrition, data$JobSatisfaction)
barplot(plot_jobsat, main="Job Satisfaction vs Attrition", xlab="Job Satisfaction", legend=rownames(plo
```



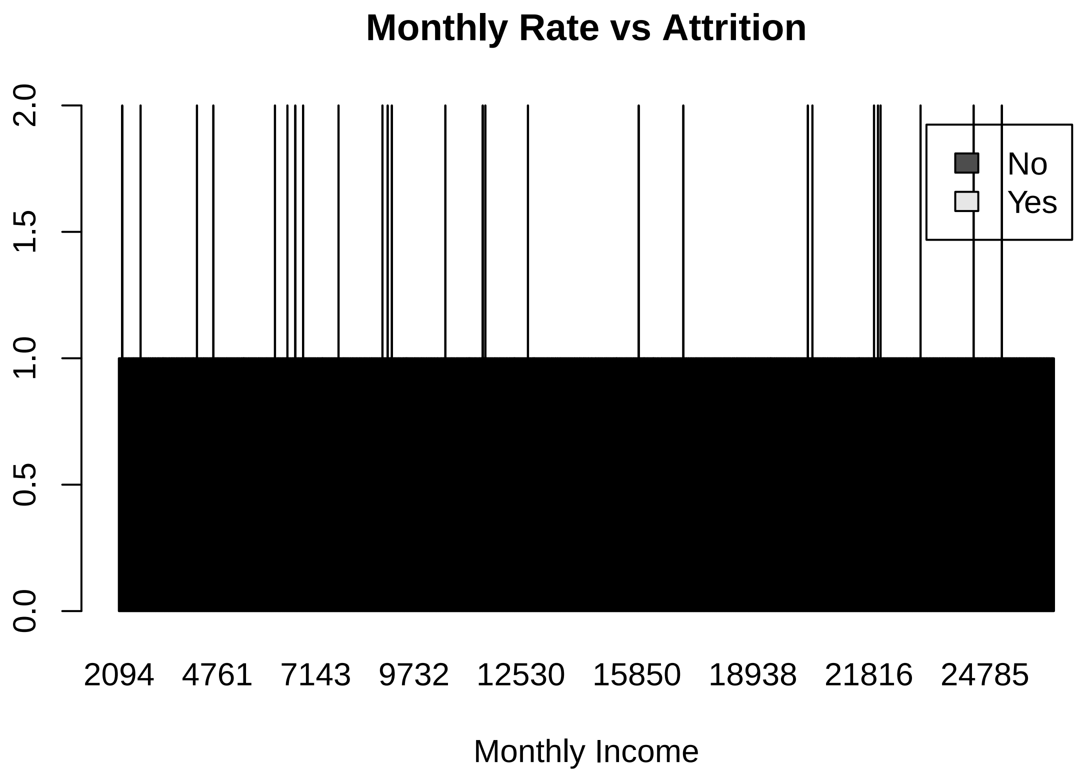

```
#
plot_maritial = table(data$Attrition, data$MaritalStatus)
barplot(plot_maritial, main="Maritial Status vs Attrition", xlab="Maritial Status", legend=rownames(plo
```



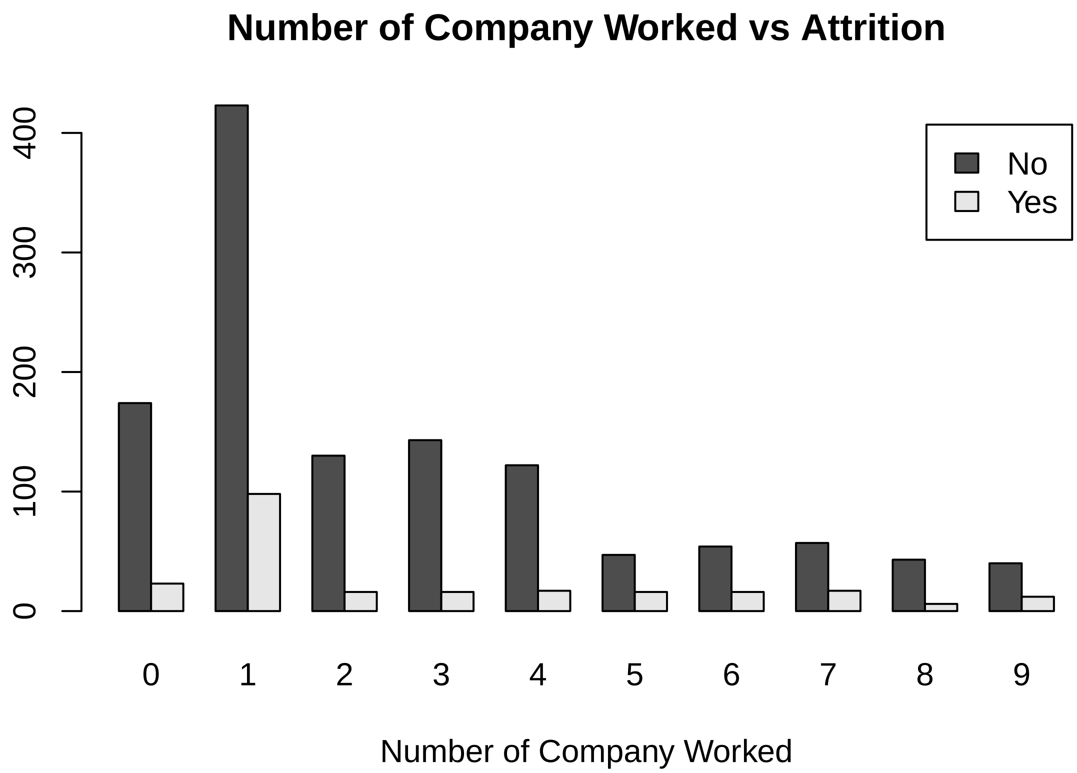

```
#
plot_monthlyincome = table(data$Attrition, data$MonthlyIncome)
barplot(plot_monthlyincome, main="Monthly Income vs Attrition", xlab = "Monthly Income", legend=rowname
```



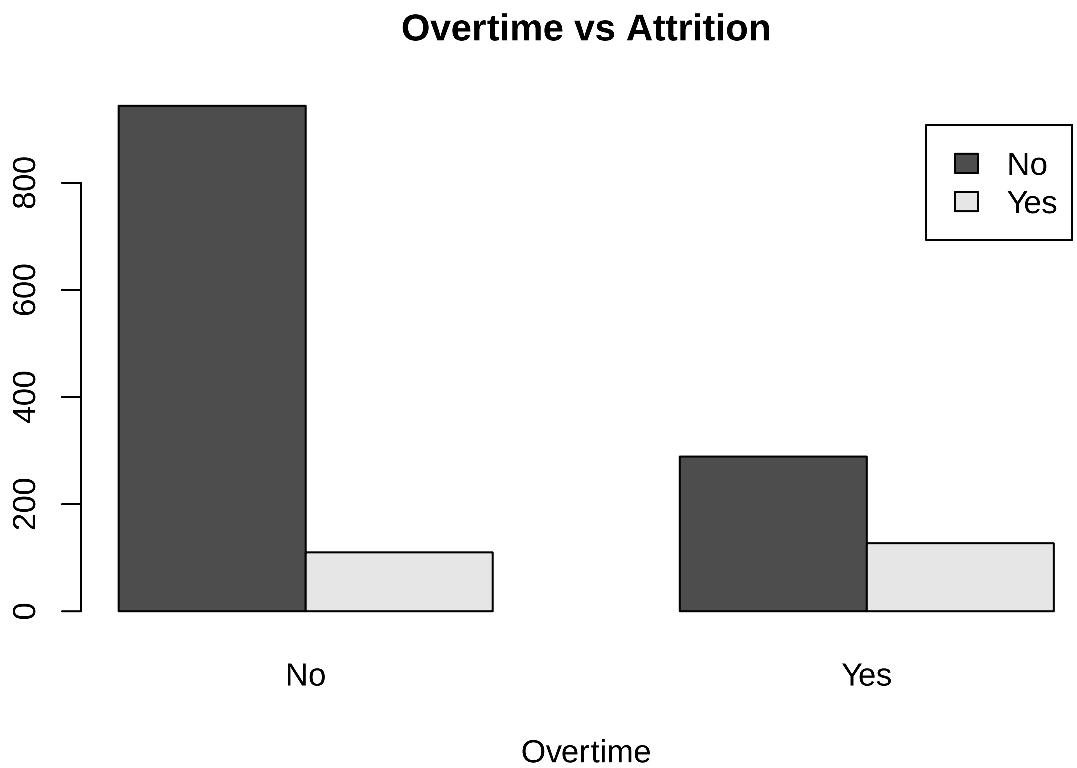

```
#
plot_monthlyrate = table(data$Attrition, data$MonthlyRate)
barplot(plot_monthlyrate, main="Monthly Rate vs Attrition", xlab = "Monthly Income", legend=rownames(pl
```



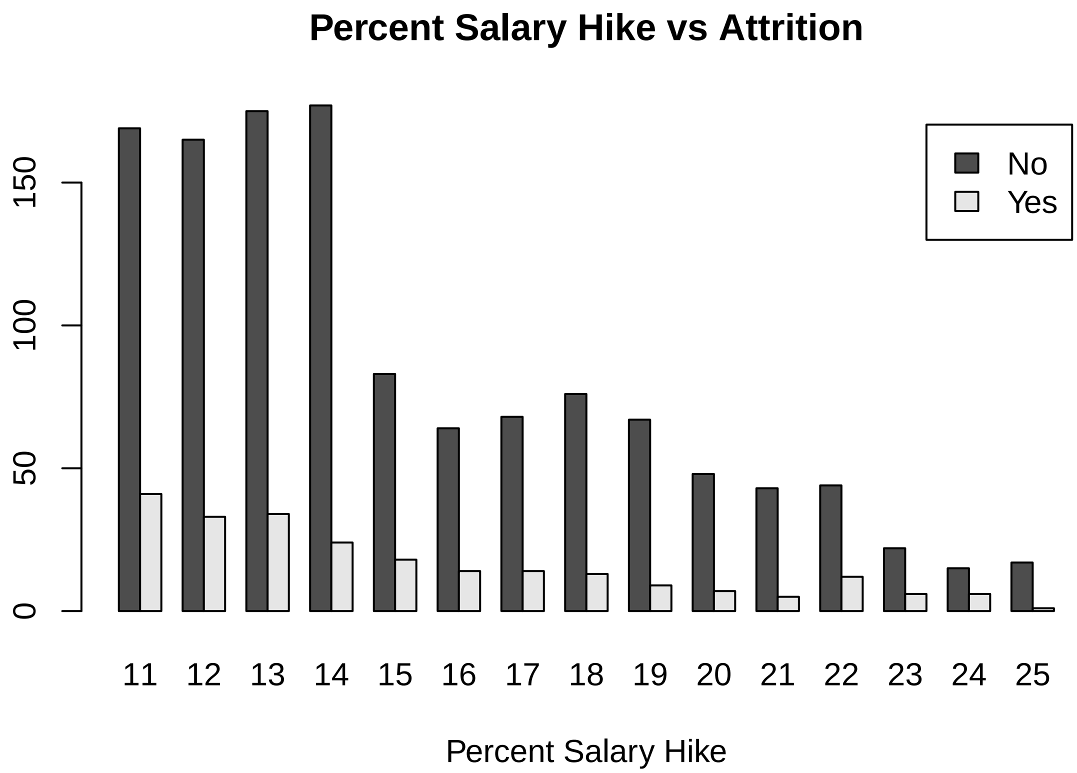

