Employee Demographic Analysis and Predictions

Joel Laskow

2023-12-05

Introduction:

The purpose of this document is to guide readers through our analysis of employee information provided by FritoLay. In this study we sought to explore variables associated by employee attrition and employee salary (monthly income). From our findings we build predictive models to: 1) Predict employee attrition 2) Predict employee monthly income with multiple linear regression

We have also provided a link to an RShiny app we've developed for users to visualize employee age distribution by department.

Libraries

```
library(base)
library(class)
library(leaps)
library(caret)
library(caTools)
library(dplyr)
library(fastDummies)
library(naniar)
library(ggplot2)
library(GGally)
library(e1071)
library(RCurl)
library(aws.s3)
```

Load reference data

```
attritiondata_original<-read.table(textConnection(getURL(
  "https://s3.us-east-2.amazonaws.com/msds.ds.6306.2/CaseStudy2-data.csv"
)), sep=",", header=TRUE)

referencedata<-data.frame(attritiondata_original)


for (i in names(referencedata)) {
  if (class(referencedata[[i]]) == "character") {
    referencedata[[i]] <- factor(referencedata[[i]])
  }
}
```

Load "No Attrition" data

```
CaseStudy2CompSet.No.Attrition <- read.csv("/cloud/project/CaseStudy2CompSet No Attrition.csv", header="

noattrition<-data.frame(CaseStudy2CompSet.No.Attrition)

noattrition2<-noattrition
```

Load “No Salary” data

```
CaseStudy2CompSet.No.Salary...Sheet1 <- read.csv("/cloud/project/CaseStudy2CompSet No Salary - Sheet1.c

nosalaries<-data.frame(CaseStudy2CompSet.No.Salary...Sheet1)
```

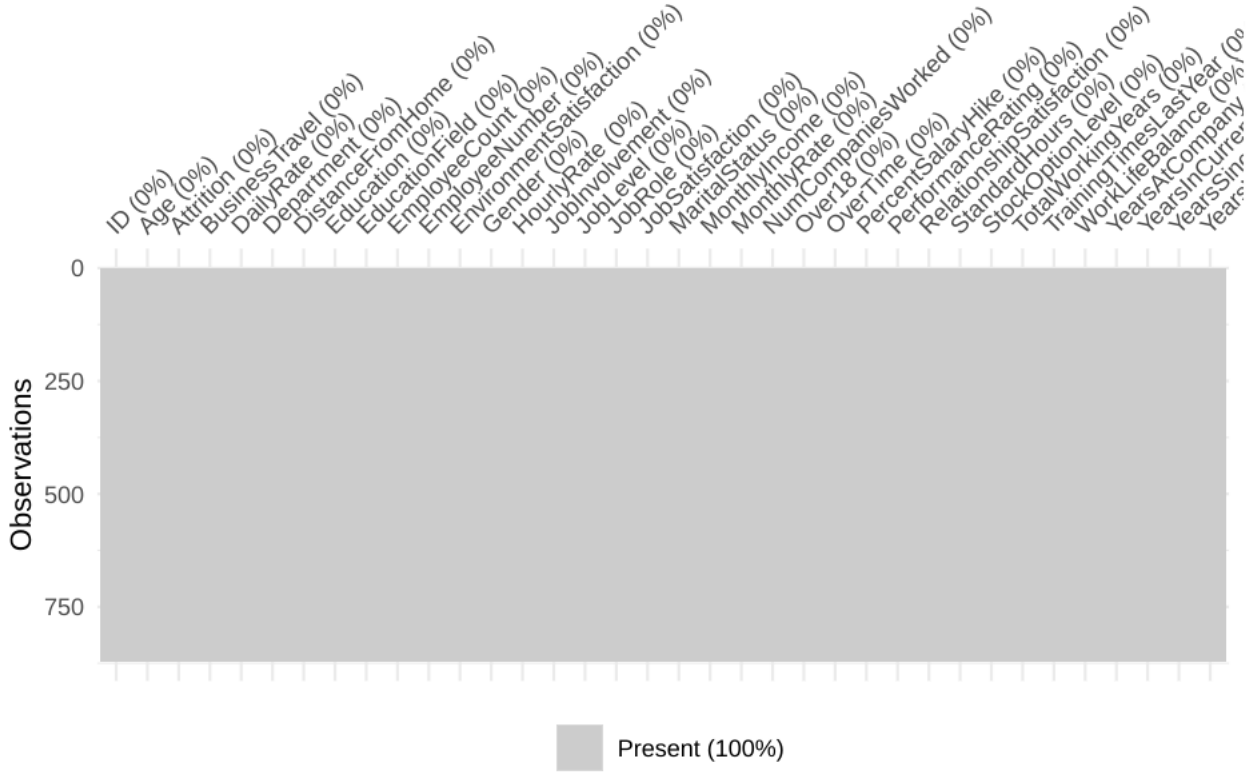
Convert all datasets to dataframes for convenience

```
nosalaries<-data.frame(nosalaries)
noattrition<-data.frame(noattrition)
data<-data.frame(referencedata)
```

