CaseStudy2

William Jones

2023-04-08

#Downloading the Files needed for the Analysis

```
noattr = read.csv("D:/Github/MSDS_6306_Doing-Data-Science/Unit 14 and 15 Case Study 2/CaseStudy2CompSet
nosal = read.csv("D:/Github/MSDS_6306_Doing-Data-Science/Unit 14 and 15 Case Study 2/CaseStudy2CompSet N
case = read.csv("D:/Github/MSDS_6306_Doing-Data-Science/Unit 14 and 15 Case Study 2/CaseStudy2-data.csv

#noattr = read.csv("/Users/williamjones/Downloads/CaseStudy2CompSet No Attrition.csv")
#nosal = read.csv("/Users/williamjones/Downloads/CaseStudy2CompSet No Salary.csv")
#case = read.csv("/Users/williamjones/Downloads/CaseStudy2-data.csv")


#Checking to see if there are a NA values in the columns

colSums(is.na(case))
```

```
##                       ID                      Age                Attrition
##                        0                        0                        0
##           BusinessTravel                DailyRate               Department
##                        0                        0                        0
##         DistanceFromHome                Education           EducationField
##                        0                        0                        0
##            EmployeeCount           EmployeeNumber  EnvironmentSatisfaction
##                        0                        0                        0
##                   Gender               HourlyRate           JobInvolvement
##                        0                        0                        0
##                 JobLevel                  JobRole          JobSatisfaction
##                        0                        0                        0
##            MaritalStatus            MonthlyIncome              MonthlyRate
##                        0                        0                        0
##       NumCompaniesWorked                   Over18                 OverTime
##                        0                        0                        0
##        PercentSalaryHike        PerformanceRating RelationshipSatisfaction
##                        0                        0                        0
##            StandardHours         StockOptionLevel        TotalWorkingYears
##                        0                        0                        0
##    TrainingTimesLastYear          WorkLifeBalance           YearsAtCompany
##                        0                        0                        0
##       YearsInCurrentRole  YearsSinceLastPromotion     YearsWithCurrManager
##                        0                        0                        0
```

```r
colSums(is.na(nosal))
```

```
##                       ID                      Age                Attrition
##                        0                        0                        0
##           BusinessTravel                DailyRate               Department
##                        0                        0                        0
##         DistanceFromHome                Education           EducationField
##                        0                        0                        0
##            EmployeeCount           EmployeeNumber  EnvironmentSatisfaction
##                        0                        0                        0
##                   Gender               HourlyRate           JobInvolvement
##                        0                        0                        0
##                 JobLevel                  JobRole          JobSatisfaction
##                        0                        0                        0
##            MaritalStatus              MonthlyRate       NumCompaniesWorked
##                        0                        0                        0
##                   Over18                 OverTime        PercentSalaryHike
##                        0                        0                        0
##        PerformanceRating RelationshipSatisfaction            StandardHours
##                        0                        0                        0
##         StockOptionLevel        TotalWorkingYears    TrainingTimesLastYear
##                        0                        0                        0
##          WorkLifeBalance           YearsAtCompany       YearsInCurrentRole
##                        0                        0                        0
##  YearsSinceLastPromotion     YearsWithCurrManager
##                        0                        0
```

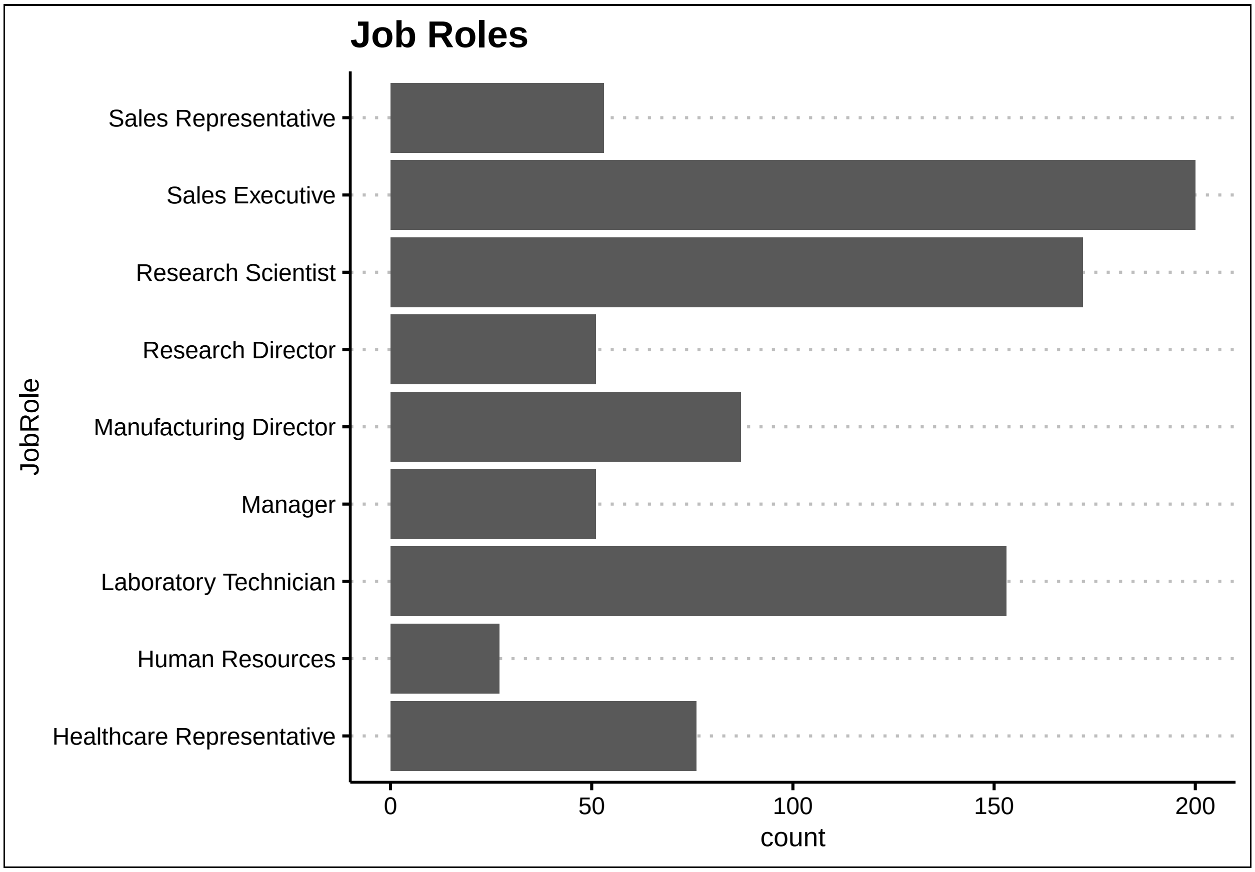
#Analysis of trends per job rol

```r
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## v purrr   1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

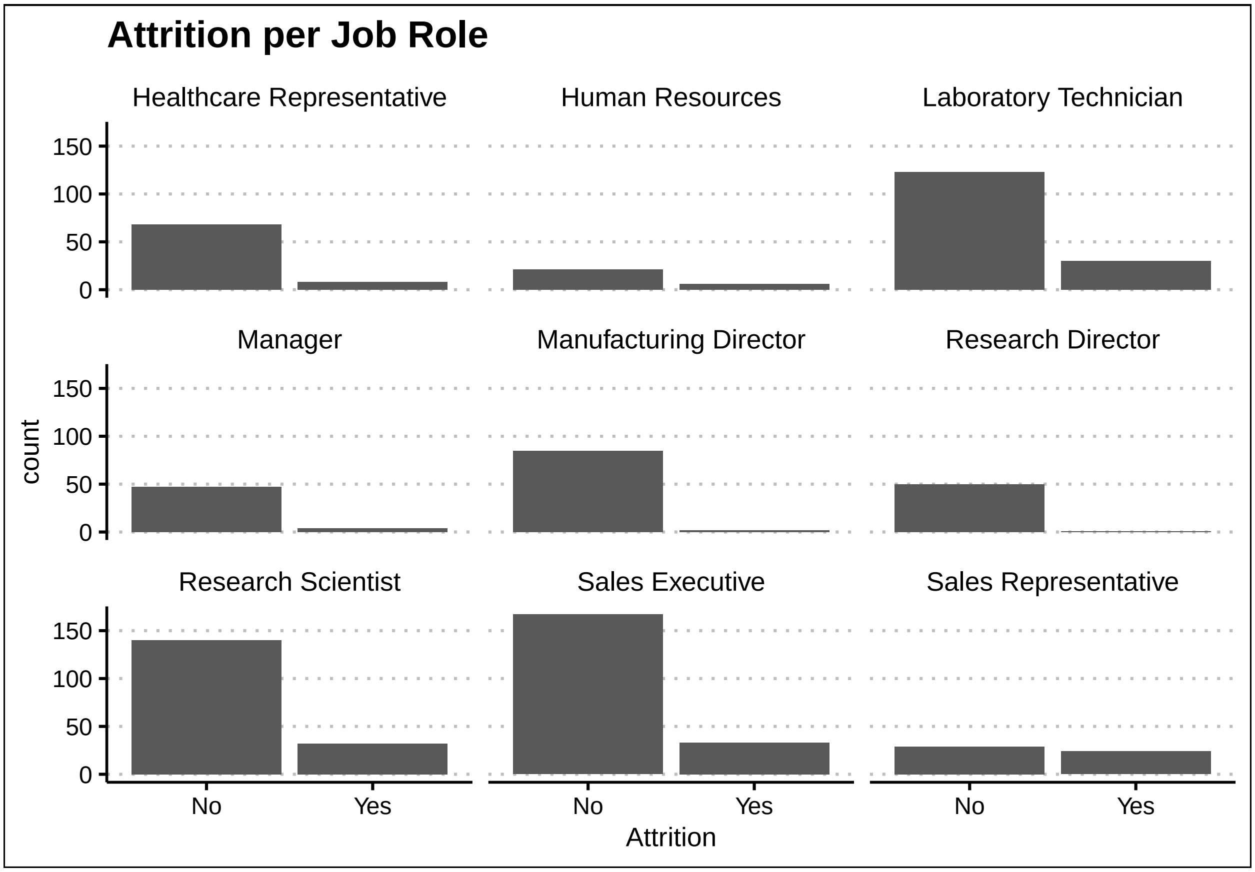
```r
library(ggthemes)
#job role breakdown
case %>% ggplot(aes(y=JobRole)) + geom_histogram(stat="count") +ggtitle("Job Roles") + theme_clean()
```