```
#
plot_numcompany = table(data$Attrition, data$NumCompaniesWorked)
barplot(plot_numcompany, main="Number of Company Worked vs Attrition", xlab="Number of Company Worked",
```



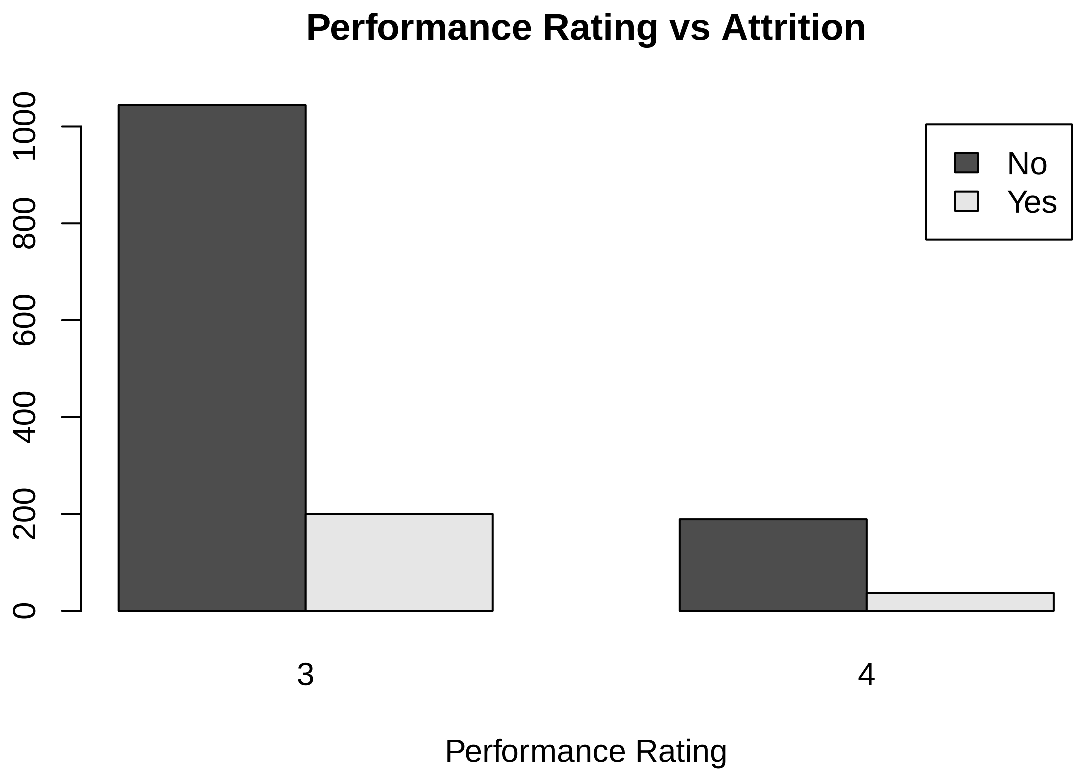

```
#
plot_overtime = table(data$Attrition, data$OverTime)
barplot(plot_overtime, main="Overtime vs Attrition", xlab="Overtime", legend=rownames(plot_overtime), be
```



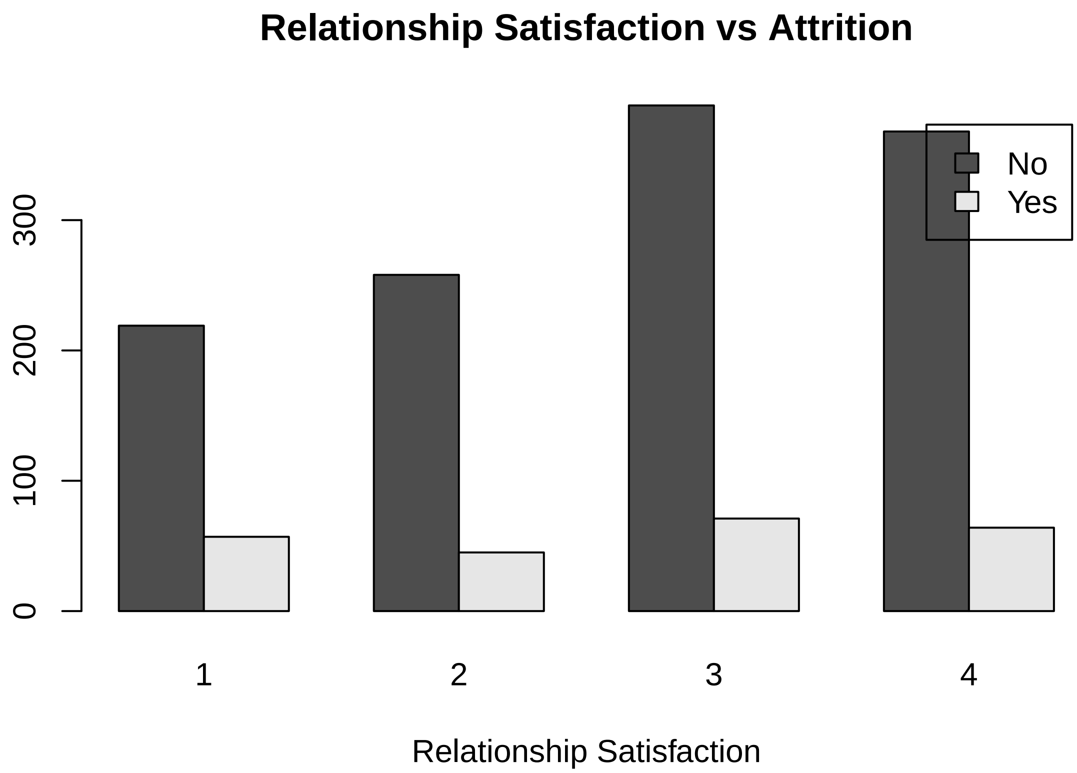

```
#
plot_salaryhike = table(data$Attrition, data$PercentSalaryHike)
barplot(plot_salaryhike, main="Percent Salary Hike vs Attrition", xlab="Percent Salary Hike", legend=ro
```



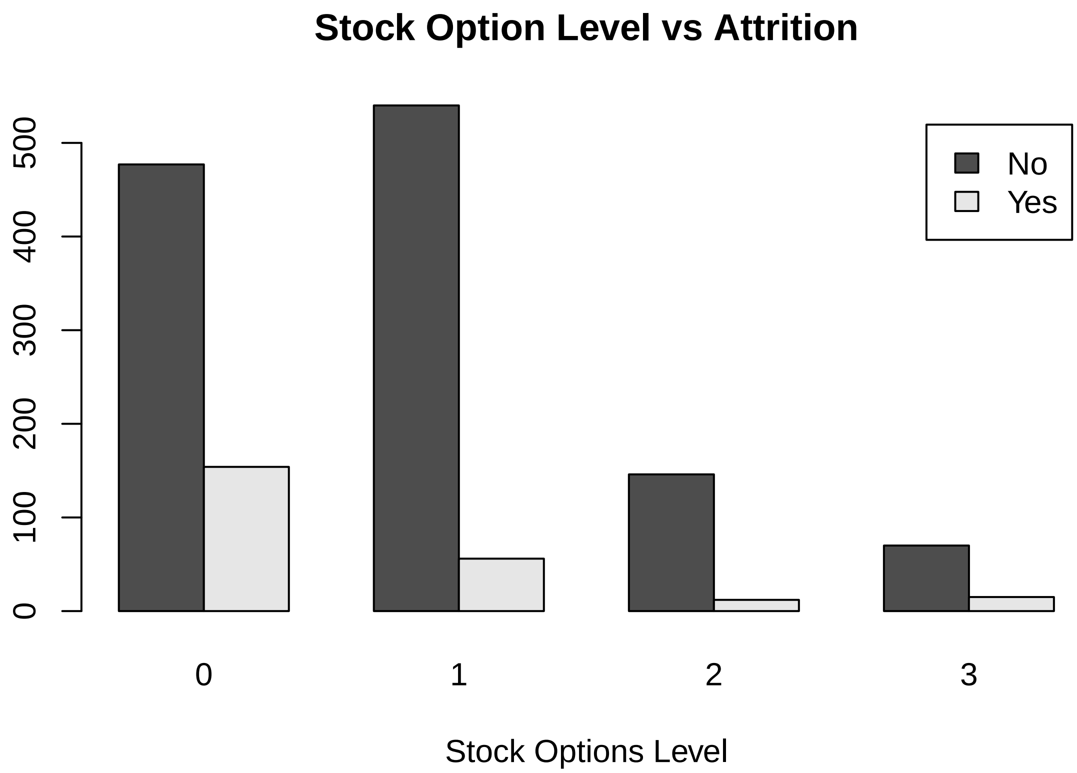

```
#
plot_rating = table(data$Attrition, data$PerformanceRating)
barplot(plot_rating, main="Performance Rating vs Attrition", xlab="Performance Rating", legend=rownames
```



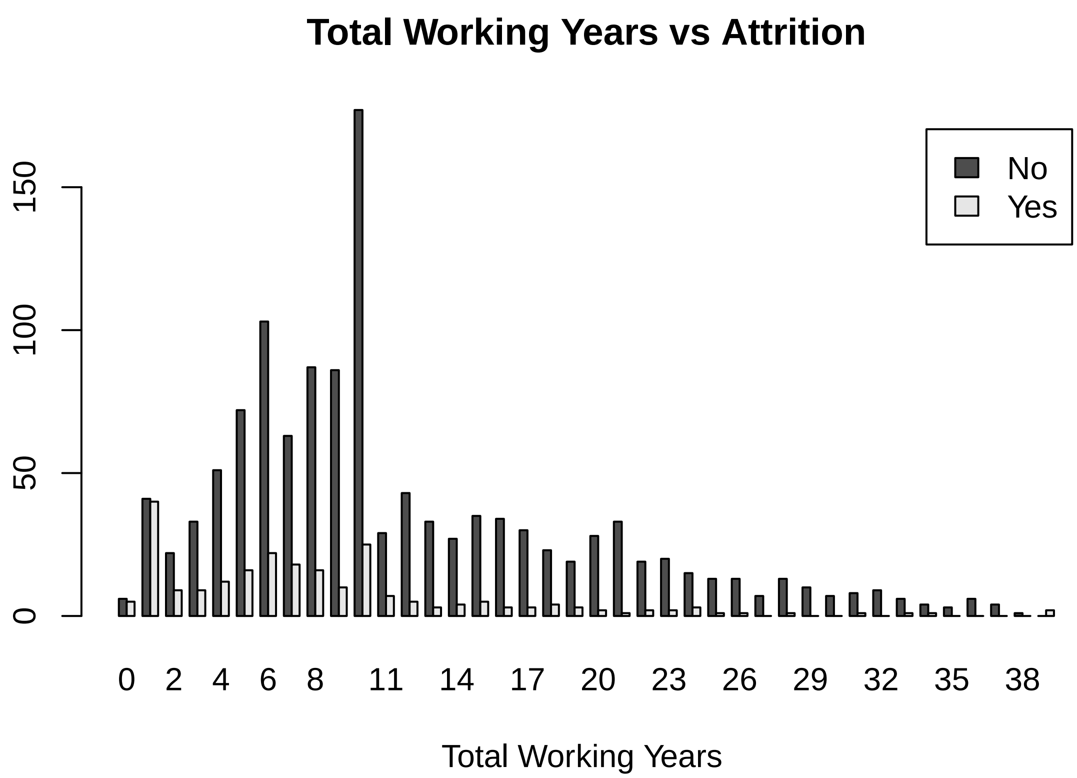

```
#
plot_relsat = table(data$Attrition, data$RelationshipSatisfaction)
barplot(plot_relsat, main="Relationship Satisfaction vs Attrition", xlab="Relationship Satisfaction", le
```



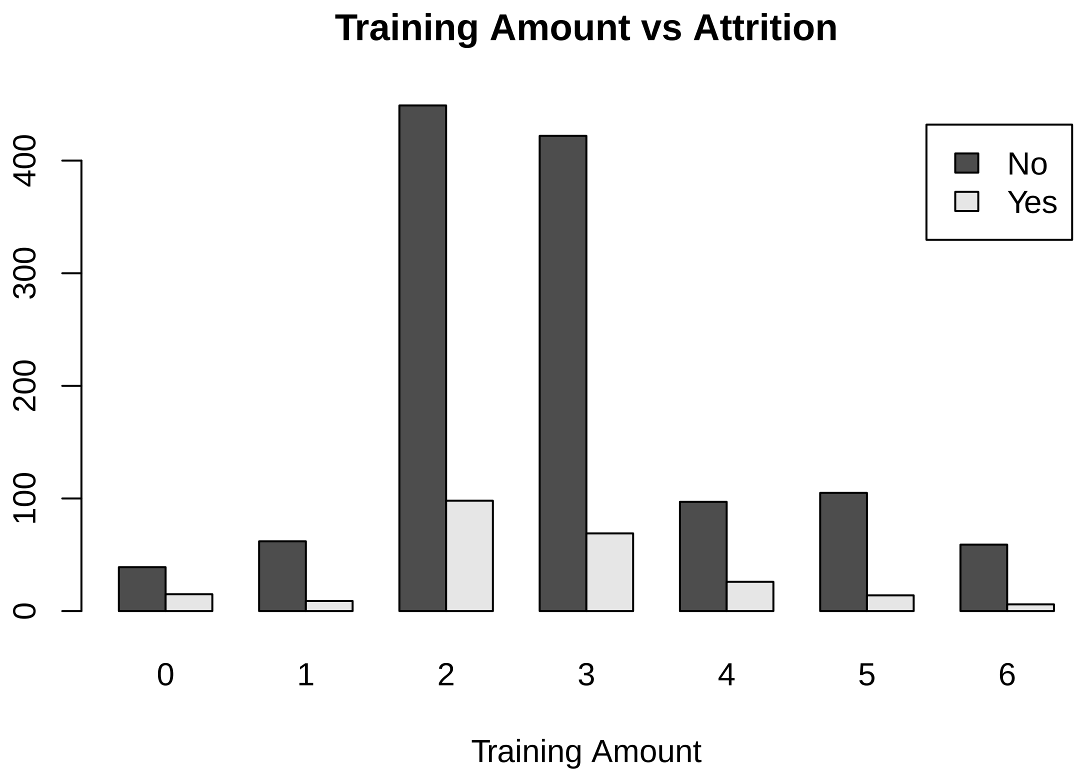