Assessing data frames for missing values:

Full Dataset

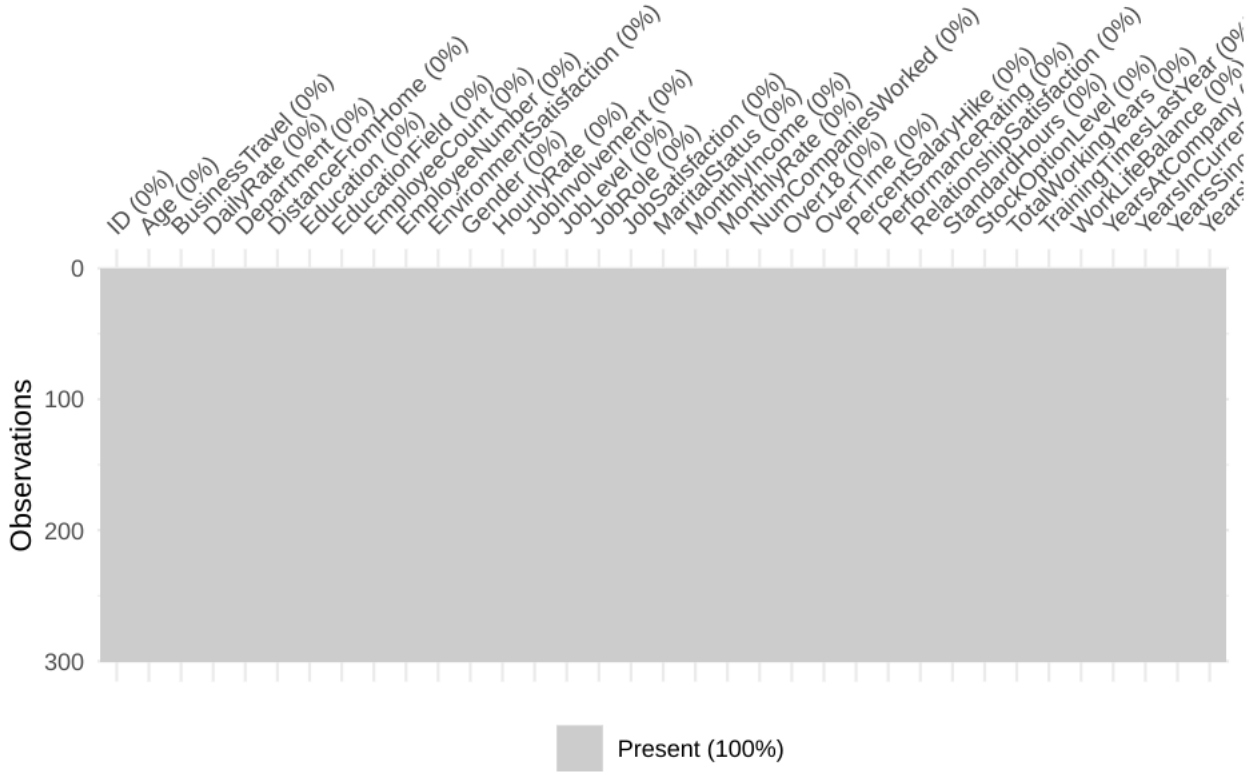
```
# Reference data

vis_miss(data)
```