```
## Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
## `binwidth`, `bins`, and `pad`
```



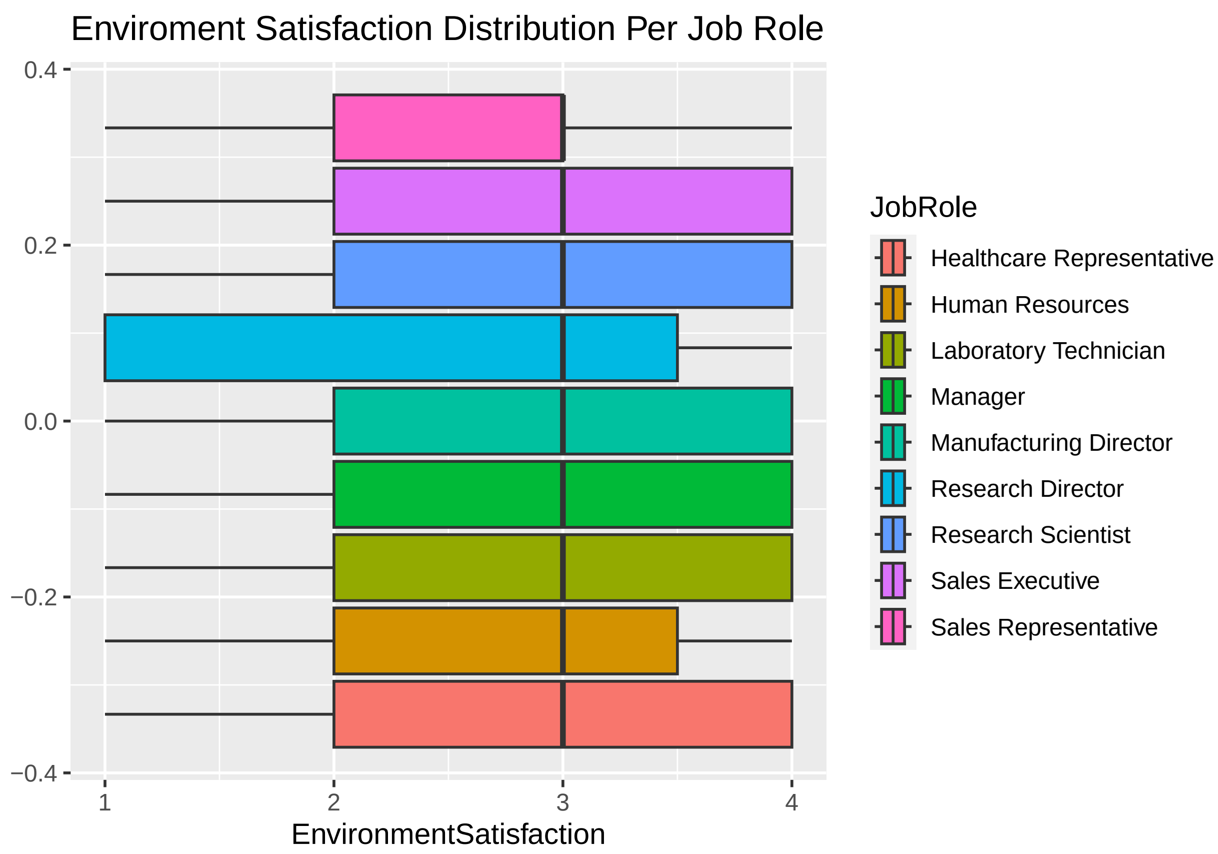

```
#job role attrition
case %>% ggplot(aes(x = Attrition)) + facet_wrap(~JobRole) + geom_histogram(stat="count") + ggtitle("At
```

```
## Warning in geom_histogram(stat = "count"): Ignoring unknown parameters:
## ‘binwidth‘, ‘bins‘, and ‘pad‘
```



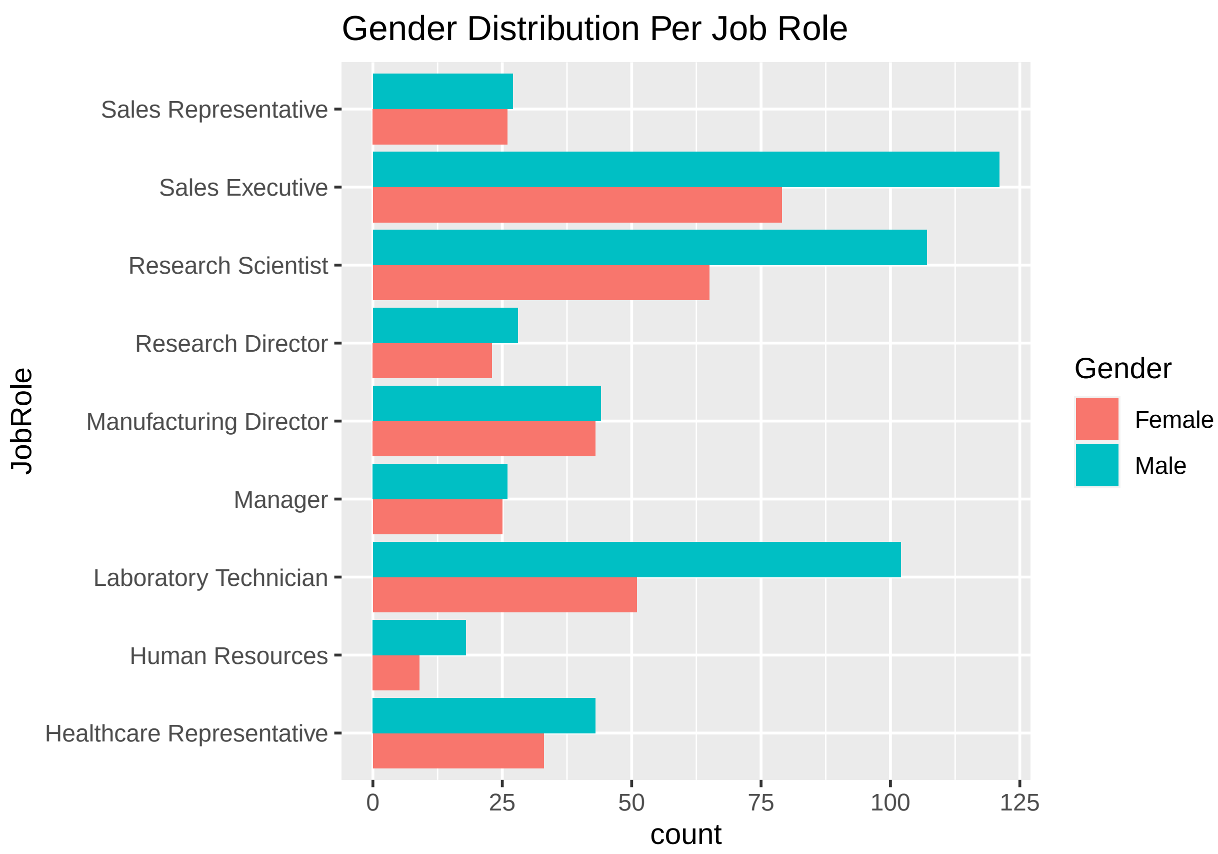

```
#job role salary
case %>% ggplot(aes(x = MonthlyIncome, fill = JobRole)) + geom_boxplot()  + ggtitle("Salary Per Job Rol
```