```
#
plot_options = table(data$Attrition, data$StockOptionLevel)
barplot(plot_options, main="Stock Option Level vs Attrition", xlab="Stock Options Level", legend=rowname
```



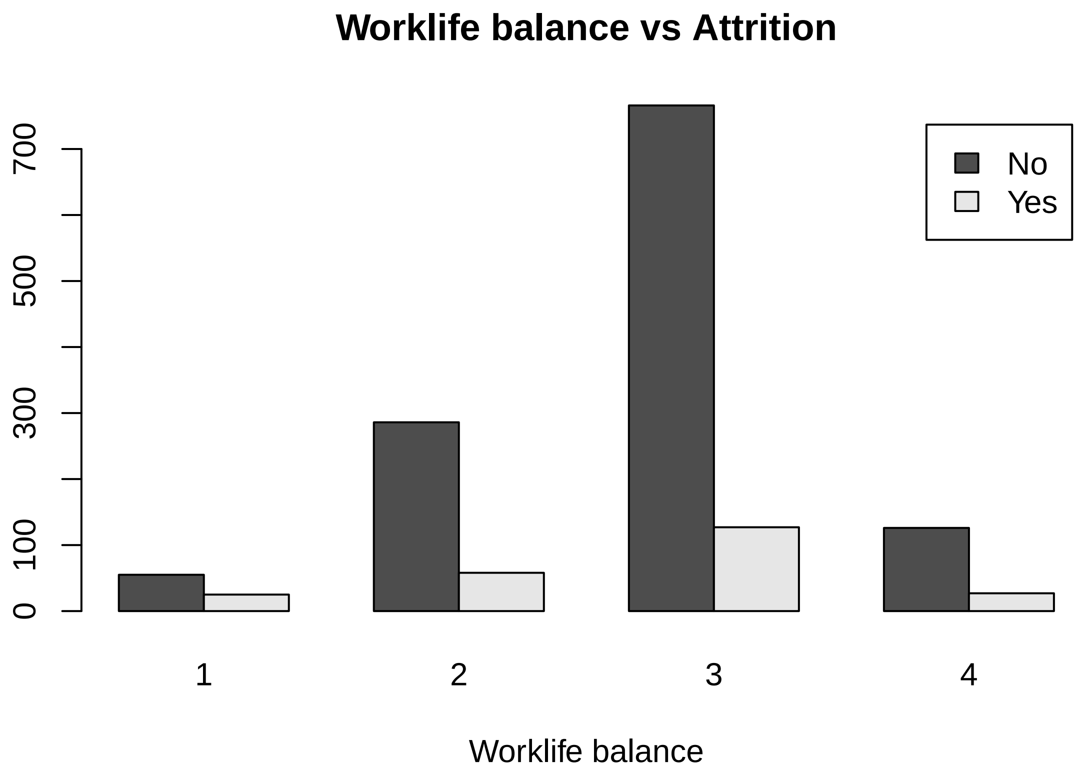

```
#
plot_totalworkyears = table(data$Attrition, data$TotalWorkingYears)
barplot(plot_totalworkyears, main ="Total Working Years vs Attrition", xlab = "Total Working Years ", le
```



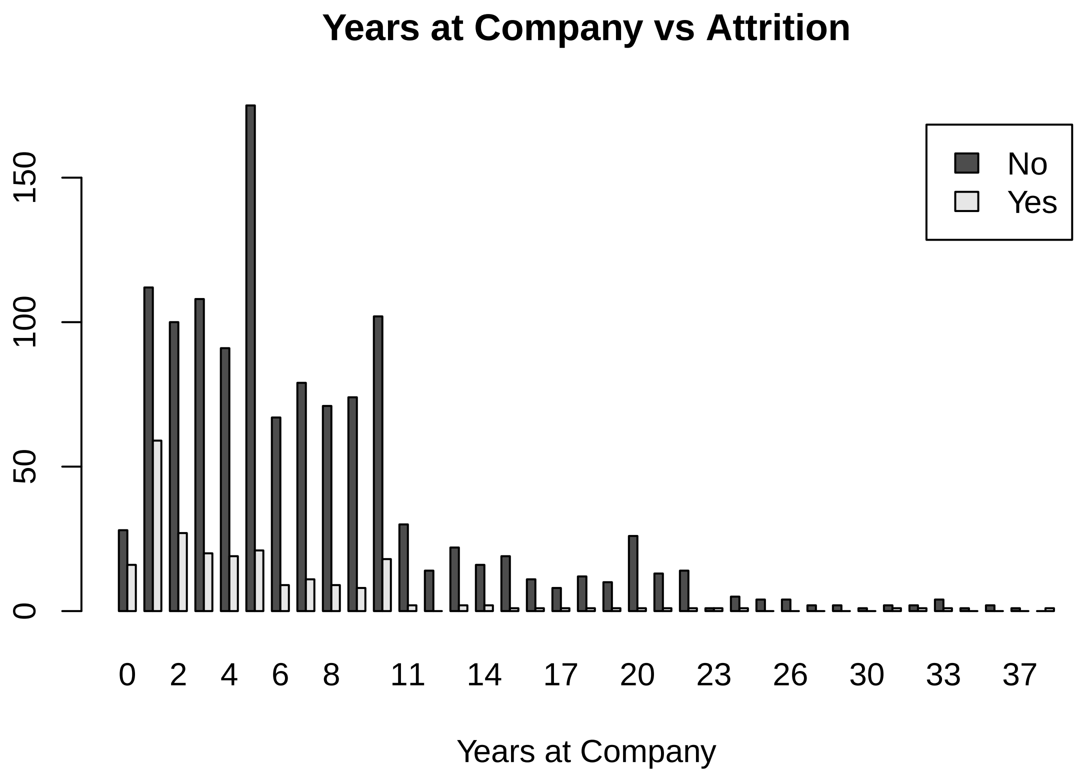

```r
#
plot_training = table(data$Attrition, data$TrainingTimesLastYear)
barplot(plot_training, main="Training Amount vs Attrition", xlab="Training Amount", legend=rownames(plo
```




```
#
plot_worklife = table(data$Attrition, data$WorkLifeBalance)
barplot(plot_worklife, main="Worklife balance vs Attrition", xlab="Worklife balance", legend=rownames(pl
```

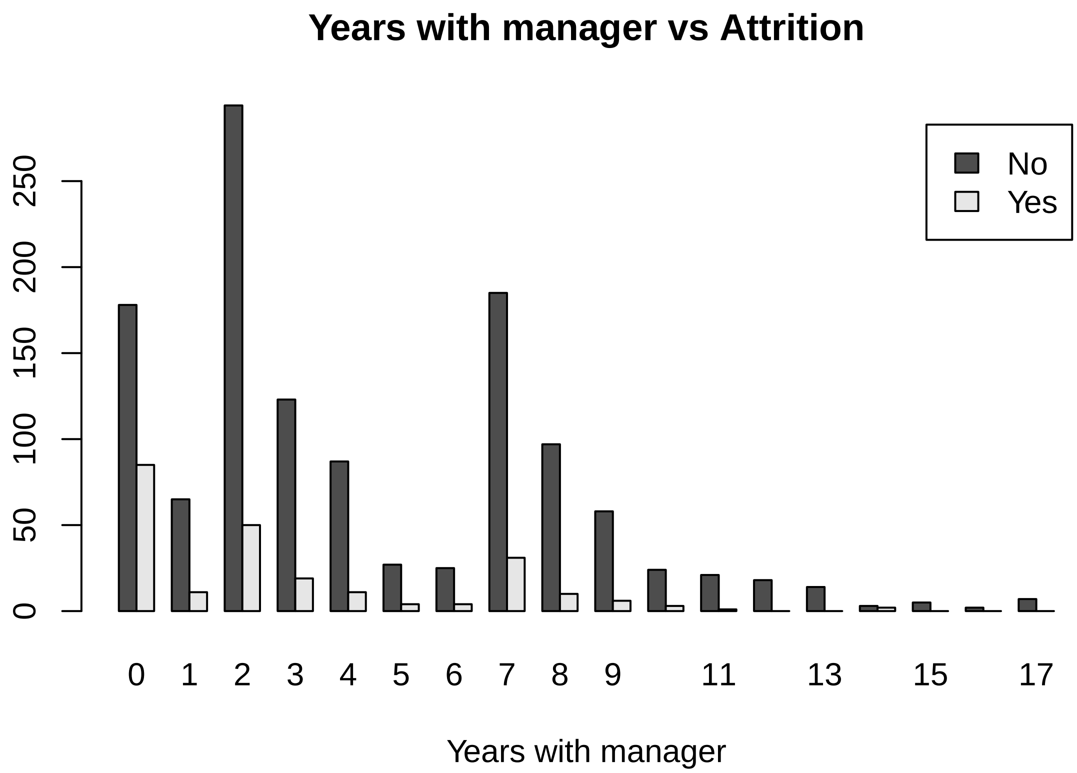



```
#
plot_yearscompany = table(data$Attrition, data$YearsAtCompany)
barplot(plot_yearscompany, main="Years at Company vs Attrition", xlab="Years at Company", legend=rownam
```

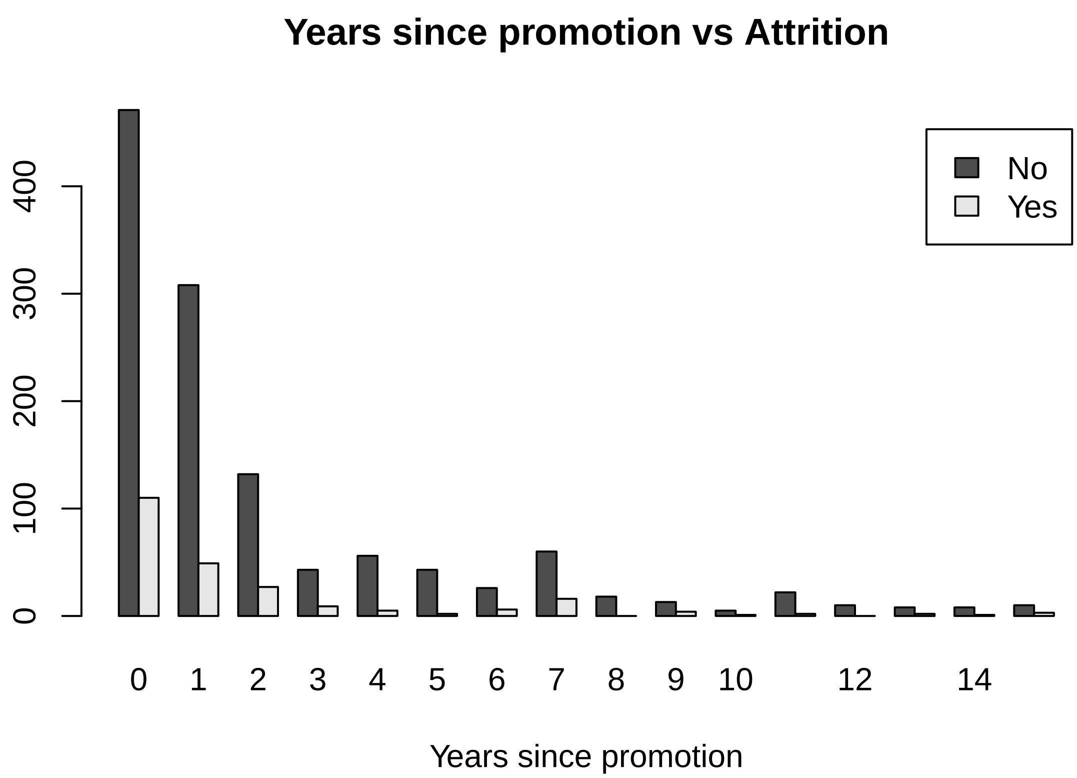