No values missing

Dataframe without attrition

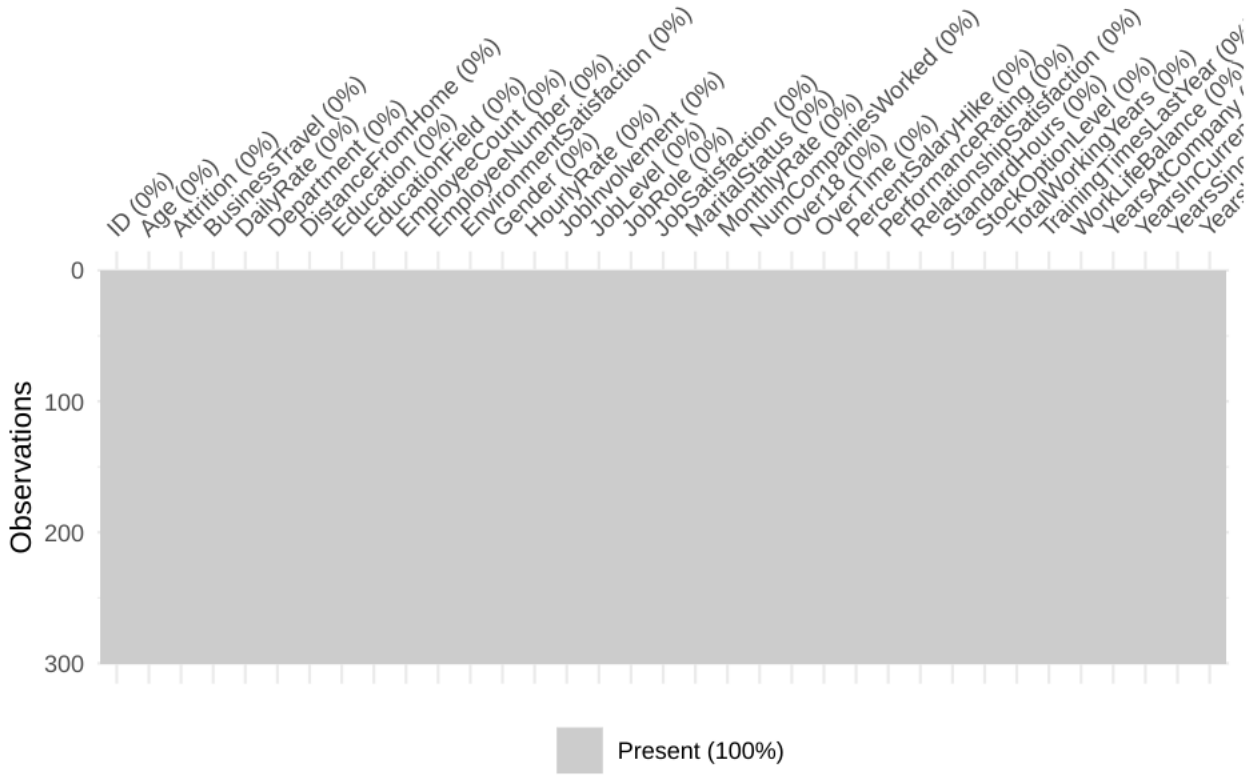
```
vis_miss(noattrition2)
```



No values missing

Dataframe without salaries

```
vis_miss(nosalaries)
```

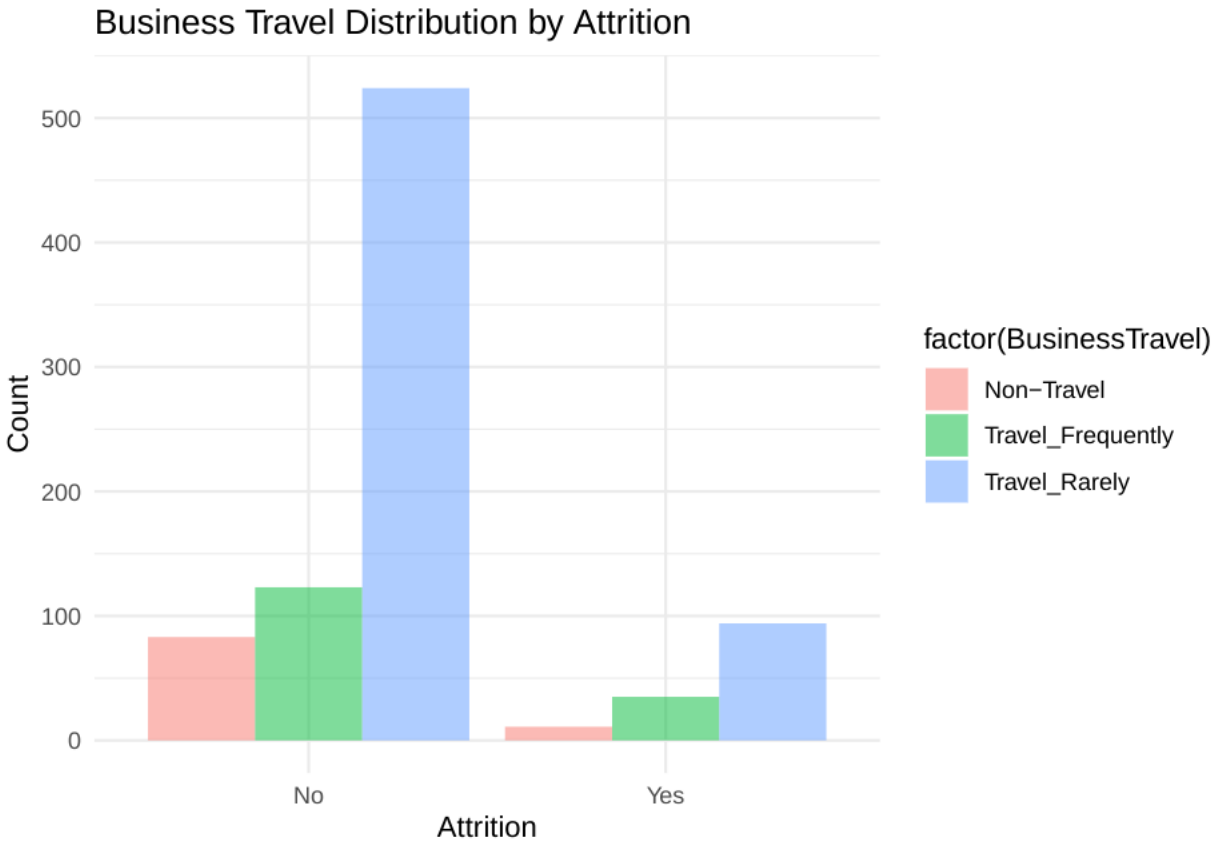


No data is missing from any of our datasets. We can proceed with analysis.

Part 1: Attrition Analysis