```r
#Enviroment Satisfaction
case %>% ggplot(aes(x=EnvironmentSatisfaction, fill = JobRole)) + geom_boxplot() +
  ggtitle("Enviroment Satisfaction Distribution Per Job Role")
```



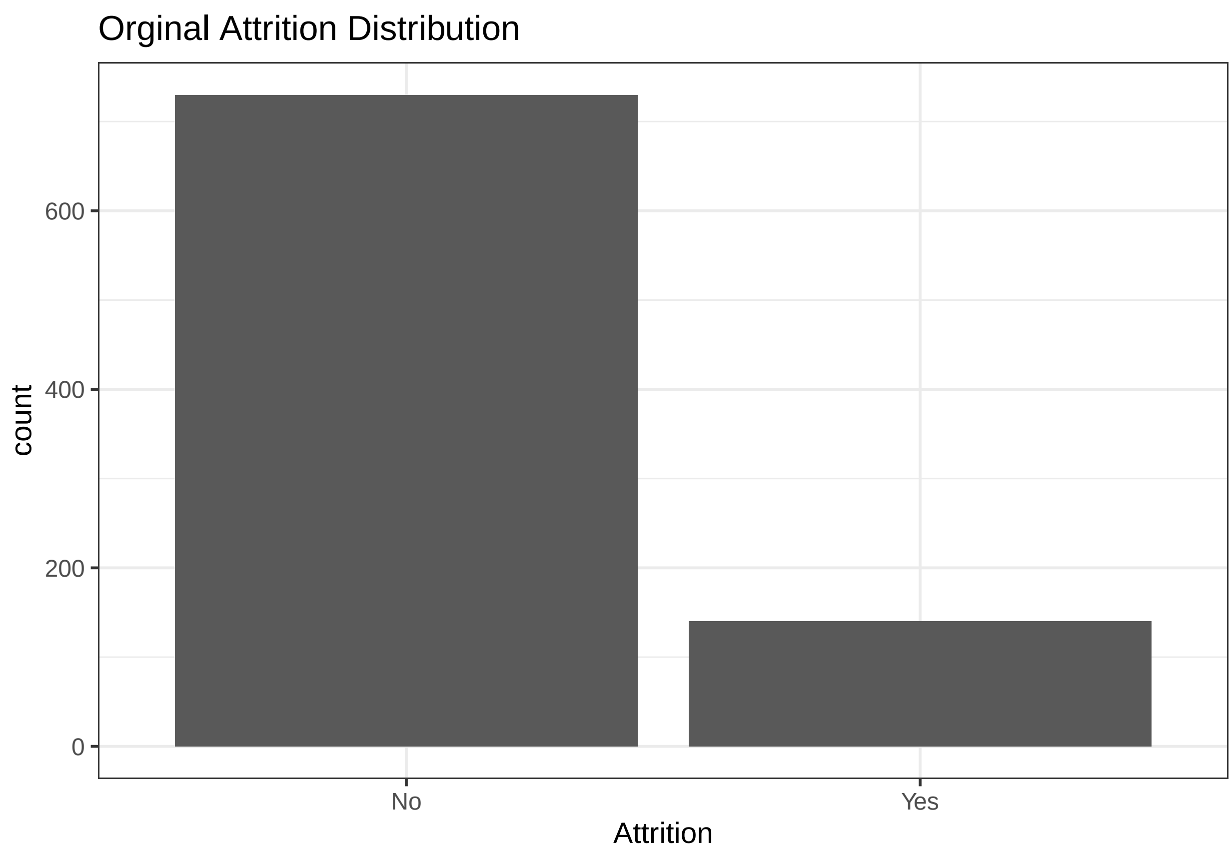

```
#Gender
case %>% ggplot(aes(y= JobRole, fill=Gender)) + geom_histogram(stat="count", position="dodge") + ggtitl
```

```
## Warning in geom_histogram(stat = "count", position = "dodge"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```



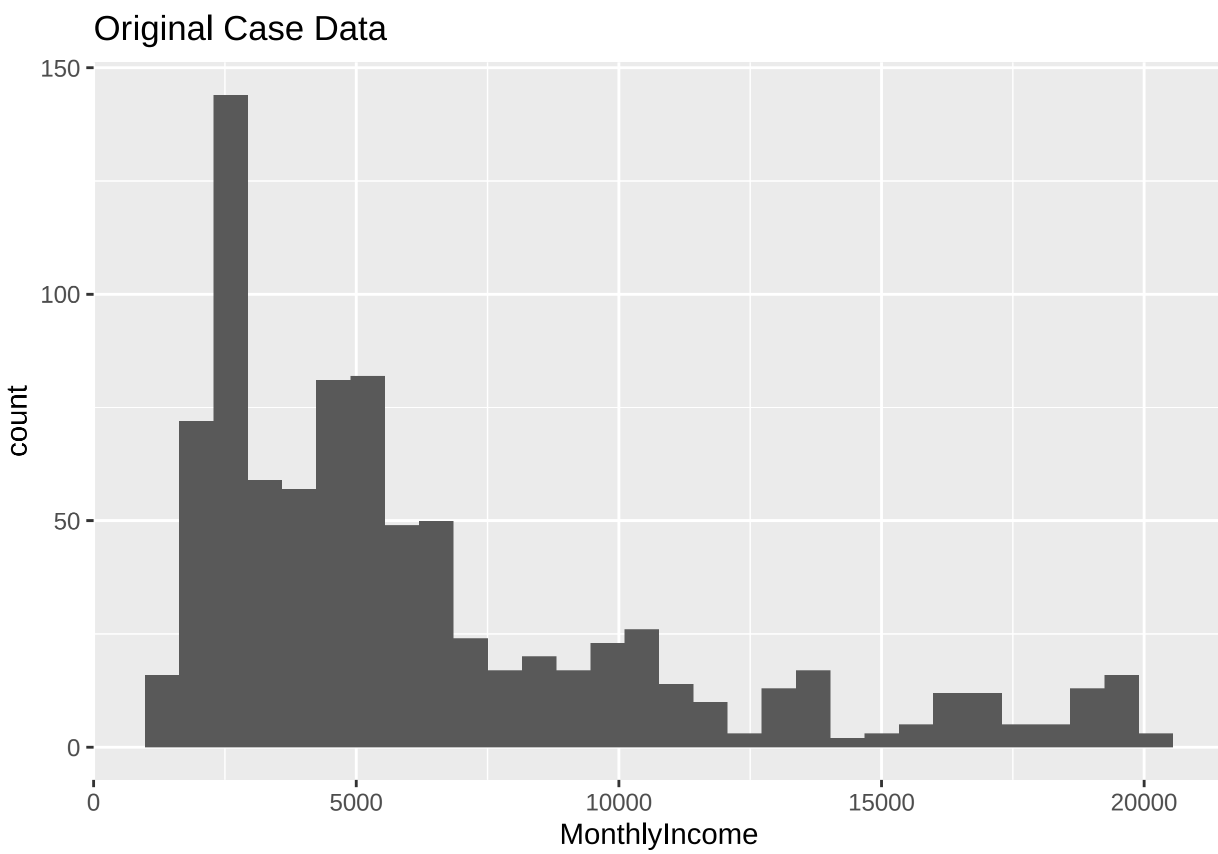
#Intial Analysis of the data

```
library(ggplot2)
library(dplyr)
library(ggthemes)
#Distribution of the Attrition Rate
case %>% ggplot(aes(x= Attrition), color = Attrition) + geom_bar(stat="count") + theme_bw() + ggtitle("(
```




```r
#Distibution of Salary
case %>% ggplot(aes(x=MonthlyIncome)) + geom_histogram() + ggtitle("Original Case Data")
```

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



#Since there are no missing variables, I will check the columns to see which ones need to be dropped then convert catagorical variables to #numeric for correlation

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.2.3
```

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

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

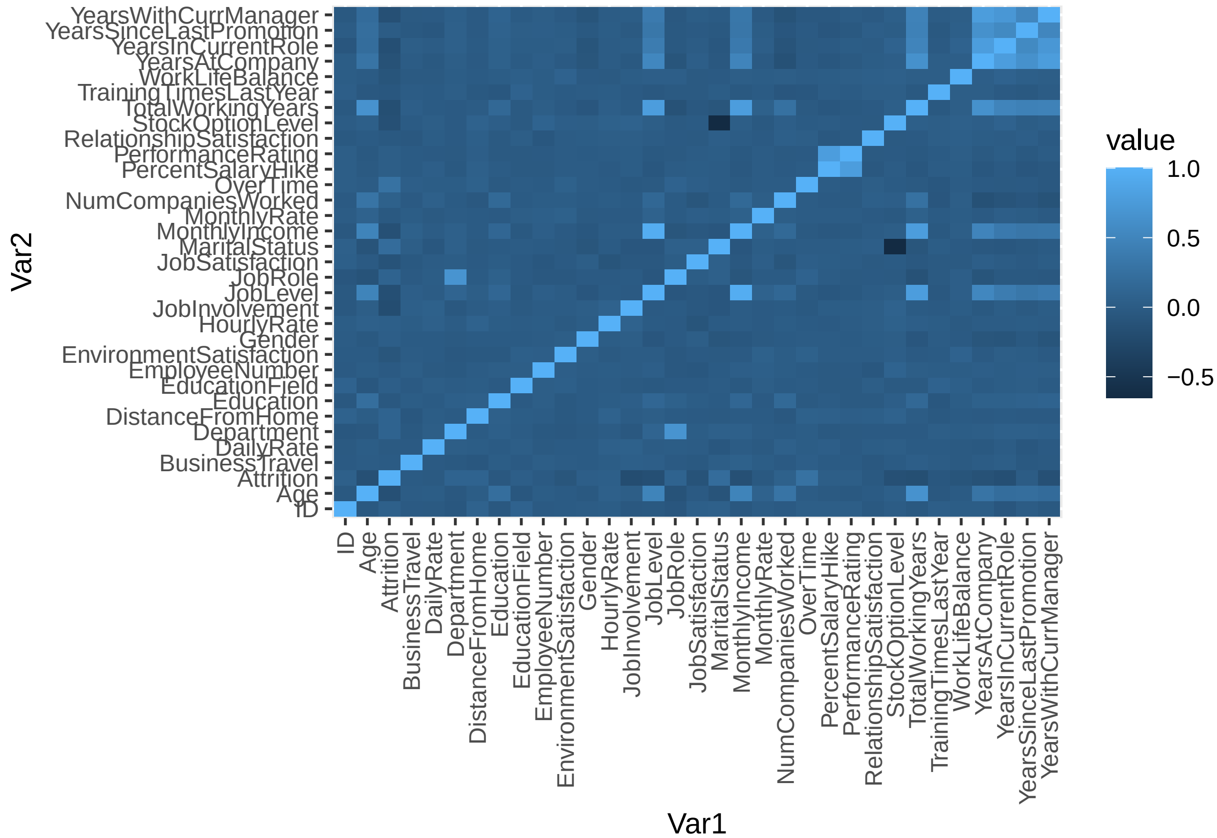
```
library(ggplot2)
library(dplyr)
library(tidyr)
#checking number of distinct values in columns
sapply(case, function(x) n_distinct(x))
```