```
#
plot_yearsrole = table(data$Attrition, data$YearsInCurrentRole)
barplot(plot_yearsrole, main="Year in Role vs Attrition", xlab="Years in Role", legend = rownames(plot_
```


```
#
plot_yearsmanager = table(data$Attrition, data$YearsWithCurrManager)
barplot(plot_yearsmanager, main="Years with manager vs Attrition", xlab = "Years with manager", legend=
```




```
#
plot_promotion = table(data$Attrition, data$YearsSinceLastPromotion)
barplot(plot_promotion, main="Years since promotion vs Attrition", xlab="Years since promotion", legend=
```



Categorical Variables vs Dependent Variable (Frequency Tables)

```
plot_travel = table(data$Attrition, data$BusinessTravel)
prop.table(plot_travel, 2)
```

```
##
##       Non-Travel Travel_Frequently Travel_Rarely
##   No   0.9200000         0.7509025     0.8504314
##   Yes  0.0800000         0.2490975     0.1495686
```

```
#
plot_dept = table(data$Attrition, data$Department)
prop.table(plot_dept, 2)
```

```
##
##       Human Resources Research & Development     Sales
##   No        0.8095238              0.8616025 0.7937220
##   Yes       0.1904762              0.1383975 0.2062780
```

```
#
plot_education = table(data$Attrition, data$Education)
Names = c("Below College", "College", "bachelor", "Masters", "PHD")
colnames(plot_education) <- Names
prop.table(plot_education, 2)
```

```
##
##       Below College   College  bachelor   Masters       PHD
##   No      0.8176471 0.8439716 0.8269231 0.8542714 0.8958333
##   Yes     0.1823529 0.1560284 0.1730769 0.1457286 0.1041667
```

```r
#
plot_field = table(data$Attrition, data$EducationField)
prop.table(plot_field, 2)
```

```
##
##       Human Resources Life Sciences Marketing   Medical     Other
##   No        0.7407407     0.8531353 0.7798742 0.8642241 0.8658537
##   Yes       0.2592593     0.1468647 0.2201258 0.1357759 0.1341463
##
##       Technical Degree
##   No         0.7575758
##   Yes        0.2424242
```

```r
#
plot_envsat = table(data$Attrition, data$EnvironmentSatisfaction)
Names = c("Low", "Medium", "High", "Very High")
colnames(plot_envsat) <- Names
prop.table(plot_envsat, 2)
```

```
##
##             Low    Medium      High Very High
##   No  0.7464789 0.8501742 0.8631347 0.8654709
##   Yes 0.2535211 0.1498258 0.1368653 0.1345291
```

```r
#
plot_gender = table(data$Attrition, data$Gender)
prop.table(plot_gender, 2)
```

```
##
##          Female      Male
##   No  0.8520408 0.8299320
##   Yes 0.1479592 0.1700680
```

```r
#
plot_involvement = table(data$Attrition, data$JobInvolvement)
Names = c("Low", "Medium", "High", "Very High")
colnames(plot_involvement) <- Names
prop.table(plot_involvement, 2)
```

```
##
##              Low     Medium       High  Very High
##   No  0.66265060 0.81066667 0.85599078 0.90972222
##   Yes 0.33734940 0.18933333 0.14400922 0.09027778
```

```r
#
plot_joblevel = table(data$Attrition, data$JobLevel)
Names = c("Lowest", "Low", "Medium", "High", "Highest")
colnames(plot_joblevel) <- Names
prop.table(plot_joblevel, 2)
```

```
##
##           Lowest        Low     Medium       High    Highest
##   No  0.73664825 0.90262172 0.85321101 0.95283019 0.92753623
##   Yes 0.26335175 0.09737828 0.14678899 0.04716981 0.07246377
```

```r
#
plot_jobrole = table(data$Attrition, data$JobRole)
prop.table(plot_jobrole, 2)
```

```
##
##       Healthcare Representative Human Resources Laboratory Technician
##   No                 0.93129771      0.76923077            0.76061776
##   Yes                0.06870229      0.23076923            0.23938224
##
##          Manager Manufacturing Director Research Director Research Scientist
##   No  0.95098039             0.93103448        0.97500000         0.83904110
##   Yes 0.04901961             0.06896552        0.02500000         0.16095890
##
##       Sales Executive Sales Representative
##   No       0.82515337           0.60240964
##   Yes      0.17484663           0.39759036
```

```r
#
plot_jobsat = table(data$Attrition, data$JobSatisfaction)
Names = c("Low", "Medium", "High", "Very High")
colnames(plot_jobsat) <- Names
prop.table(plot_jobsat, 2)
```

```
##
##             Low    Medium      High Very High
##   No  0.7716263 0.8357143 0.8348416 0.8867102
##   Yes 0.2283737 0.1642857 0.1651584 0.1132898
```

```r
#
plot_maritial = table(data$Attrition, data$MaritalStatus)
prop.table(plot_maritial, 2)
```

```
##
##        Divorced   Married    Single
##   No  0.8990826 0.8751857 0.7446809
##   Yes 0.1009174 0.1248143 0.2553191
```

```r
#
plot_overtime = table(data$Attrition, data$OverTime)
prop.table(plot_overtime, 2)
```

```
##
##              No       Yes
##   No  0.8956357 0.6947115
##   Yes 0.1043643 0.3052885
```

```r
#
plot_rating = table(data$Attrition, data$PerformanceRating)
Names = c("Excellent", "Outstanding")
colnames(plot_rating) <- Names
prop.table(plot_rating, 2)
```

```
##
##       Excellent Outstanding
##   No  0.8392283   0.8362832
##   Yes 0.1607717   0.1637168
```

```r
#
plot_relsat = table(data$Attrition, data$RelationshipSatisfaction)
Names = c("Low", "Medium", "High", "Very High")
colnames(plot_relsat) <- Names
prop.table(plot_relsat, 2)
```

```
##
##             Low    Medium      High Very High
##   No  0.7934783 0.8514851 0.8453159 0.8518519
##   Yes 0.2065217 0.1485149 0.1546841 0.1481481
```

```r
#
plot_options = table(data$Attrition, data$StockOptionLevel)
Names = c("None", "Low", "Medium", "High")
colnames(plot_options) <- Names
prop.table(plot_options, 2)
```

```
##
##             None        Low     Medium       High
##   No  0.75594295 0.90604027 0.92405063 0.82352941
##   Yes 0.24405705 0.09395973 0.07594937 0.17647059
```

```r
#
plot_worklife = table(data$Attrition, data$WorkLifeBalance)
Names = c("Low", "Medium", "High", "Very High")
colnames(plot_worklife) <- Names
prop.table(plot_worklife, 2)
```

```
##
##             Low    Medium      High Very High
##   No  0.6875000 0.8313953 0.8577828 0.8235294
##   Yes 0.3125000 0.1686047 0.1422172 0.1764706
```

```
#
```

Results

Age - young employees have a higher attrition rate

Business Travel - employees who travel frequently have highe

Daily rate - no significant difference

Department - Sales and Human Resources have higher attrition rates than R&D

Distance from Home - employees who commute farther have higher attrition

Education - employees with lower education (no college, college, etc.) are somewhat more likely to quit

Education Field - there are differences by field, people in Sales and HR most likley to quit

Employee Count - N/A (variable will be removed)

Employee Number - N/A (variable will be removed)

Environment Satisfaction - employees with low environment satisfaction much more likley to quit

Gender - male and femal employees quit at similar rates

Hourly rate - no significant difference

Job Involvement - employees with low job involvement much more likly to quit

Job Level - employees at low levels more likley to quit

Job Role - employees in Sales and Human Resources more likly to quit

Job satisfaction - employees with low job satisfaction levels more likley to quit

Maritial Status - single employees more liklely to quit

Monthly Income - employees with lower incomes more likly to quit

Monthly rate - no significant difference

Number of companies worked - employyes who have worked for few companies more likley to leave

Over 18 - N/A (variable will be removed)

Overtime - employees who get overtime much more likley to quit

Percent salary hike - no significant differenc between employees who stay or leave

Performance rating - no significant difference between emploees who stay and leave.

Relationship Satisfaction - employees with very low relationship satisfaction more likley to quit

Standard hours - N/A (variable will be removed)

Stock Options - employees with no options more likley to quit

Total working years - employees with fewer working years more likley to quit

Training times - people with very little training more likley to quit

Worklife balance - employees with low worklife balance much more likley to quit

Years at company - employees with fewer years more likley to quit

Years in current role - employyes most likley to leave in first few years in role

Years since last promotion - no significant difference between employees who stay or leave

Years with current manager - employees most likley to quit first year

Correlation

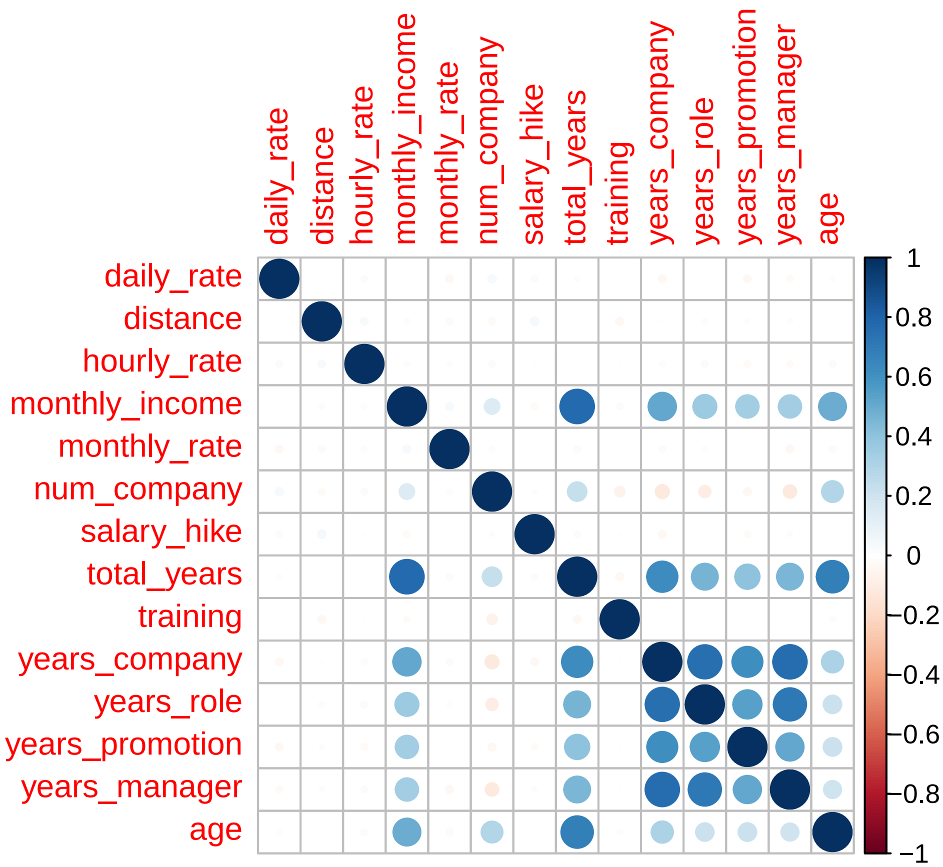
analyze correlation of numerical variables