We will build frequency plots of Attrition based on different explanatory variables

```
ggplot(data, aes(x = factor(Attrition), fill = factor(BusinessTravel))) +
  geom_bar(position = "dodge", alpha = 0.5) +
  labs(x = "Attrition", y = "Count", title = "Business Travel Distribution by Attrition") +
  theme_minimal()
```



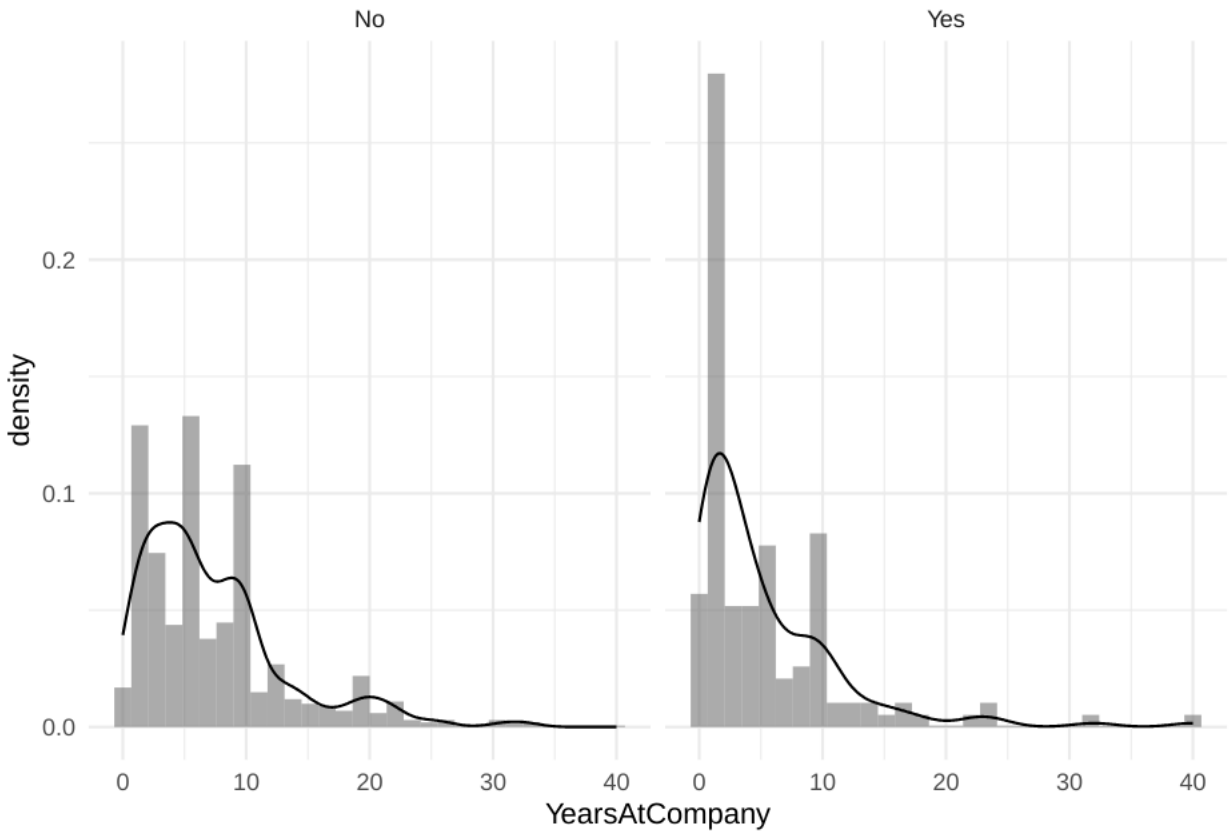
No clear association with BusinessTravel