```
##                     ID                    Age              Attrition
##                    870                     43                      2
##         BusinessTravel              DailyRate             Department
##                      3                    627                      3
```

```
##         DistanceFromHome                Education           EducationField
##                       29                        5                        6
##            EmployeeCount           EmployeeNumber  EnvironmentSatisfaction
##                        1                      870                        4
##                   Gender               HourlyRate           JobInvolvement
##                        2                       71                        4
##                 JobLevel                  JobRole          JobSatisfaction
##                        5                        9                        4
##            MaritalStatus            MonthlyIncome              MonthlyRate
##                        3                      826                      852
##       NumCompaniesWorked                   Over18                 OverTime
##                       10                        1                        2
##        PercentSalaryHike        PerformanceRating RelationshipSatisfaction
##                       15                        2                        4
##            StandardHours         StockOptionLevel        TotalWorkingYears
##                        1                        4                       39
##    TrainingTimesLastYear          WorkLifeBalance           YearsAtCompany
##                        7                        4                       32
##       YearsInCurrentRole  YearsSinceLastPromotion     YearsWithCurrManager
##                       19                       16                       17
```

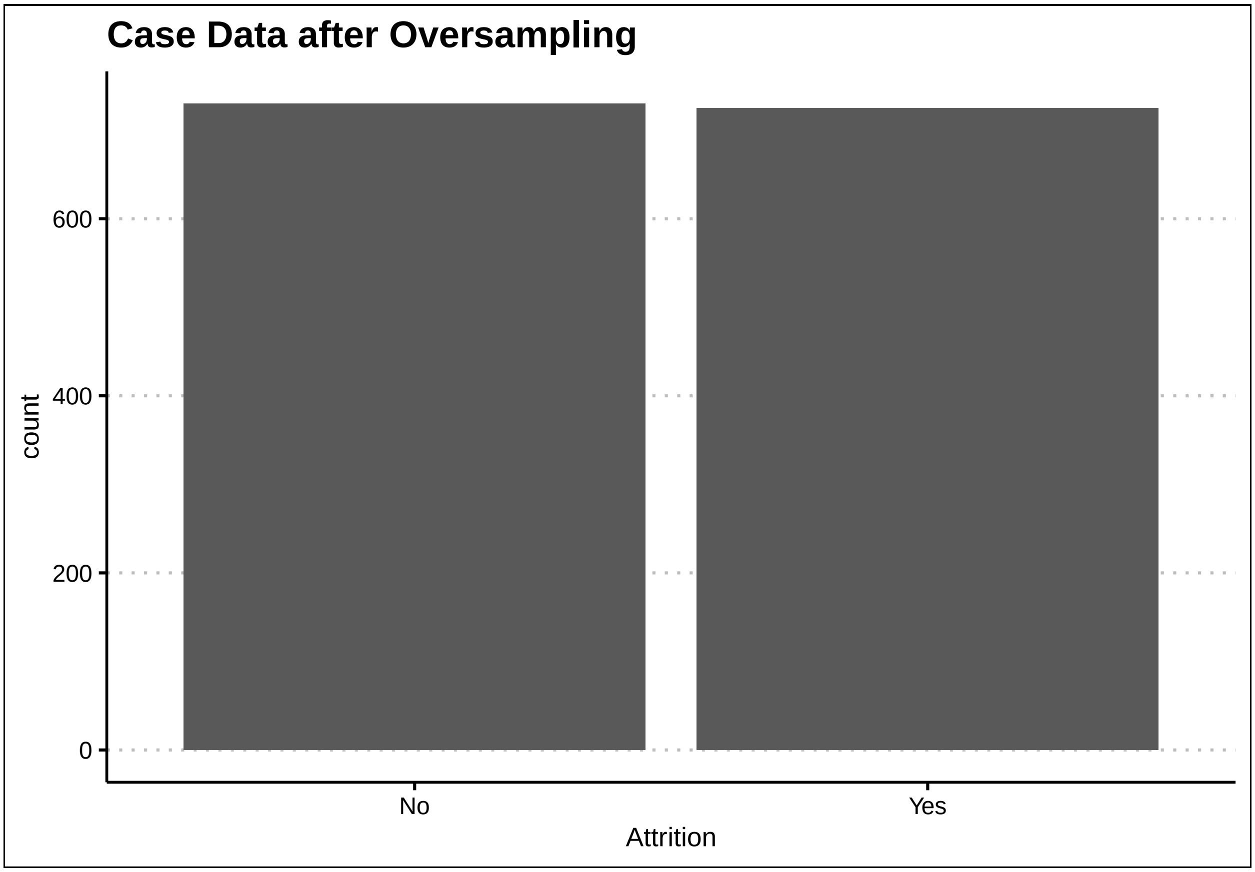
```
#droppinmg columns that have only one unique value
case = subset(case, select = -c(10, 23, 28))
#dropping columns
noattr = subset(noattr, select = -c(9, 22, 27))
#converting catagorical variables to factors
 case[, c(3, 4, 6, 9, 12, 16, 18, 22)] <- lapply(case[, c(3, 4, 6, 9, 12, 16, 18, 22)], as.factor)
#copy dataframe with different memory address
case_f = data.frame(case)
noattr[, c(3, 5, 8, 11, 15, 17, 21)] <- lapply(noattr[, c(3, 5, 8, 11, 15, 17, 21)], as.factor)
noattr_f = data.frame(noattr)
#converting factor columns to numeric

case[,  c(3, 4, 6, 9, 12, 16, 18, 22)] <- sapply(case[,  c(3, 4, 6, 9, 12, 16, 18, 22)], unclass)
noattr[, c(3, 5, 8, 11, 15, 17, 21)] <- sapply(noattr[, c(3, 5, 8, 11, 15, 17, 21)], unclass)
#correlation matrix
cormat <- round(cor(case), 2)
melted_cormat <- melt(cormat)
ggplot(data = melted_cormat, aes(x=Var1, y=Var2, fill=value)) + geom_tile() + theme(axis.text.x = elemen
```



The distribution of the Attrition is heavily un balanced. To help in classification algorithms Oversampling needs to be done to balance the dataset without losing any observations.

```
set.seed(12345)
#subsetting miniority class
case_minority <- case_f %>% filter(Attrition == "Yes")
maj <- nrow(case_f[case_f$Attrition == 'No', ])
min <- nrow(case_f[case_f$Attrition == 'Yes', ])
#oversampling the minority class to create a somewhat balanced dataset
set = maj-min-5
for (i in 1:set){
  case_f[nrow(case_f) + 1,] <- sample_n(case_minority, 1)
}
case_f %>% ggplot(aes(x=Attrition)) + geom_bar(stat="count") + theme_clean() + ggtitle("Case Data after
```



Creating Knn model for prediction

A power model to show the best k value to use for the model

```
library(class)
library(e1071)
library(caret)
```

```
## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.2.3

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(ggplot2)
library(dplyr)
#creating dataframes for metrics
```

```r
accs = data.frame(accuracy = numeric(25), k = numeric(25))
sens = data.frame(sensitivity = numeric(25), k = numeric(25))
spec = data.frame(specificity = numeric(25), k = numeric(25))
#changing model to numeric for knn
case_n = data.frame(case_f)
case_n[,  c(3, 4, 6, 9, 12, 16, 18, 22)] <- sapply(case_n[,  c(3, 4, 6, 9, 12, 16, 18, 22)], unclass)

#Figuring out which K value to us
for(i in 1:25)
{
  #Knn cross validation model
  classifications = knn.cv(case_n[,-3],case_n$Attrition, prob = TRUE, k = i, use.all = FALSE)
  #creating a table
  table(case_n$Attrition,classifications)
  #Confusion Matrix
  CM = confusionMatrix(table(case_n$Attrition,classifications))

  #Adding the metrics to their perspective dataframes
  accs$accuracy[i] = CM$overall[1]
  sens$sensitivity[i] = CM$byClass[1]
  spec$specificity[i] = CM$byClass[2]
  #adding k value to dataframes
  accs$k[i] = i
  sens$k[i] = i
  spec$k[i] = i
}
#Plotting the metrics
ggplot() +
  geom_line(data = accs, aes(k,accuracy,  colour ="Accuracy")) +
  geom_line(data = sens ,aes(k,sensitivity,  colour ="Sensitivity")) +
  geom_line(data = spec, aes(k,specificity, colour = "Specificity")) +
  ggtitle("Attrition Case Study") +
  ylab("ratio") +
  xlab("K") +
  scale_color_manual(values = c("Accuracy" = "blue", "Sensitivity" = "red", "Specificity" = "purple")) +
  labs(color = "Metric")
```

Looking at the outputted matrix of the power model it seems that a K value around 1-10 would be the best K value for a model that has atleast 60% in specificity and Sensitivity

Running the knn model with the specified k value