```
# combine numeric variables into a new dataframe
df1 <- cbind(data$DailyRate, data$DistanceFromHome, data$HourlyRate, data$MonthlyIncome, data$MonthlyRat
dfnum <- data.frame(df1)
names <- c("daily_rate", "distance", "hourly_rate", "monthly_income", "monthly_rate", "num_company", "sa
colnames(dfnum) <- names
# calculate correlation
dfnum.cor = cor(dfnum)
# install .packages("corrplot")
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.6.3
```

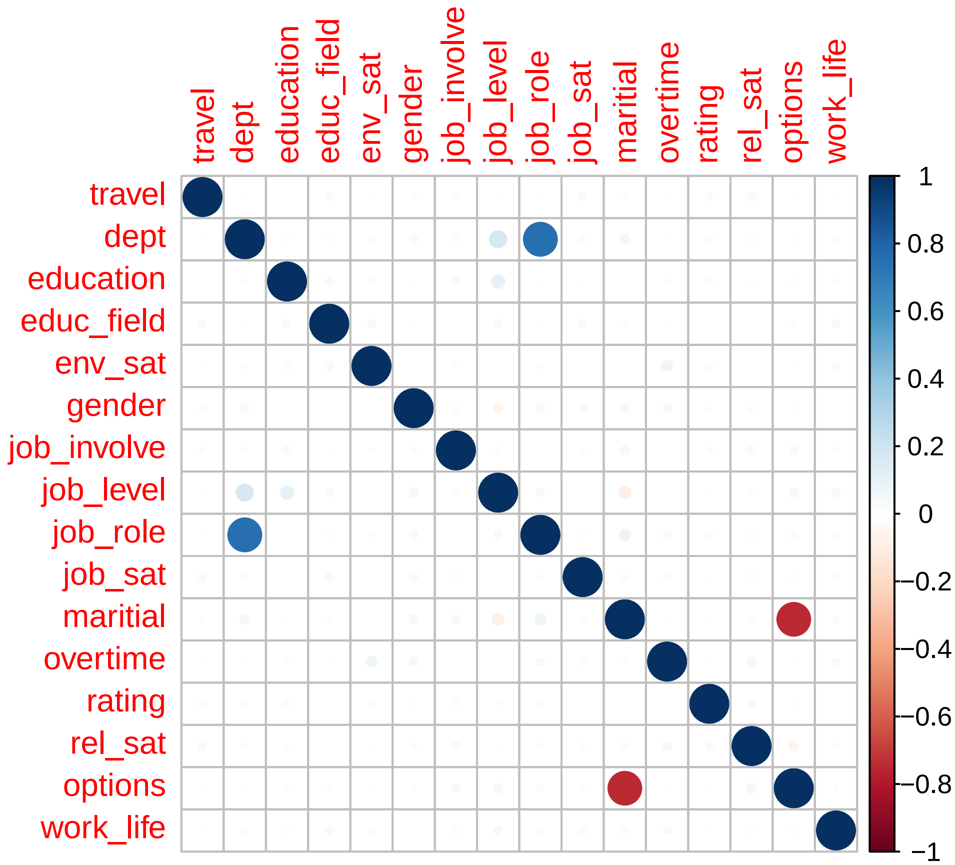
```
## corrplot 0.84 loaded
```

```
corrplot(dfnum.cor)
```




```
# combine categorical varibles into a new dataframe
df2 <- cbind(data$BusinessTravel, data$Department, data$Education, data$EducationField, data$Environment
names1 <- c("travel", "dept", "education", "educ_field", "env_sat", "gender", "job_involve", "job_level
#
```

```r
dfcat <- data.frame(df2)
colnames(dfcat) <- names1
# calculate correlation
dfcat.cor = cor(dfcat, method = "spearman")
library(corrplot)
corrplot(dfcat.cor)
```



Results

An analysis of corr and corrplot shows that a number of variables seem to be strongly correlated:

Numerical Variables - monthly income vs total working years (corr = 0.77) - age vs total working years(corr = 0.68) - years at company vs total working years (corr =0.63) - years at company vs years in current role (corr = 0.76) - years with current manager vs years at company (corr =0.77) - years with current manager vs years in current role (corr=0.77) - years at company vs years since last promotion (corr =0.62)

Categorical Variables - job role vs dept (corr = 0.66) - maritial status vs stock option level (corr = 0.75)

Data Preparation

```r
# remove 4 redundant variables identified as redundant in EDA
att <- data[,-c(8,9,21,26)]
```

Multi-Collinearity

```
#convert factors to numeric
att$Leave <- ifelse(att$Attrition == "Yes",1,0)
att$BusinessTravel <-as.numeric(att$BusinessTravel)
att$Department <- as.numeric(att$Department)
att$EducationField <- as.numeric(att$EducationField)
att$Gender <- as.numeric(att$Gender)
att$JobRole <- as.numeric(att$JobRole)
att$MaritalStatus <-as.numeric(att$MaritalStatus)
att$OverTime <-as.numeric(att$OverTime)
```

Test for multi-collinearity using Farrar-Gauber Test

```
# install.packages("mctest")
library(mctest)
Y <- att$Leave
X <-att[,-c(1,32)]
omcdiag(X,Y)
```

```
##
## Call:
## omcdiag(x = X, y = Y)
##
##
## Overall Multicollinearity Diagnostics
##
##                        MC Results detection
## Determinant |X'X|:         0.0001         1
## Farrar Chi-Square:     13932.0467         1
## Red Indicator:             0.1548         0
## Sum of Lambda Inverse:    70.6887         0
## Theil's Method:            2.9790         1
## Condition Number:         99.8255         1
##
## 1 --> COLLINEARITY is detected by the test
## 0 --> COLLINEARITY is not detected by the test
```

Test original dataset variables for multi-collinearity

```
df <- att[,-1]
output <- lm(Leave ~., data = df)
# install.packages("car")
library(car)
```

```
## Warning: package 'car' was built under R version 3.6.3
```

```
## Loading required package: carData
```

```r
car::vif(output)
```

```
##          BusinessTravel                DailyRate               Department
##                1.016413                 1.023990                 1.942150
##        DistanceFromHome                Education           EducationField
##                1.017135                 1.063531                 1.016236
##  EnvironmentSatisfaction                   Gender               HourlyRate
##                1.017516                 1.019383                 1.021142
##          JobInvolvement                 JobLevel                  JobRole
##                1.020804                11.821396                 1.894260
##         JobSatisfaction            MaritalStatus            MonthlyIncome
##                1.020727                 1.840999                11.052627
##             MonthlyRate       NumCompaniesWorked                 OverTime
##                1.015602                 1.261957                 1.028587
##       PercentSalaryHike        PerformanceRating RelationshipSatisfaction
##                2.521576                 2.519366                 1.020852
##        StockOptionLevel        TotalWorkingYears    TrainingTimesLastYear
##                1.819542                 4.824448                 1.023713
##         WorkLifeBalance           YearsAtCompany       YearsInCurrentRole
##                1.018516                 4.601972                 2.728267
##  YearsSinceLastPromotion     YearsWithCurrManager                      Age
##                1.678879                 2.782899                 2.054172
```

Revise dataset to exlude highly correlated variables, by dropping variables and reviewing vif results This was done multiple times

```r
df1 <- df[,-c(15, 20, 23, 26, 28, 29)]
output <- lm(Leave ~., data = df1)
library(car)
car::vif(output)
```

```
##           BusinessTravel                DailyRate               Department
##                 1.011084                 1.016497                 1.887442
##         DistanceFromHome                Education           EducationField
##                 1.012099                 1.061722                 1.014760
##  EnvironmentSatisfaction                   Gender               HourlyRate
##                 1.015603                 1.018244                 1.019271
##           JobInvolvement                 JobLevel                  JobRole
##                 1.014094                 1.601555                 1.879019
##          JobSatisfaction            MaritalStatus              MonthlyRate
##                 1.017694                 1.837748                 1.012114
##       NumCompaniesWorked                 OverTime        PercentSalaryHike
##                 1.157171                 1.026056                 1.009521
## RelationshipSatisfaction         StockOptionLevel    TrainingTimesLastYear
##                 1.016740                 1.813892                 1.022258
##          WorkLifeBalance       YearsInCurrentRole                      Age
##                 1.014024                 1.229205                 1.531951
```

Results - four variables identified as redundant in EDA removed - six variables removed due to multi-collinearity analysis - Final analysis of vif shows multi-collinearity greatly reduced (max value < 2)

Variables removed during Multi-collinearity: - monthly income - performance Rating - total working years - years at company - years with current manager - years since last promotion

Imbalanced Data

Analyze the dataset to determine if the data is unbalanced this will be done by analyzing the original dataset vs a revised dataset and comparing the resultsb the dataset will be revised using SMOTE

Prepare dataset

```
mydata <- data[,-c(8,9,21,26)] # remove EDA redundant variables
mydata$Leave <- ifelse(mydata$Attrition == "Yes",1,0)
mydata1 <- mydata[,-c(16,21,24,27,29, 30)] #remove multi-collinearity variables
mydata2 <- mydata1
mydata2$Education <- as.factor(mydata2$Education)
mydata2$EnvironmentSatisfaction <- as.factor(mydata2$EnvironmentSatisfaction)
mydata2$JobInvolvement <- as.factor(mydata2$JobInvolvement)
mydata2$JobLevel <- as.factor(mydata2$JobLevel)
mydata2$JobSatisfaction <- as.factor(mydata2$JobSatisfaction)
mydata2$RelationshipSatisfaction <- as.factor(mydata2$RelationshipSatisfaction)
mydata2$StockOptionLevel <- as.factor(mydata2$StockOptionLevel)
mydata2$WorkLifeBalance <- as.factor(mydata2$WorkLifeBalance)
mydata3 <- mydata2
```

Baseline Model

```
set.seed(2020)
mydata_GLM <- mydata2[,-1]
train_index <- sample(1:nrow(mydata_GLM), .7*nrow(mydata_GLM))
traindata <- mydata_GLM[train_index,]
testdata <- mydata_GLM[-train_index,]
model_GLM <- glm(Leave ~. , family = "binomial", data = traindata)
```

Baseline Confusion Matrix

```
pred_logistic <- predict(model_GLM, type = "response", newdata = testdata)
table(testdata$Leave, pred_logistic > .5)
```

```
##
##     FALSE TRUE
##   0   352   19
##   1    43   27
```

Baseline AUC

```
# install.packages("ROCR")
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.6.3
```

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

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

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

```
glm_ROC <- pred_logistic
pred_GLM <- prediction(glm_ROC, testdata$Leave)
auc_GLM <- performance(pred_GLM, "auc")
auc_GLM <- round(as.numeric(auc_GLM@y.values),2)
auc_GLM
```

```
## [1] 0.83
```

Adjust data using SMOTE

```
#install.packages("DMwR")
library(DMwR)
```

```
## Warning: package 'DMwR' was built under R version 3.6.3
```

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

```
## Loading required package: grid
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
set.seed(2020)
train_index <- sample(1:nrow(mydata3), .7*nrow(mydata3))
traindata <- mydata3[train_index,]
testdata <- mydata3[-train_index,]
SMOTE_data <- SMOTE(Attrition ~ ., traindata, perc.over = 200, k=5, perc.under=300)
summary(SMOTE_data$Attrition)
```

```
##   No  Yes
## 1002  501
```

Rerun the model with revised data set This was done several times SMOTE and the results were compared using AUC and confusion matrix

```
set.seed(2020)
mydata_GLM <- SMOTE_data[,-1]
train_index <- sample(1:nrow(mydata_GLM), .7*nrow(mydata_GLM))
traindata <- mydata_GLM[train_index,]
testdata <- mydata_GLM[-train_index,]
model_GLM <- glm(Leave ~. , family = "binomial", data = traindata)
```

Revised confusion matrix

```
pred_logistic <- predict(model_GLM, type = "response", newdata = testdata)
table(testdata$Leave, pred_logistic > .5)
```

```
##
##     FALSE TRUE
##   0   276   24
##   1    37  114
```

Revised AUC

```
# install.packages("ROCR")
library(ROCR)
glm_ROC <- pred_logistic
pred_GLM <- prediction(glm_ROC, testdata$Leave)
auc_GLM <- performance(pred_GLM, "auc")
auc_GLM <- round(as.numeric(auc_GLM@y.values),2)
auc_GLM
```

```
## [1] 0.9
```

Results

Revised dataset significantly improves the results: TP: baseline model 56%, revised model 82% FP: baseline mode 44%, revised model 18% TN: baseline model 89%, revised model 89% FN: baseline model 11%, revised model 1% AUC: baseline model 0.83, revised model 0.90

Models

Logistic reqression

```
set.seed(2020)
mydata_GLM <- SMOTE_data[,-1]
train_index <- sample(1:nrow(mydata_GLM), .7*nrow(mydata_GLM))
traindata <- mydata_GLM[train_index,]
testdata <- mydata_GLM[-train_index,]
model_GLM <- glm(Leave ~. , family = "binomial", data = traindata)
```

Confusion Matrix

```
pred_logistic <- predict(model_GLM, type = "response", newdata = testdata)
table(testdata$Leave, pred_logistic > .5)
```