What about YearsWithCompany?

```
ggplot(data, aes(x = YearsAtCompany)) +
  geom_histogram(position = "dodge", alpha = 0.5, aes(y = ..density..)) +
  theme_minimal() +
  geom_density(alpha = 0.5)+facet_wrap(~Attrition)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

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

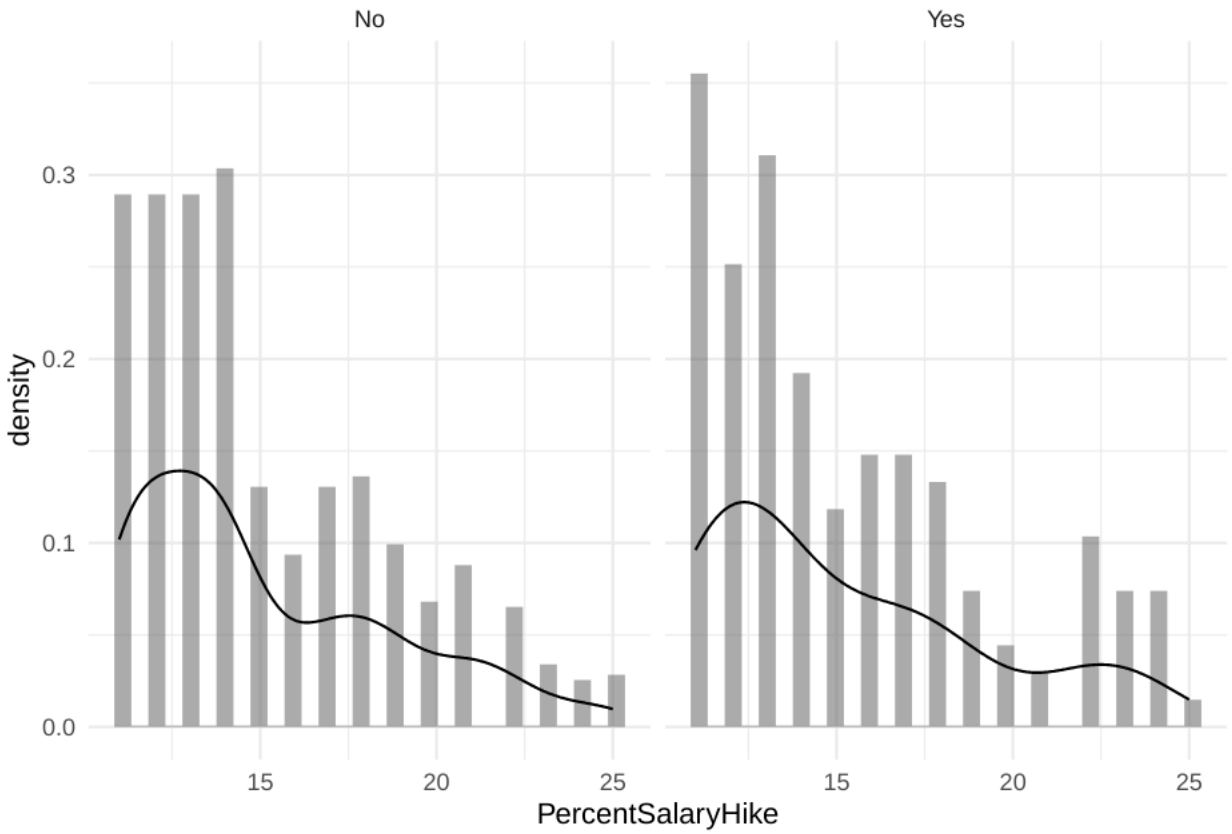


Again, no evidence of change in distribution

What about PercentSalaryHike?