```
library(tidyr)
library(caret)
#sample size
smpsize <- floor(0.8 * nrow(case_f))
#partition
set.seed(123)
ind <- sample(seq_len(nrow(case_f)), size = smpsize)
train<- case_f[ind, ]
test <- case_f[-ind, ]

trctrl <- trainControl(method = "repeatedcv", number = 10, repeats = 3)
set.seed(3333)

knn_fit <- train(Attrition ~.,
                 data = train,
                 method = "knn",
                 trControl=trctrl,
```

```r
                 preProcess = c("center", "scale"),
                 tuneLength = 10)
knn_fit
```

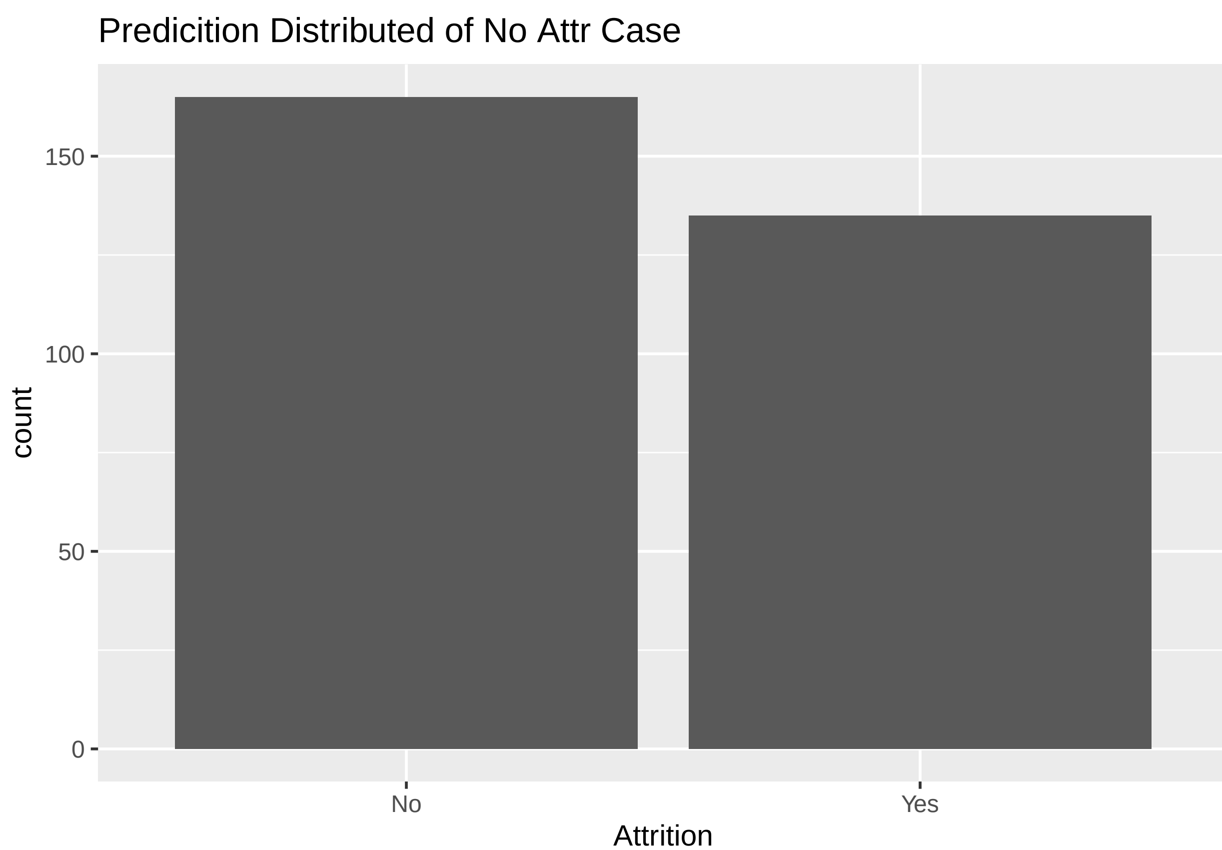
```
## k-Nearest Neighbors
##
## 1164 samples
##   32 predictor
##    2 classes: 'No', 'Yes'
##
## Pre-processing: centered (46), scaled (46)
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 1047, 1048, 1047, 1048, 1048, 1048, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.7588859  0.5160461
##    7  0.7603325  0.5192079
##    9  0.7345368  0.4679331
##   11  0.7319604  0.4629220
##   13  0.7311008  0.4614178
##   15  0.7127591  0.4251670
##   17  0.7199307  0.4395155
##   19  0.7265178  0.4526435
##   21  0.7336821  0.4671567
##   23  0.7262329  0.4523990
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 7.
```

```r
#prediction on the test case
pred <- predict(knn_fit, newdata= test)
confusionMatrix(pred, test$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No   96  23
##        Yes  60 112
##
##                Accuracy : 0.7148
##                  95% CI : (0.6592, 0.7659)
##     No Information Rate : 0.5361
##     P-Value [Acc > NIR] : 3.331e-10
##
##                   Kappa : 0.437
##
##  Mcnemar's Test P-Value : 7.766e-05
##
##             Sensitivity : 0.6154
##             Specificity : 0.8296
##          Pos Pred Value : 0.8067
```

```
##          Neg Pred Value : 0.6512
##              Prevalence : 0.5361
##          Detection Rate : 0.3299
##    Detection Prevalence : 0.4089
##       Balanced Accuracy : 0.7225
##
##        'Positive' Class : No
##
```

```r
#prediction on no attr case
noattr_f$Attrition <- predict(knn_fit, newdata = noattr_f)
noattr_f %>% ggplot(aes(x = Attrition)) + geom_bar(stat="count") + ggtitle("Predicition Distributed of N
```




```r
#isolating the attrition and id
ans <- noattr_f[c("ID", "Attrition")]
#putting this to its own csv
write.csv(ans, "D:/Downloads/Case2PredictionsJones Attrition.csv")
```

The model choose a k of 5 to be the best fit to predict the test case. Which predicted a somewhat normally distributed attrition of Nos and Yes.

Now we will try to conduct a predictive model on trying to predict the monthly income. For this model we will try a regression model

```r
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.2.3
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(tidyr)
library(car)
```

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

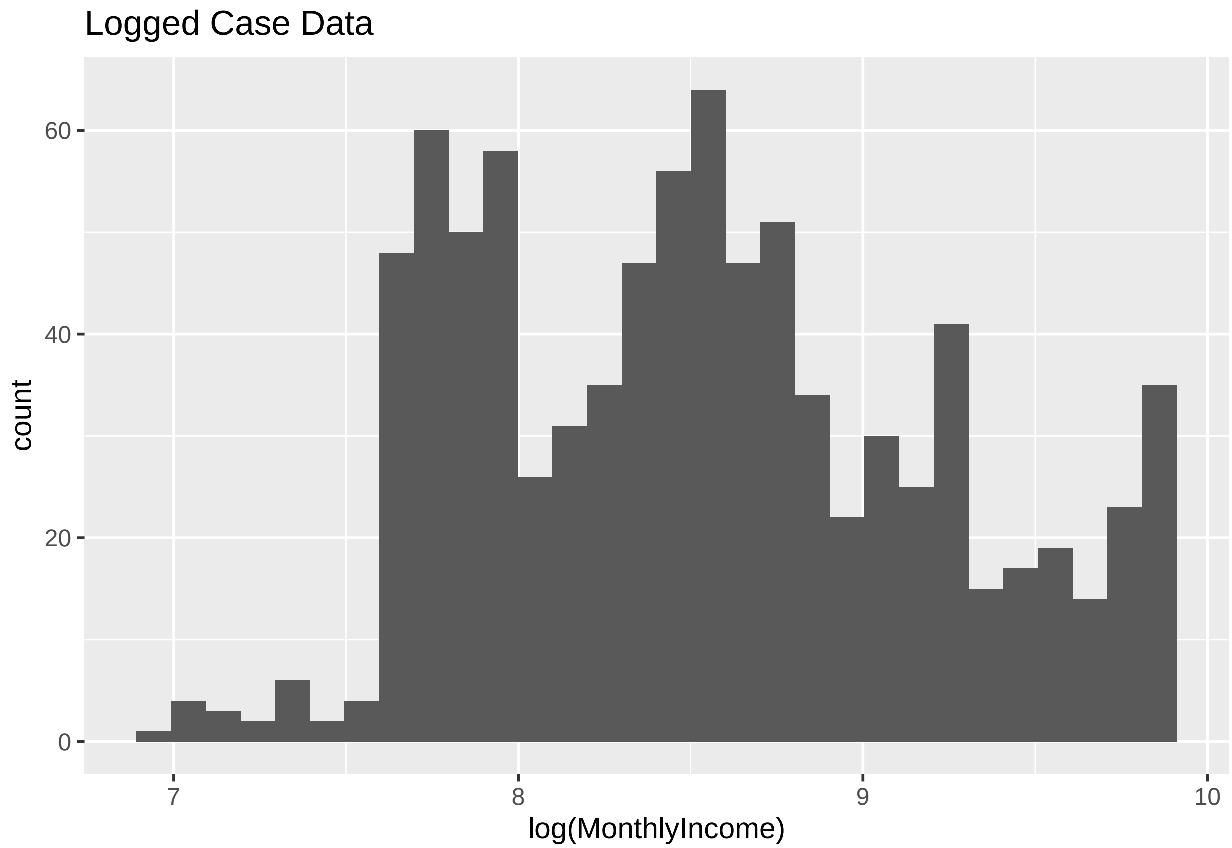
```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