```
##
##     FALSE TRUE
##   0   276   24
##   1    37  114
```

AUC

```r
# install.packages("ROCR")
library(ROCR)
glm_ROC <- pred_logistic
pred_GLM <- prediction(glm_ROC, testdata$Leave)
auc_GLM <- performance(pred_GLM, "auc")
auc_GLM <- round(as.numeric(auc_GLM@y.values),2)
auc_GLM
```

```
## [1] 0.9
```

Decision Tree

```r
set.seed(2020)
mydata_tree <- SMOTE_data [,-26]
train_index <- sample(1:nrow(mydata_tree), .7*nrow(mydata_tree))
traindata <- mydata_tree[train_index,]
testdata <- mydata_tree[-train_index,]
# install.packages("rpart")
library(rpart)
model_tree <- rpart(Attrition ~., data = traindata, method="class")
```

Confusion Matrix

```r
pred_tree <- predict(model_tree, type = "class", newdata = testdata)
table(testdata$Attrition, pred_tree)
```

```
##      pred_tree
##         No Yes
##   No   267  33
##   Yes   60  91
```

AUC

```r
# install.packages("ROCR")
library(ROCR)
DT_ROC <- predict(model_tree, testdata)
pred_DT <- prediction(DT_ROC[,2], testdata$Attrition)
auc_tree <- performance(pred_DT, "auc")
auc_tree <- round(as.numeric(auc_tree@y.values),2)
auc_tree
```

```
## [1] 0.82
```

Random Forest

```r
set.seed(2020)
mydata_RF <- SMOTE_data [,-26]
train_index <- sample(1:nrow(mydata_RF), .7*nrow(mydata_RF))
traindata <- mydata_RF[train_index,]
testdata <- mydata_RF[-train_index,]
# install.packages("randomForest")
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 3.6.3
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```r
model_RF <- randomForest(Attrition ~., data = traindata)
```

Confusion Matrix

```r
predict_RF <- predict(model_RF, newdata = testdata)
table(testdata$Attrition, predict_RF)
```

```
##      predict_RF
##        No Yes
##   No  292   8
##   Yes  43 108
```

AUC

```r
# install.packages("ROCR")
library(ROCR)
RF_ROC <- predict(model_RF, testdata, type="prob")
pred_RF <- prediction(RF_ROC[,2], testdata$Attrition)
auc_RF <- performance(pred_RF, "auc")
auc_RF <- round(as.numeric(auc_RF@y.values),2)
auc_RF
```

```
## [1] 0.97
```

Initial Results Logistic Regresion: accuracy = 87%, AUC = .90 Decision Tree: accuracy = 79%, AUC =0.82 Random Forest: accuracy = 89%, AUC =.97

Random Forest scores the best on both accuracy and AUC

Random Forest Confusion Matrix TN = 87% FN = 13% TP = 93% TP = 5% Accuracy = 90%

Feature Selection

Analyze the Random Forest model to determine if there are any variables which are not significant This will be done using the Boruta feature selection package

Boruta Feature Selection

```r
set.seed(2020)
mydata_RF <- SMOTE_data [,-26]
train_index <- sample(1:nrow(mydata_RF), .7*nrow(mydata_RF))
traindata <- mydata_RF[train_index,]
testdata <- mydata_RF[-train_index,]
# install.packages("Boruta")
library(Boruta)
```

```
## Warning: package 'Boruta' was built under R version 3.6.3

## Loading required package: ranger

## Warning: package 'ranger' was built under R version 3.6.3

##
## Attaching package: 'ranger'

## The following object is masked from 'package:randomForest':
##
##     importance
```

```r
Boruta_output <- Boruta(Attrition ~ ., data = traindata, pValue = .01, doTrace=0)
```

Boruta Significant values (incuding tentatives)

```r
boruta_signif <- getSelectedAttributes(Boruta_output, withTentative = TRUE)
print(boruta_signif)
```

```
##  [1] "BusinessTravel"           "DailyRate"
##  [3] "Department"               "DistanceFromHome"
##  [5] "Education"                "EducationField"
##  [7] "EnvironmentSatisfaction"  "Gender"
##  [9] "HourlyRate"               "JobInvolvement"
## [11] "JobLevel"                 "JobRole"
## [13] "JobSatisfaction"          "MaritalStatus"
## [15] "MonthlyRate"              "NumCompaniesWorked"
## [17] "OverTime"                 "PercentSalaryHike"
## [19] "RelationshipSatisfaction" "StockOptionLevel"
## [21] "TrainingTimesLastYear"    "WorkLifeBalance"
## [23] "YearsInCurrentRole"       "Age"
```

Boruta tentative variables

```r
roughFixMod <- TentativeRoughFix(Boruta_output)
```

```
## Warning in TentativeRoughFix(Boruta_output): There are no Tentative attributes!
## Returning original object.
```

```
boruta_signif <- getSelectedAttributes(roughFixMod)
print(boruta_signif)
```

```
##  [1] "BusinessTravel"           "DailyRate"
##  [3] "Department"               "DistanceFromHome"
##  [5] "Education"                "EducationField"
##  [7] "EnvironmentSatisfaction"  "Gender"
##  [9] "HourlyRate"               "JobInvolvement"
## [11] "JobLevel"                 "JobRole"
## [13] "JobSatisfaction"          "MaritalStatus"
## [15] "MonthlyRate"              "NumCompaniesWorked"
## [17] "OverTime"                 "PercentSalaryHike"
## [19] "RelationshipSatisfaction" "StockOptionLevel"
## [21] "TrainingTimesLastYear"    "WorkLifeBalance"
## [23] "YearsInCurrentRole"       "Age"
```

Results - the Boruta regression indicated that all the variables in the dataset were statistically significant

Results

Best Model

```
set.seed(2020)
mydata_RF <- SMOTE_data [,-26]
train_index <- sample(1:nrow(mydata_RF), .7*nrow(mydata_RF))
traindata <- mydata_RF[train_index,]
testdata <- mydata_RF[-train_index,]
# install.packages("randomForest")
library(randomForest)
model_RF <- randomForest(Attrition ~., data = traindata)
```

Confusion Matrix

```
predict_RF <- predict(model_RF, newdata = testdata)
table(testdata$Attrition, predict_RF)
```

```
##      predict_RF
##        No Yes
##   No  292   8
##   Yes  43 108
```

Random Forest Confusion Matrix TN = 87% FN = 13% TP = 93% TP = 5% Accuracy = 90%

ROC Curve

```
# install.packages("ROCR")
library(ROCR)
RF_ROC <- predict(model_RF, testdata, type="prob")
pred_RF <- prediction(RF_ROC[,2], testdata$Attrition)
```

```r
auc_RF <- performance(pred_RF, "auc")
auc_RF <- round(as.numeric(auc_RF@y.values),2)
auc_RF
```

```
## [1] 0.97
```

An AUC value of 0.97 indicates that the model is highly predictive

Variable Importance

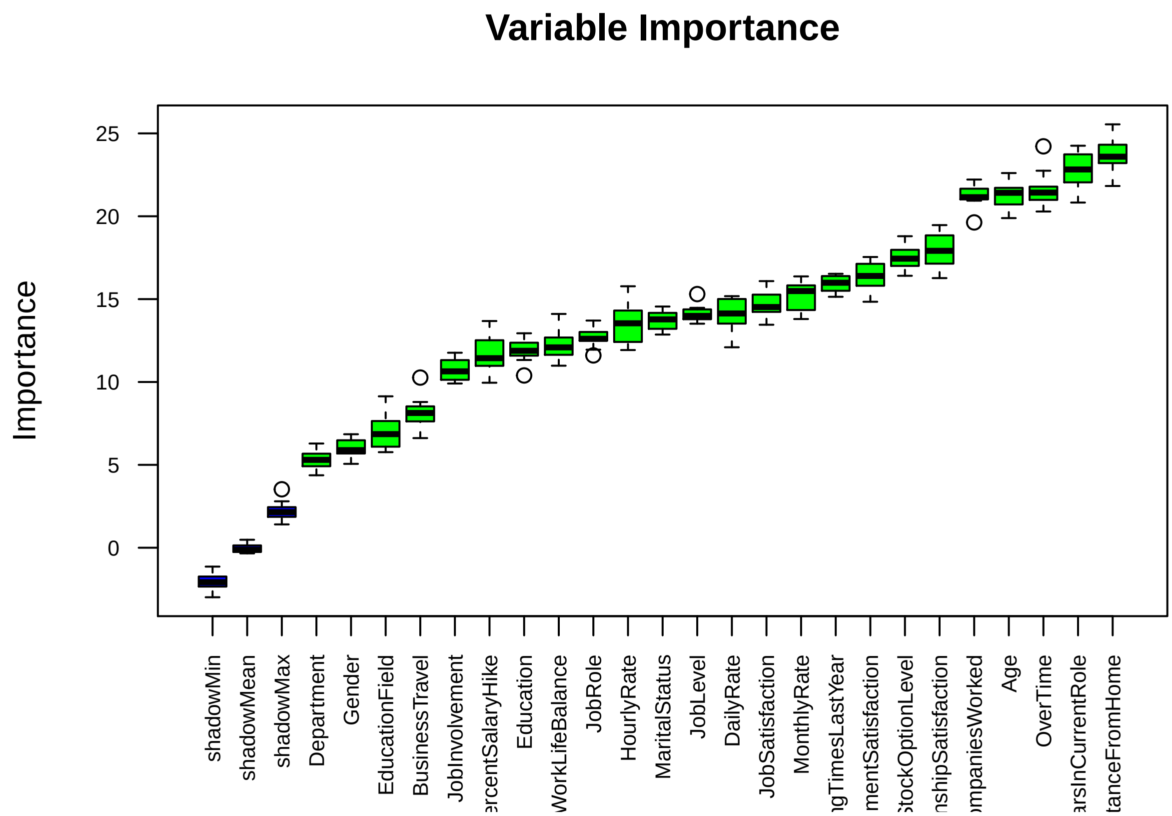
Determine the most important variables using the Boruta package

```r
# install.packages("Boruta")
library(Boruta)
imps <- attStats(roughFixMod)
imps2 <- imps[imps$decision !='Rejected', c('meanImp', 'decision')]
top <- (imps2[order(-imps2$meanImp),])
top
```

```
##                            meanImp  decision
## DistanceFromHome         23.748830 Confirmed
## YearsInCurrentRole       22.782583 Confirmed
## OverTime                 21.596411 Confirmed
## Age                      21.269503 Confirmed
## NumCompaniesWorked       21.241044 Confirmed
## RelationshipSatisfaction 17.907873 Confirmed
## StockOptionLevel         17.509776 Confirmed
## EnvironmentSatisfaction  16.419917 Confirmed
## TrainingTimesLastYear    15.914242 Confirmed
## MonthlyRate              15.186912 Confirmed
## JobSatisfaction          14.694789 Confirmed
## JobLevel                 14.120633 Confirmed
## DailyRate                14.050833 Confirmed
## MaritalStatus            13.718151 Confirmed
## HourlyRate               13.540950 Confirmed
## JobRole                  12.661747 Confirmed
## WorkLifeBalance          12.240985 Confirmed
## Education                11.902157 Confirmed
## PercentSalaryHike        11.689682 Confirmed
## JobInvolvement           10.728643 Confirmed
## BusinessTravel            8.154510 Confirmed
## EducationField            7.098902 Confirmed
## Gender                    6.000160 Confirmed
## Department                5.303465 Confirmed
```

Plot variable importance