```
ggplot(data, aes(x = PercentSalaryHike)) +
  geom_histogram(position = "dodge", alpha = 0.5, aes(y = ..density..)) +
  theme_minimal() +
  geom_density(alpha = 0.5)+facet_wrap(~Attrition)
```

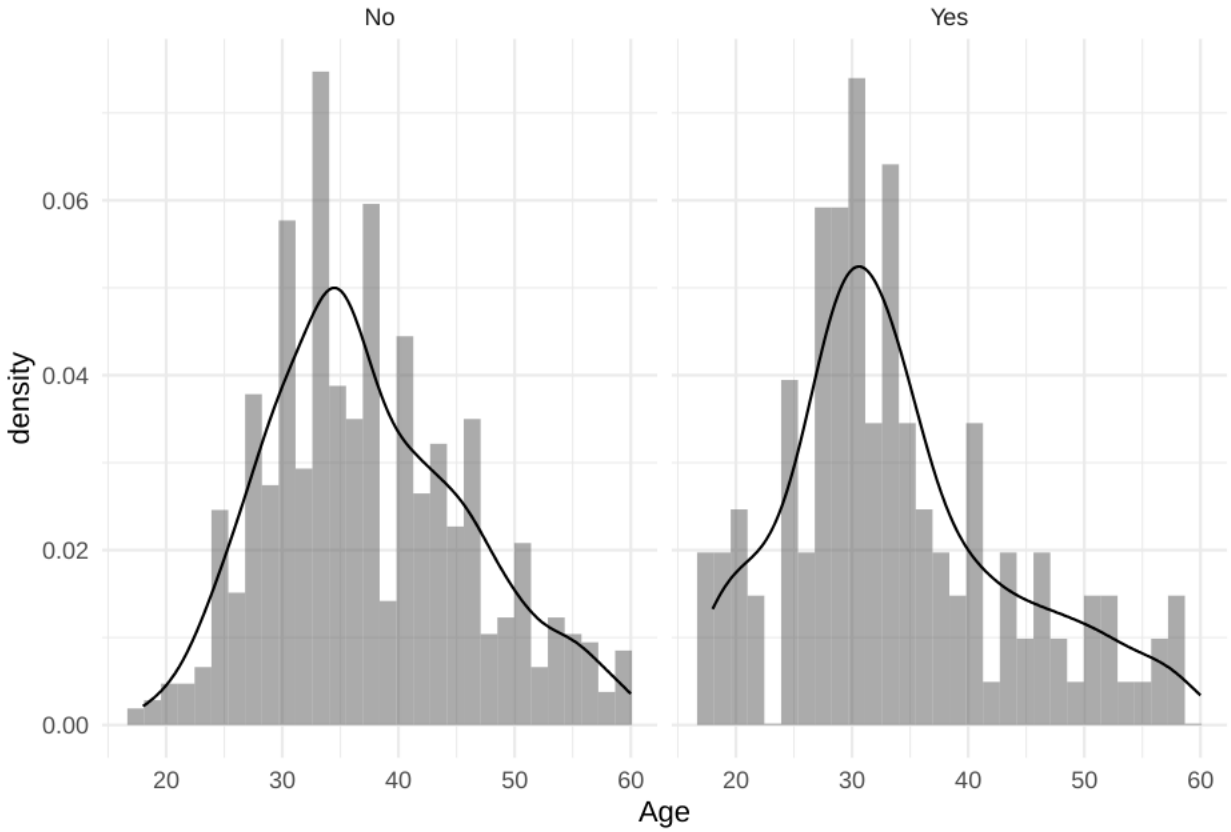
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



No evidence of change in raise distribution between those who quit and those who don't

What about Age?

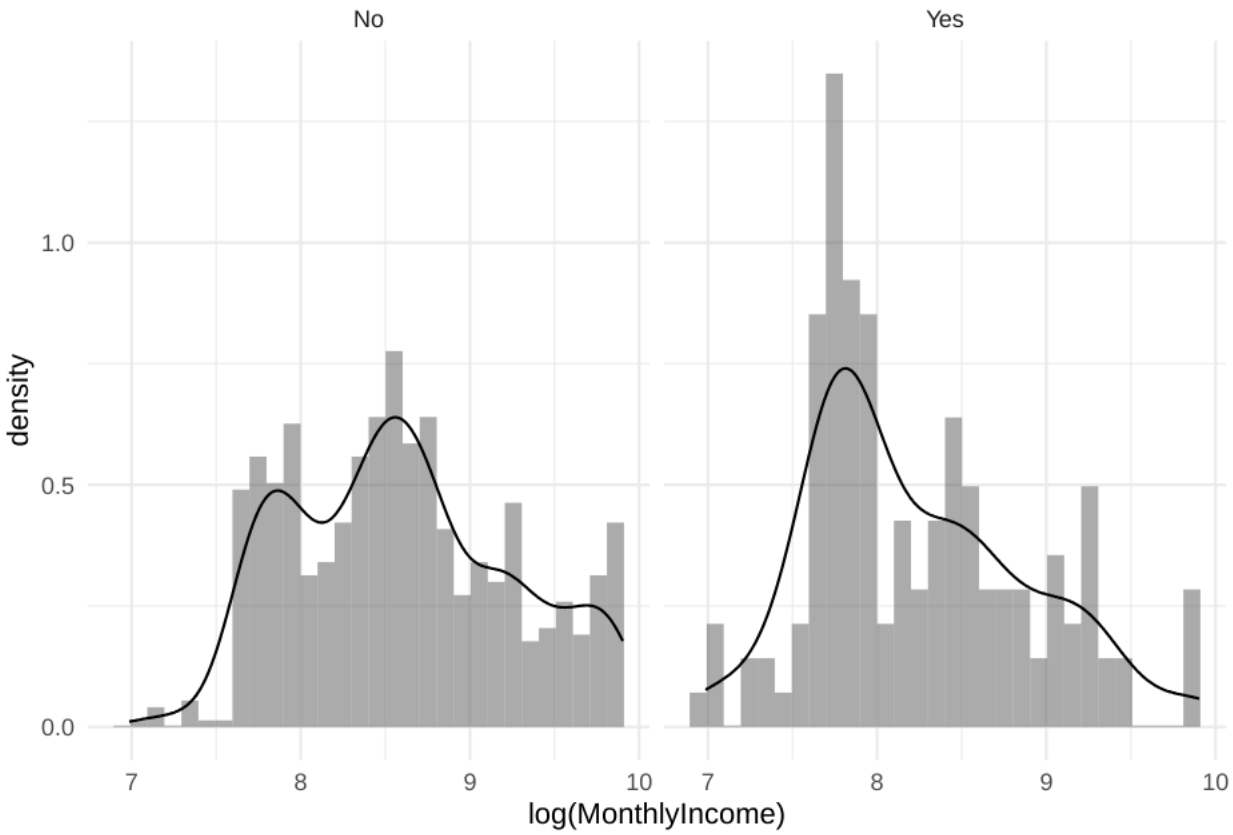
```
ggplot(data, aes(x = Age)) +
  geom_histogram(position = "dodge", alpha = 0.5, aes(y = ..density..)) +