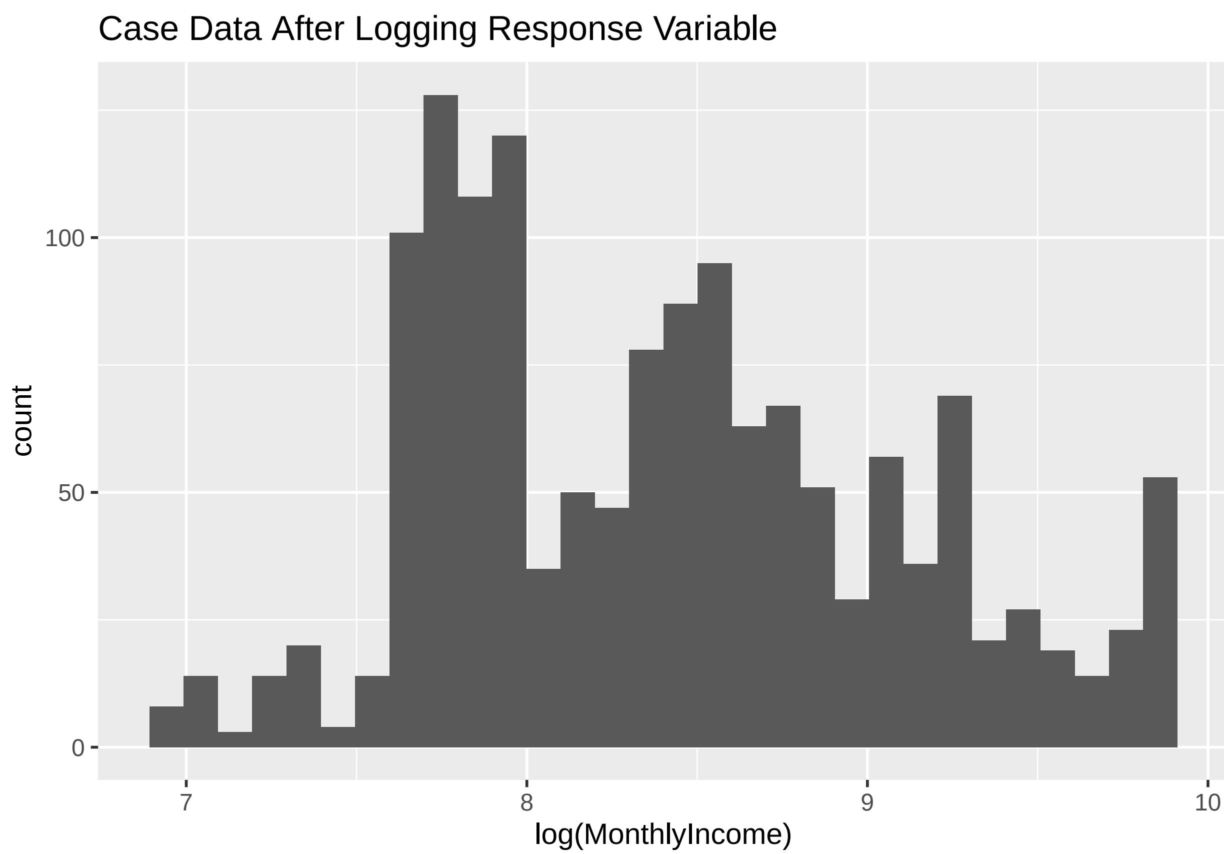
```r
#to combat the skewness of the response variable the log transformation is needed to make it normally d
case %>% ggplot(aes(x=log(MonthlyIncome))) + geom_histogram() + ggtitle("Logged Case Data")
```

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




```r
#dropping columns with only one unique variable
nosal = subset(nosal, select = -c(10, 22, 27))
#changing the columns to factors
nosal[, c(3, 4, 6, 9, 12, 16, 18, 21 )] <- lapply(nosal[, c(3, 4, 6, 9, 12, 16, 18, 21 )], as.factor)
#convert the factors to numeric
nosal[, c(3, 4, 6, 9, 12, 16, 18, 21 )] <- sapply(nosal[, c(3, 4, 6, 9, 12, 16, 18, 21 )], unclass)
#since the salary distribution is heavily skewed it needs to be transformed
case_n %>% ggplot(aes(x=log(MonthlyIncome))) + geom_histogram() + ggtitle("Case Data After Logging Respo
```