```r
plot(Boruta_output, cex.axis=.7, las=2, xlab="", main="Variable Importance")
```



Top Variables

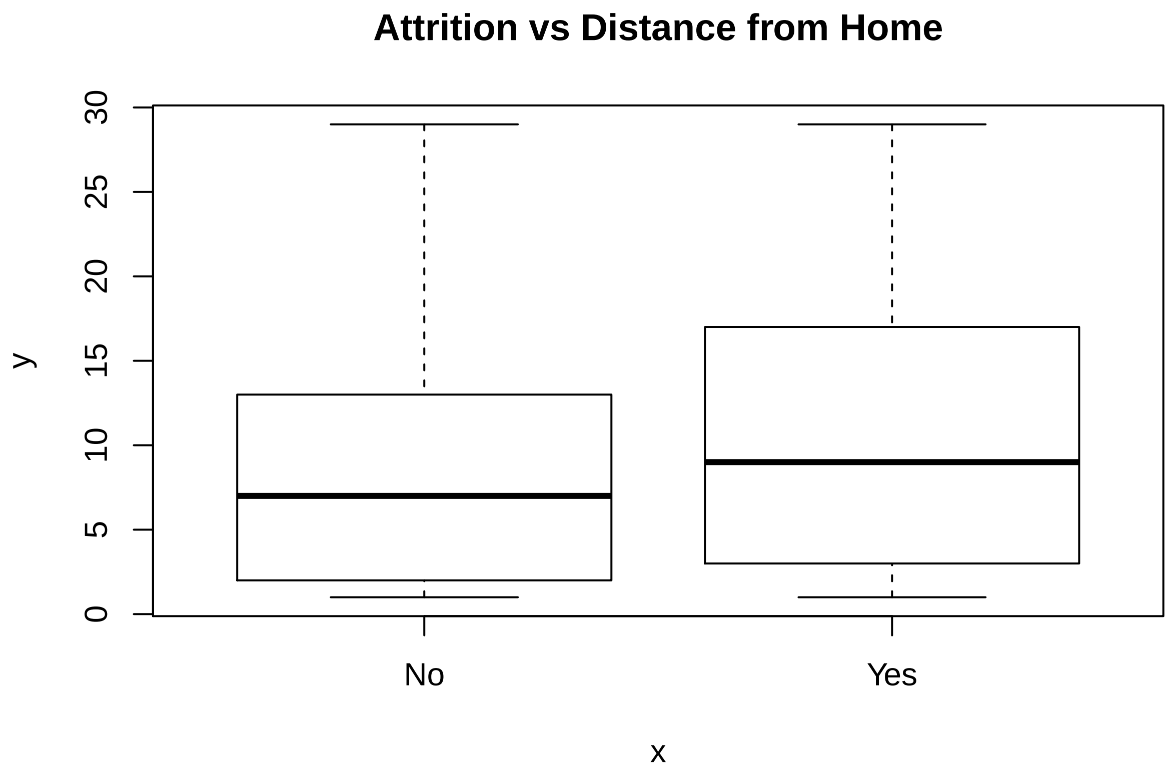
Display the top 8 variables

```
top[1:8,]
```

```
##                          meanImp  decision
## DistanceFromHome         23.74883 Confirmed
## YearsInCurrentRole       22.78258 Confirmed
## OverTime                 21.59641 Confirmed
## Age                      21.26950 Confirmed
## NumCompaniesWorked       21.24104 Confirmed
## RelationshipSatisfaction 17.90787 Confirmed
## StockOptionLevel         17.50978 Confirmed
## EnvironmentSatisfaction  16.41992 Confirmed
```

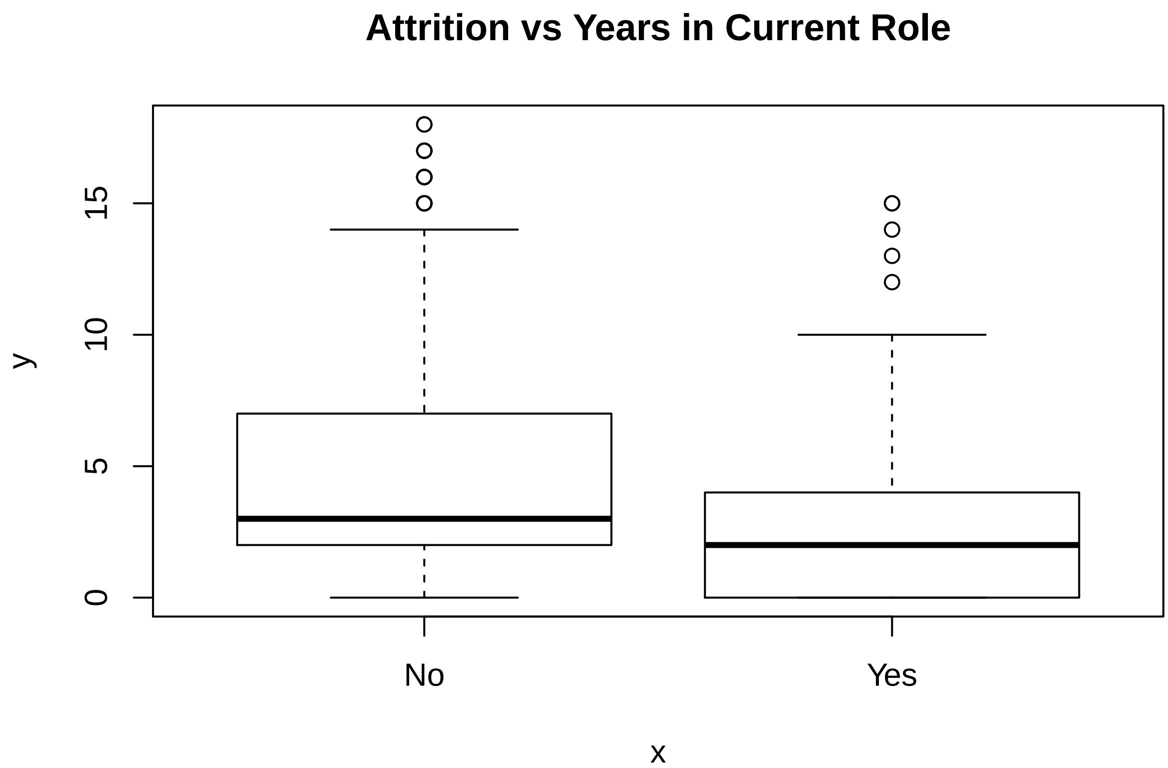
Plot Distance from Home vs Attrition

```
plot(x=data$Attrition, y=data$DistanceFromHome, main ="Attrition vs Distance from Home")
```



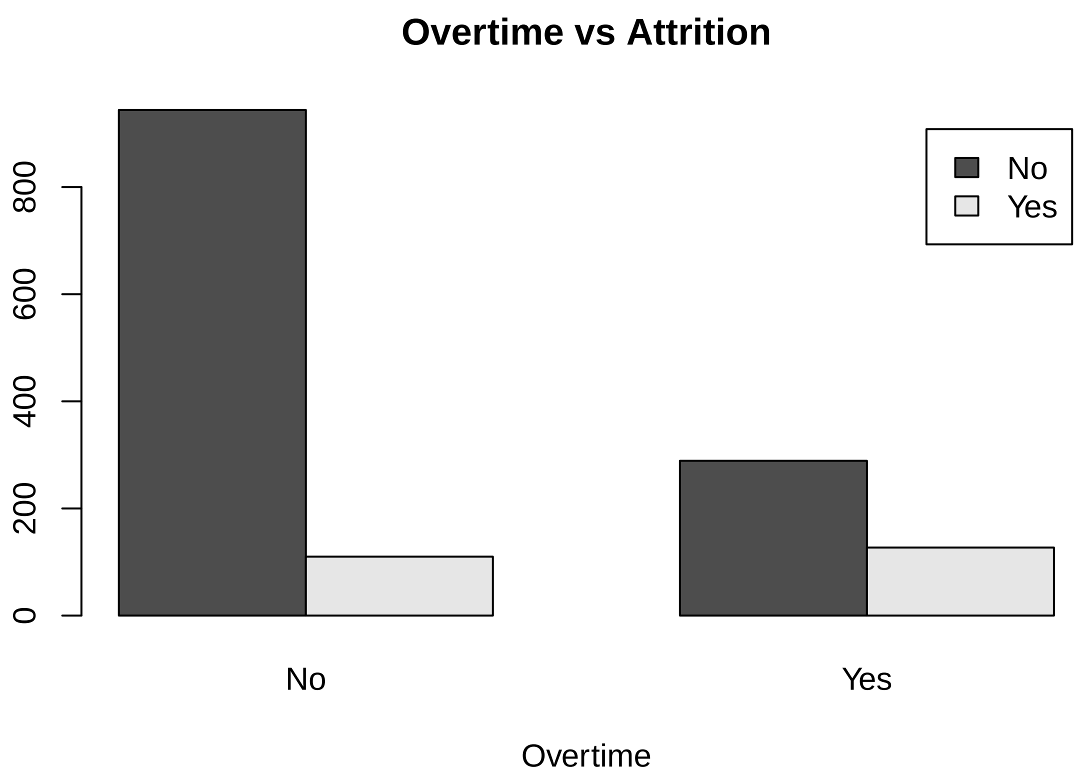
Plot Years in Current Role vs Attrition

```
plot(x=data$Attrition, y=data$YearsInCurrentRole, main="Attrition vs Years in Current Role")
```



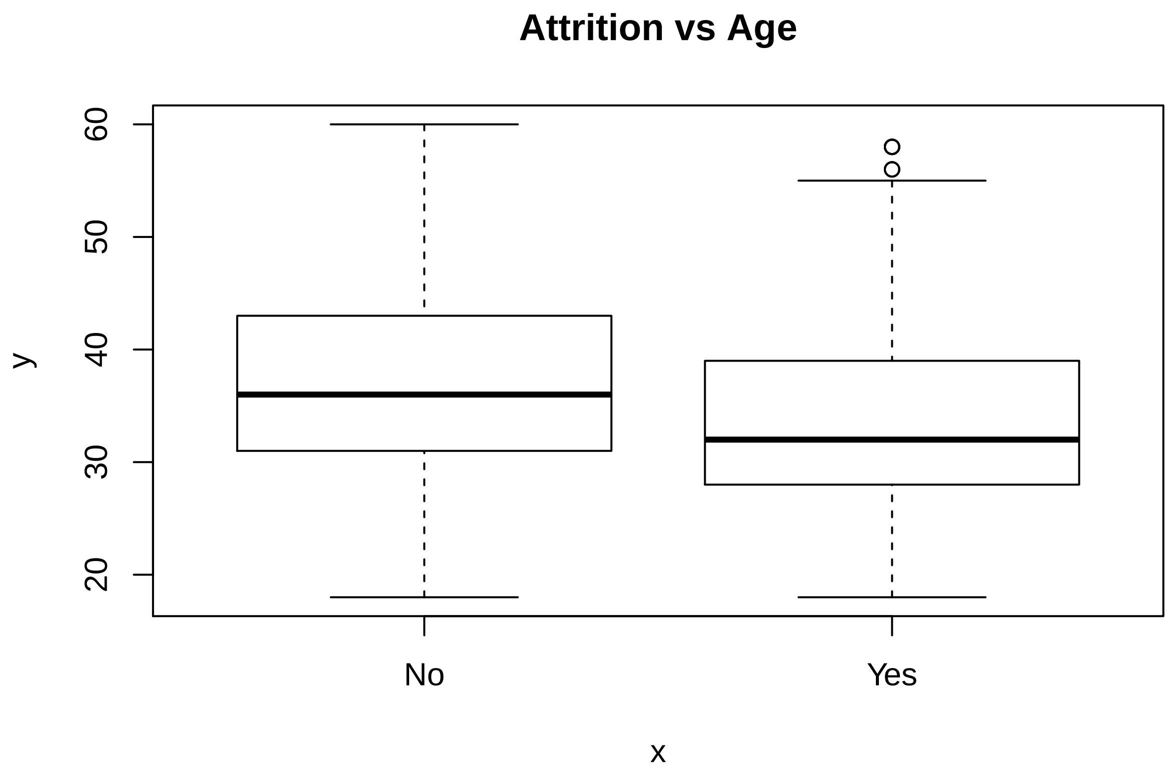
Plot Overtime vs Attrition