  theme_minimal() +
  geom_density(alpha = 0.5)+facet_wrap(~Attrition)
```



No evidence of change in age distribution between groups

What about monthly income?

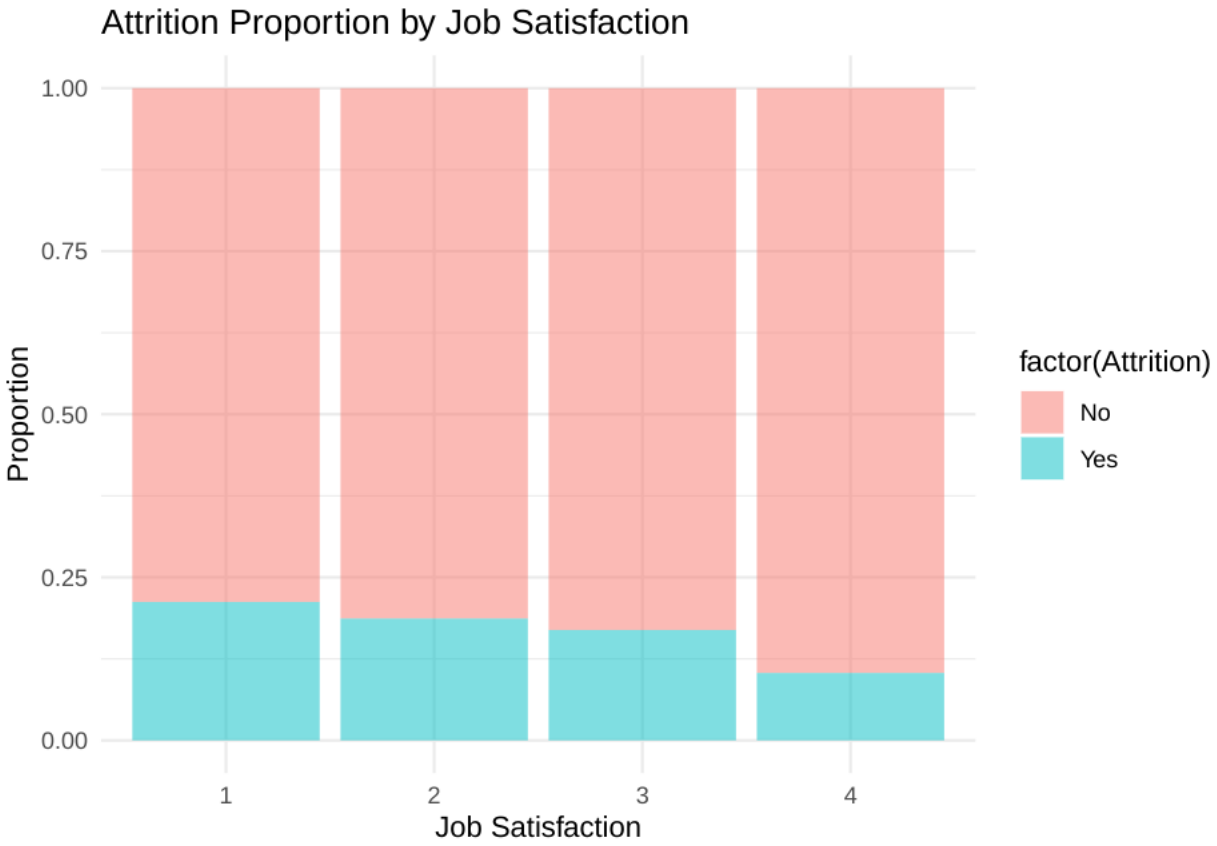
```
ggplot(data, aes(x = log(MonthlyIncome))) +
  geom_histogram(position = "dodge", alpha = 0.5, aes(y = ..density..)) +
  theme_minimal() +
  geom_density(alpha = 0.5)+facet_wrap(~Attrition)
```



Not much evidence of correlation between attrition and income

Job Satisfaction?