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

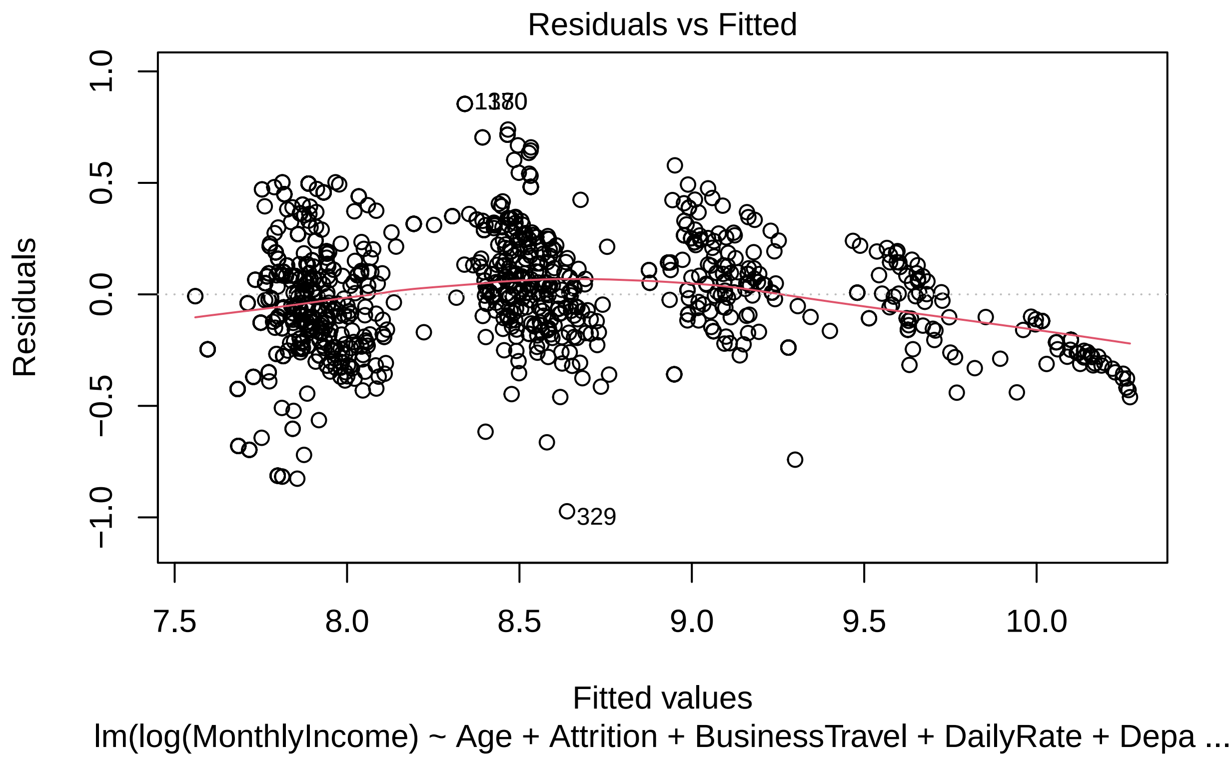


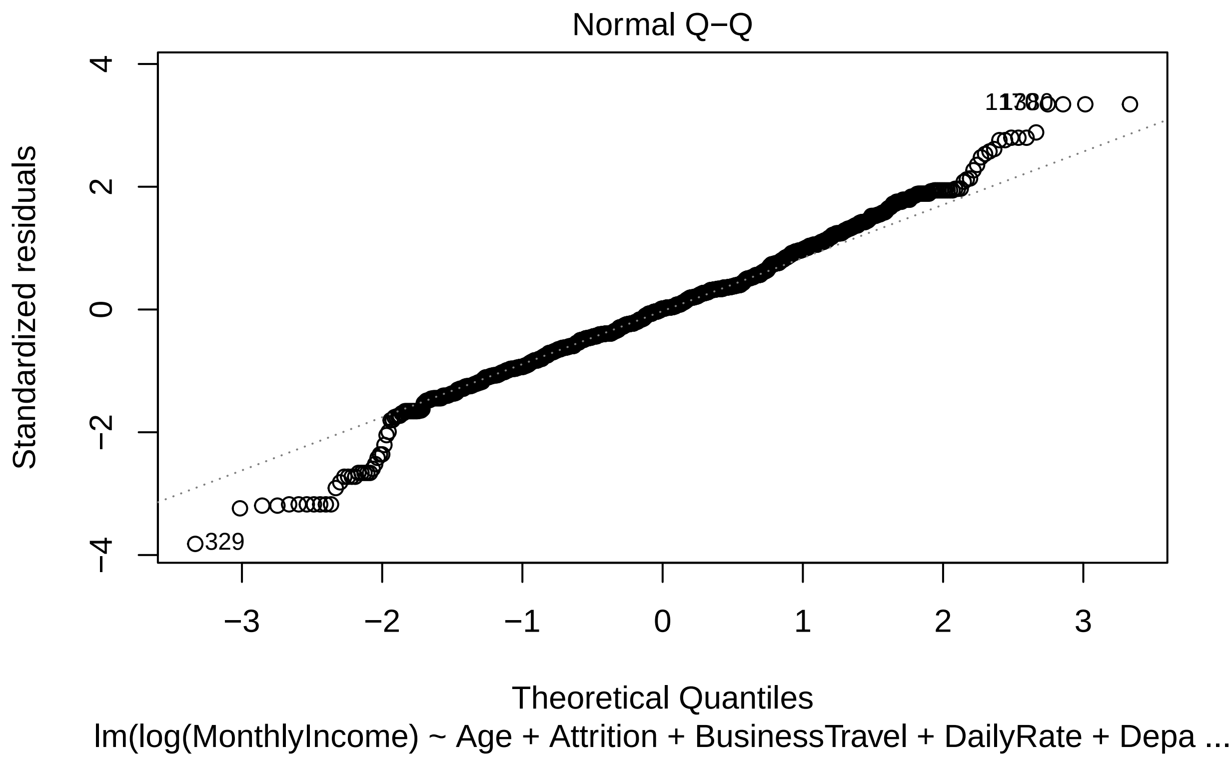

```r
#sample size
smpsize <- floor(0.8 * nrow(case_n))
#partition
set.seed(123)
ind <- sample(seq_len(nrow(case_n)), size = smpsize)
train<- case_n[ind, ]
test <- case_n[-ind, ]
#Using the no case numeric dataset to create a lm model
full.model = lm(log(MonthlyIncome)~., data = train)
#creating a stepwise model for parameter selection
step.model <- stepAIC(full.model, direction = "both",
                      trace = FALSE)
#summary
summary(step.model)
```

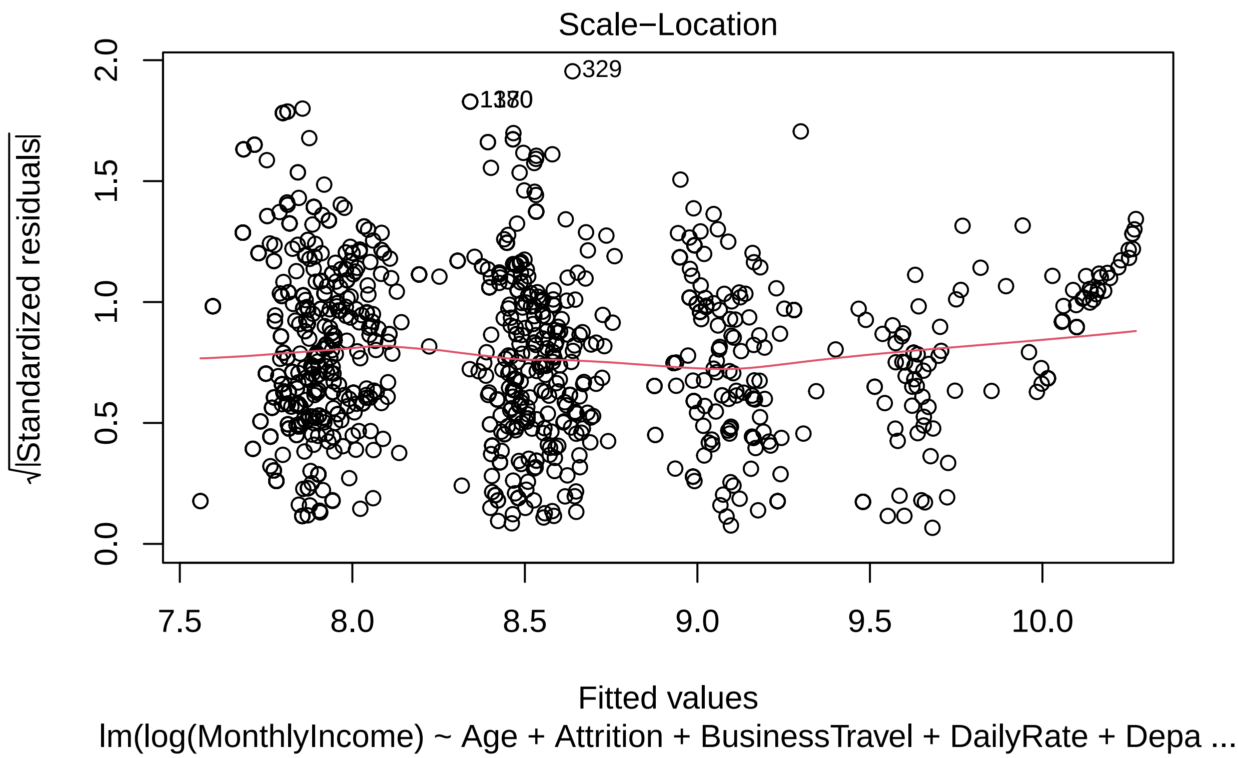
```
##
## Call:
## lm(formula = log(MonthlyIncome) ~ Age + Attrition + BusinessTravel +
##     DailyRate + Department + Education + EnvironmentSatisfaction +
##     JobLevel + JobSatisfaction + MaritalStatus + NumCompaniesWorked +
##     OverTime + PercentSalaryHike + PerformanceRating + RelationshipSatisfaction +
##     YearsInCurrentRole + YearsSinceLastPromotion, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97273 -0.15513  0.00337  0.14236  0.85416
```

```
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.432e+00  1.115e-01  66.666  < 2e-16 ***
## Age                      2.450e-03  9.951e-04   2.463 0.013943 *
## Attrition               -9.777e-02  1.799e-02  -5.434 6.73e-08 ***
## BusinessTravel           2.398e-02  1.173e-02   2.044 0.041185 *
## DailyRate                5.570e-05  1.906e-05   2.923 0.003540 **
## Department               5.018e-02  1.415e-02   3.545 0.000408 ***
## Education                1.829e-02  7.818e-03   2.340 0.019476 *
## EnvironmentSatisfaction  -1.309e-02  6.819e-03  -1.919 0.055179 .
## JobLevel                 5.200e-01  9.115e-03  57.048  < 2e-16 ***
## JobSatisfaction         -1.636e-02  6.956e-03  -2.352 0.018860 *
## MaritalStatus           -1.925e-02  1.146e-02  -1.681 0.093100 .
## NumCompaniesWorked       1.466e-02  3.145e-03   4.661 3.52e-06 ***
## OverTime                 3.422e-02  1.650e-02   2.074 0.038279 *
## PercentSalaryHike        1.082e-02  3.223e-03   3.357 0.000814 ***
## PerformanceRating       -1.051e-01  3.352e-02  -3.134 0.001766 **
## RelationshipSatisfaction -2.854e-02  6.761e-03  -4.221 2.62e-05 ***
## YearsInCurrentRole       1.266e-02  2.843e-03   4.455 9.22e-06 ***
## YearsSinceLastPromotion   6.415e-03  2.992e-03   2.144 0.032252 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2573 on 1146 degrees of freedom
## Multiple R-squared:  0.8624, Adjusted R-squared:  0.8604
## F-statistic: 422.5 on 17 and 1146 DF,  p-value: < 2.2e-16
```