```
plot_overtime = table(data$Attrition, data$OverTime)
barplot(plot_overtime, main="Overtime vs Attrition", xlab="Overtime", legend=rownames(plot_overtime), be
```



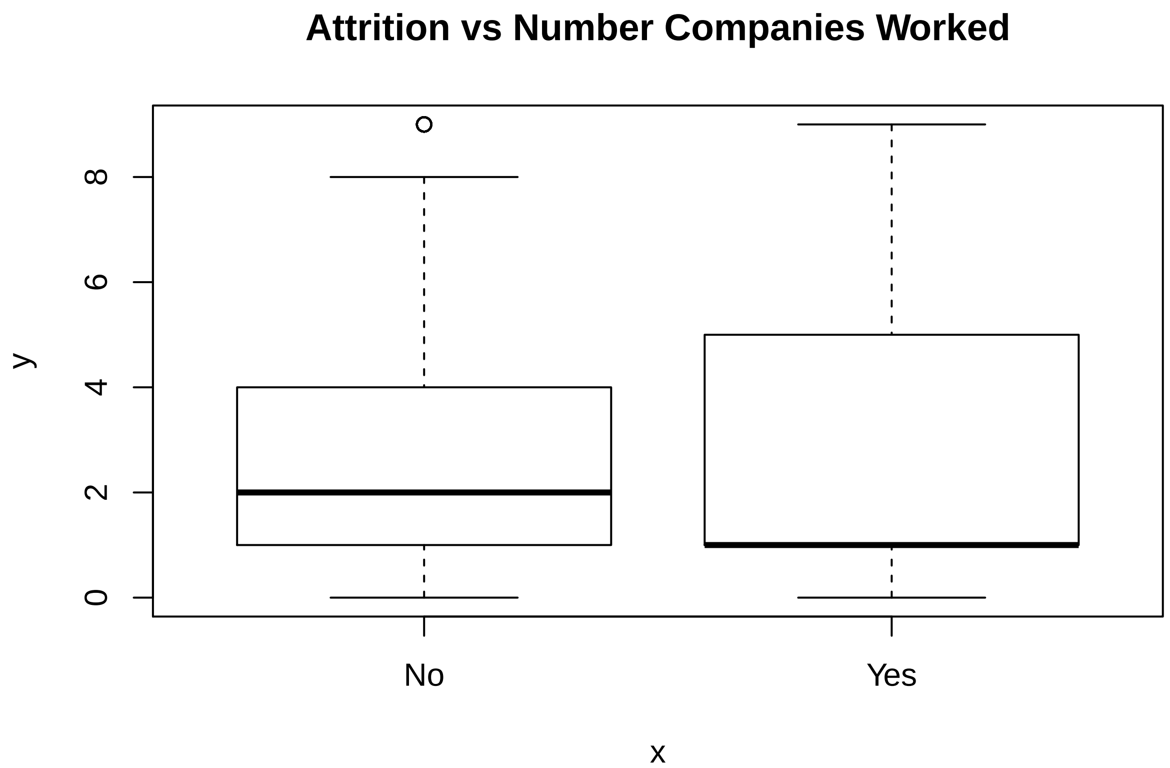
Plot Age vs Attrition

```
plot(x=data$Attrition, y=data$Age, main="Attrition vs Age")
```



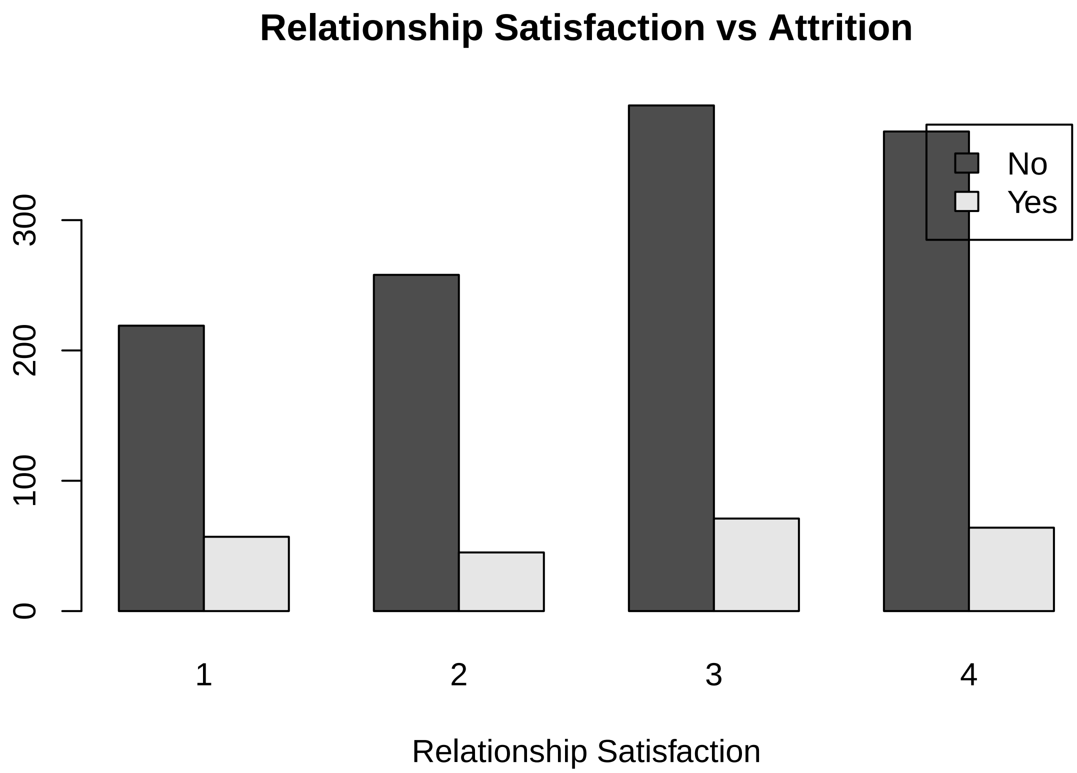
Plot Number of Companies worked vs Attrition

```
plot(x=data$Attrition, y=data$NumCompaniesWorked, main="Attrition vs Number Companies Worked")
```



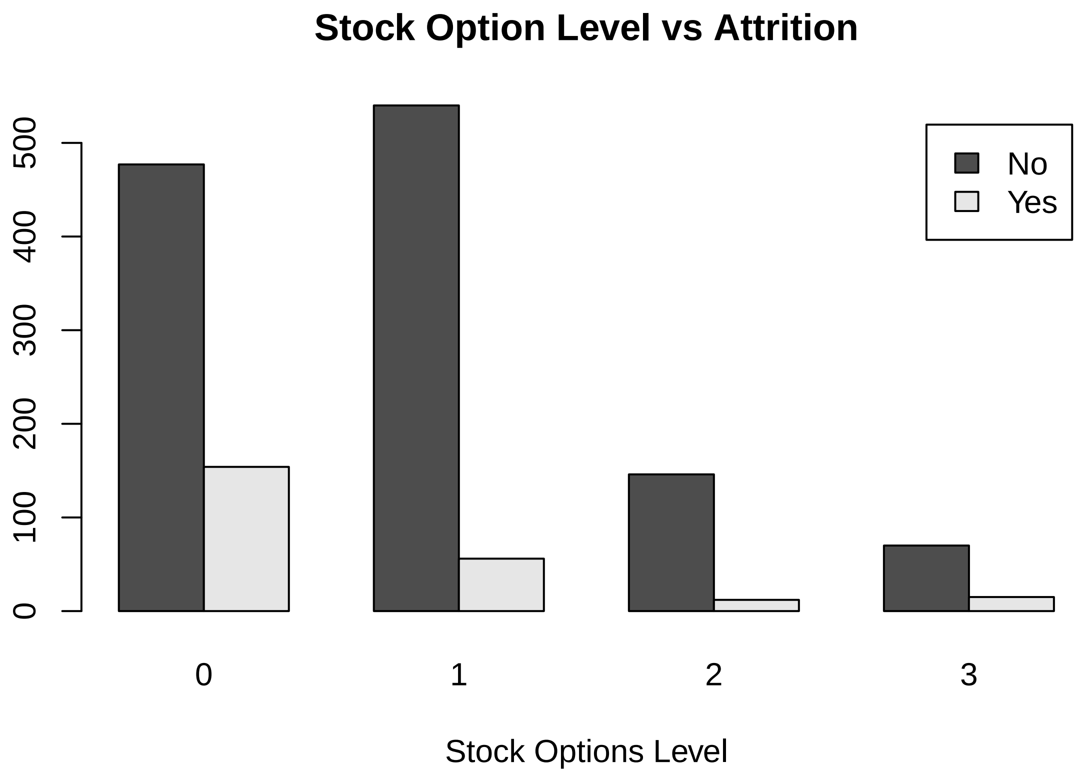
Plot Relationship Satisfaction vs Attrition

```
plot_relsat = table(data$Attrition, data$RelationshipSatisfaction)
barplot(plot_relsat, main="Relationship Satisfaction vs Attrition", xlab="Relationship Satisfaction", le
```



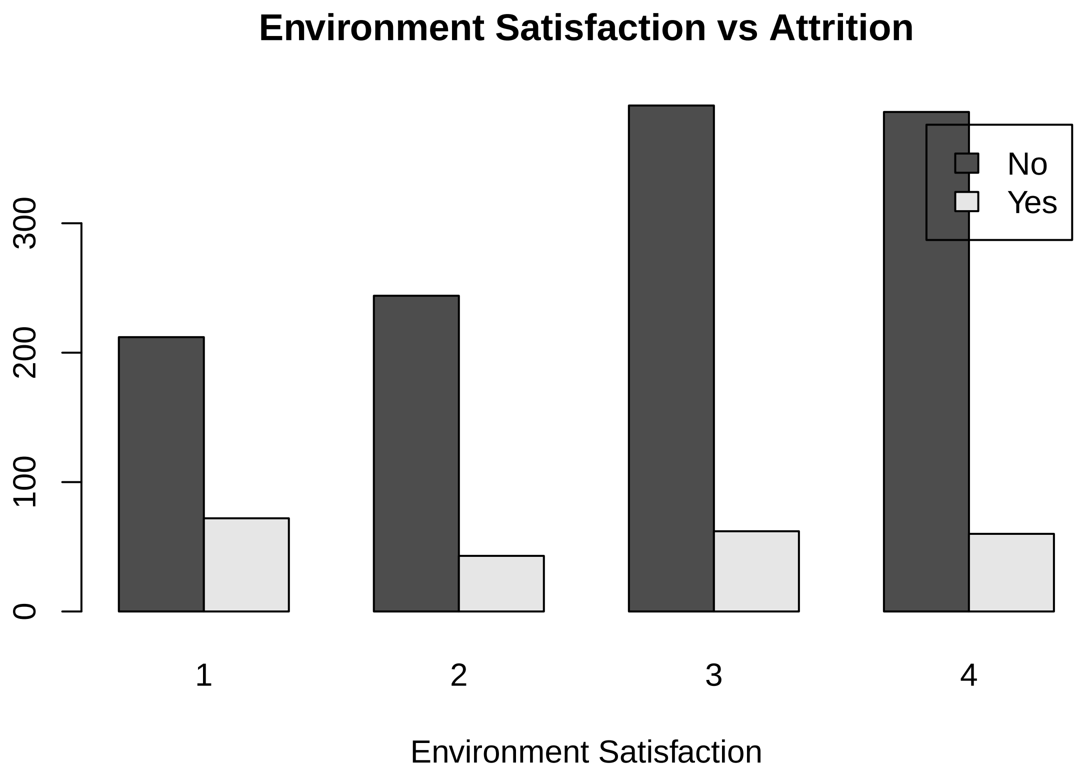
Plot Stock Options Level vs Attrition

```
plot_options = table(data$Attrition, data$StockOptionLevel)
barplot(plot_options, main="Stock Option Level vs Attrition", xlab="Stock Options Level", legend=rowname
```



Plot Environment Satisfaction vs Attrition

```
plot_envsat = table(data$Attrition, data$EnvironmentSatisfaction)
barplot(plot_envsat, main="Environment Satisfaction vs Attrition", xlab="Environment Satisfaction", lege
```



Scenarios

```
set.seed(2020)
mydata_RF <- SMOTE_data [,-26]
train_index <- sample(1:nrow(mydata_RF), .7*nrow(mydata_RF))
traindata <- mydata_RF[train_index,]
testdata <- mydata_RF[-train_index,]
# install.packages("randomForest")
library(randomForest)
model_RF <- randomForest(Attrition ~., data = traindata, ntree = 100)
```

What is the predicted attrition rate for employees < 30 years old

```
set.seed(2020)
testdata_rev <- subset(testdata, testdata$Age <= 30,)
predict_RF <- predict(model_RF, newdata = testdata_rev)
table(testdata_rev$Attrition, predict_RF)
```

```
##      predict_RF
##        No Yes
##   No   69   1
##   Yes   9  28
```

Result - predicted attrition rate is 26%

What is the predicted attrition rate for employees less than 3 years in current role

```
set.seed(2020)
testdata_rev <- subset(testdata, testdata$YearsInCurrentRole <= 3,)
predict_RF <- predict(model_RF, newdata = testdata_rev)
table(testdata_rev$Attrition, predict_RF)
```

```
##      predict_RF
##        No Yes
##   No  154   4
##   Yes  25  80
```

Result - predicted attrition rate is 30%