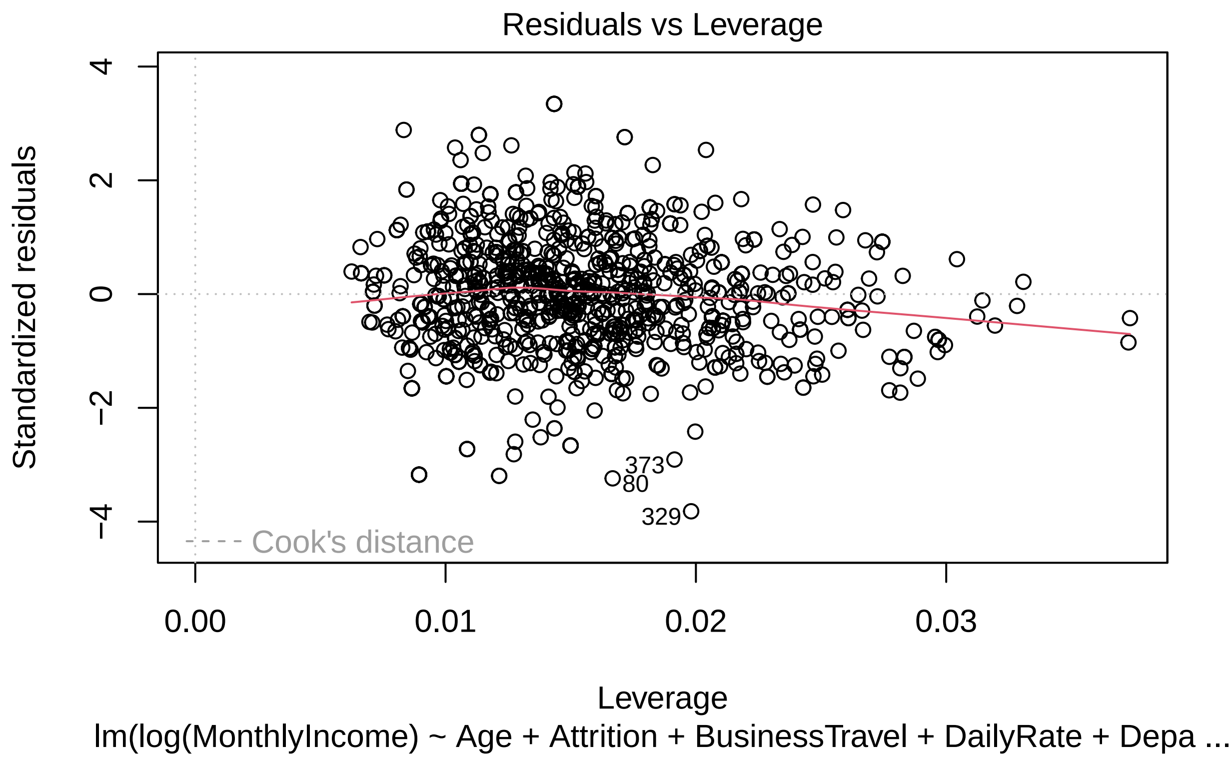
```
plot(step.model)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
1370
329
7.5
8.0
8.5
9.0
9.5
10.0
Fitted values
lm(log(MonthlyIncome) ~ Age + Attrition + BusinessTravel + DailyRate + Depa ...

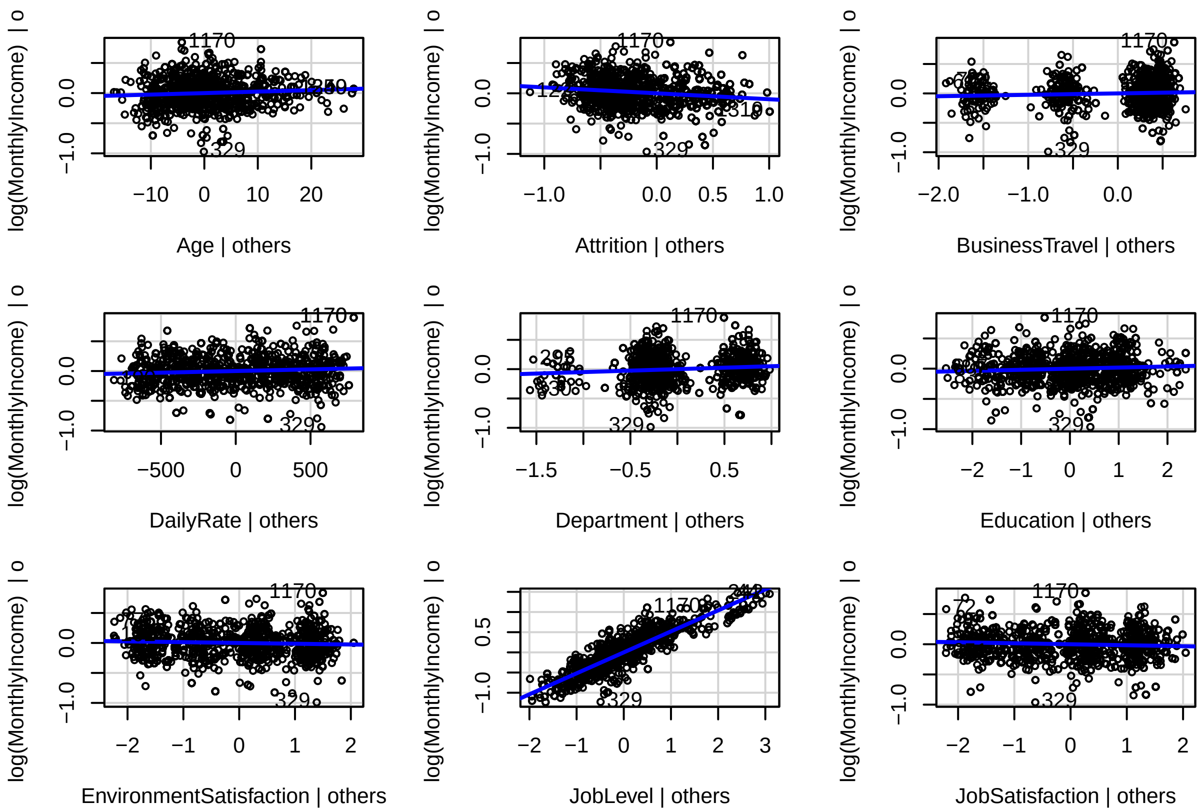

Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(log(MonthlyIncome) ~ Age + Attrition + BusinessTravel + DailyRate + Depa ...
329
−4
−2
0
2
4
−3
−2
−1
0
1
2
3

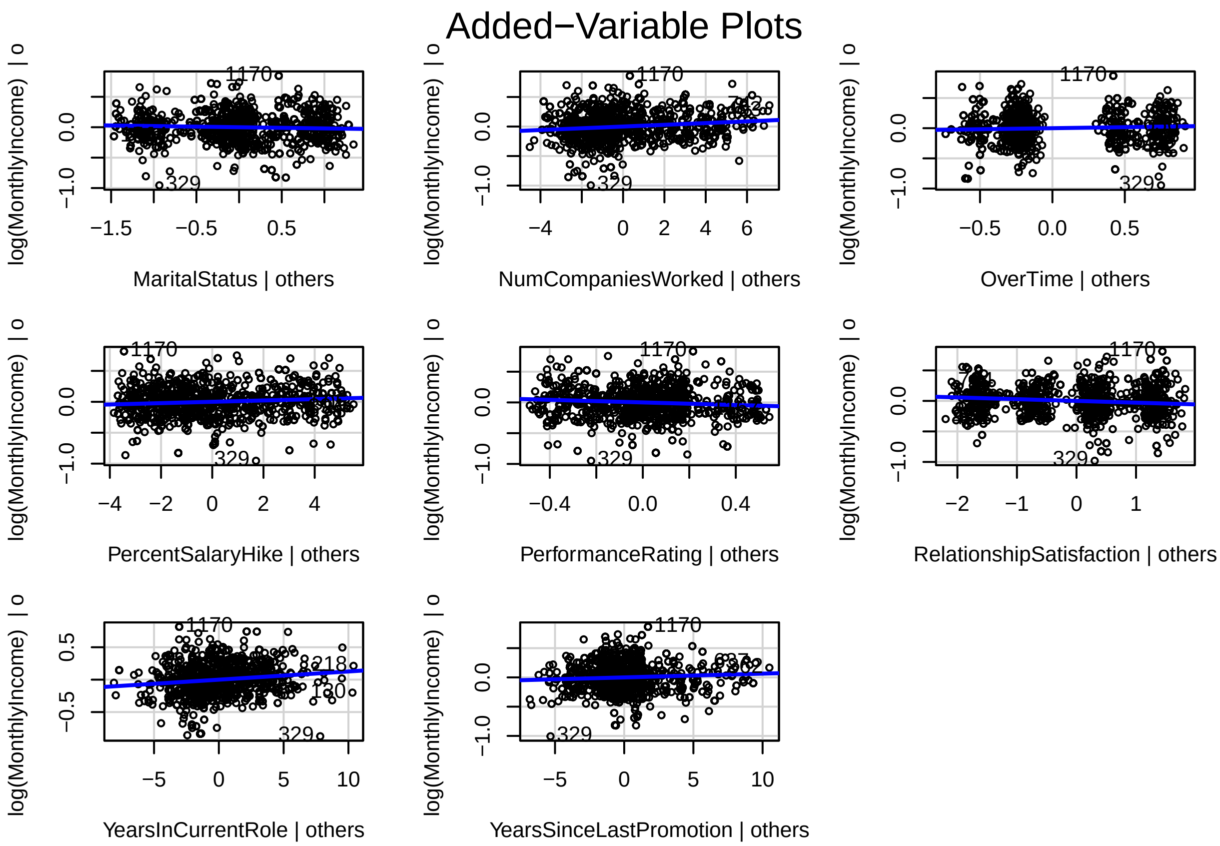


Scale−Location
329
1370
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
7.5
8.0
8.5
9.0
9.5
10.0
Fitted values
lm(log(MonthlyIncome) ~ Age + Attrition + BusinessTravel + DailyRate + Depa ...




```
#plotting the regression model
avPlots(step.model)
```


log(MonthlyIncome) | o
Age | others
Attrition | others
BusinessTravel | others
DailyRate | others
Department | others
Education | others
EnvironmentSatisfaction | others
JobLevel | others
JobSatisfaction | others



Looking at the assumptions for the regression model the data seems to follow somewhat of a straight line, being independent of each other, and having somewhat equal variance.

making prediction and testing accuracy on the test set

```
test$MonthlyIncome <- log(test$MonthlyIncome)
prediction <- step.model %>% predict(test)
#Model Performance
RMSE(prediction, test$MonthlyIncome)
```

```
## [1] 0.271951
```

```
R2(prediction, test$MonthlyIncome)
```

```
## [1] 0.8152173
```

The models difference between the true vs predicted values is only about 1.2 which is great for our model, its adjusted r2 shows that about 86% of the data in the training set is explained by the model. Using the found regression model we will now try to predict the test case with no salary

```
#prediction
pred <- step.model %>% predict(nosal)
```

```r
#convert to dataframe
pred <- as.data.frame(pred)
#rename column
colnames(pred) <- c("MonthlyIncome")
#add id column
pred$ID <- nosal$ID
#transform the income back
pred$MonthlyIncome <- exp(pred$MonthlyIncome)
#create csv file
write.csv(pred, "D:/Downloads/Case2PredictionsJones Salary.csv")
```