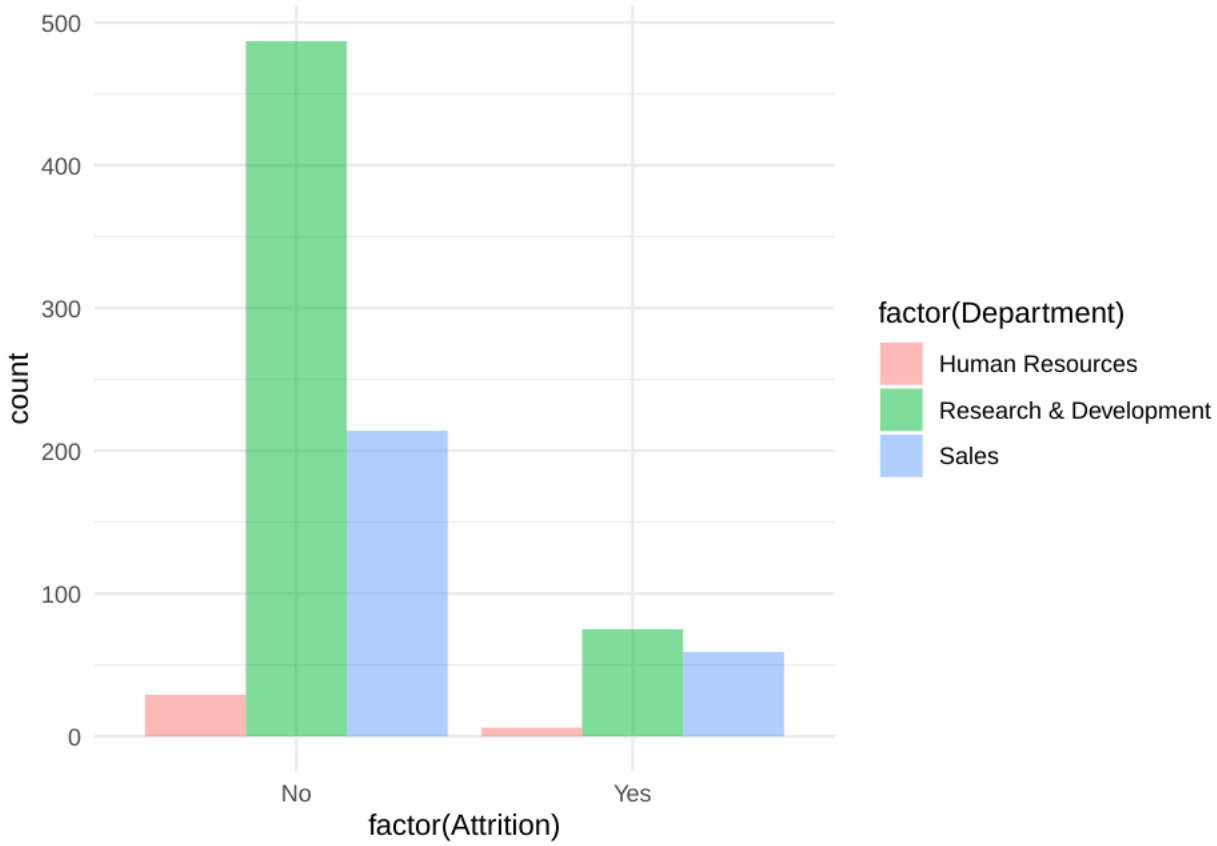
```
ggplot(data, aes(x = factor(JobSatisfaction), fill = factor(Attrition))) +
  geom_bar(position = "fill", alpha = 0.5) +
  labs(x = "Job Satisfaction", y = "Proportion", title = "Attrition Proportion by Job Satisfaction") +
  theme_minimal()
```



Almost looks like as Job Satisfaction goes up, the number of people who don't quit rises. There's also some evidence suggesting that as job satisfaction goes up, the number of people who do quit goes down.

Department?

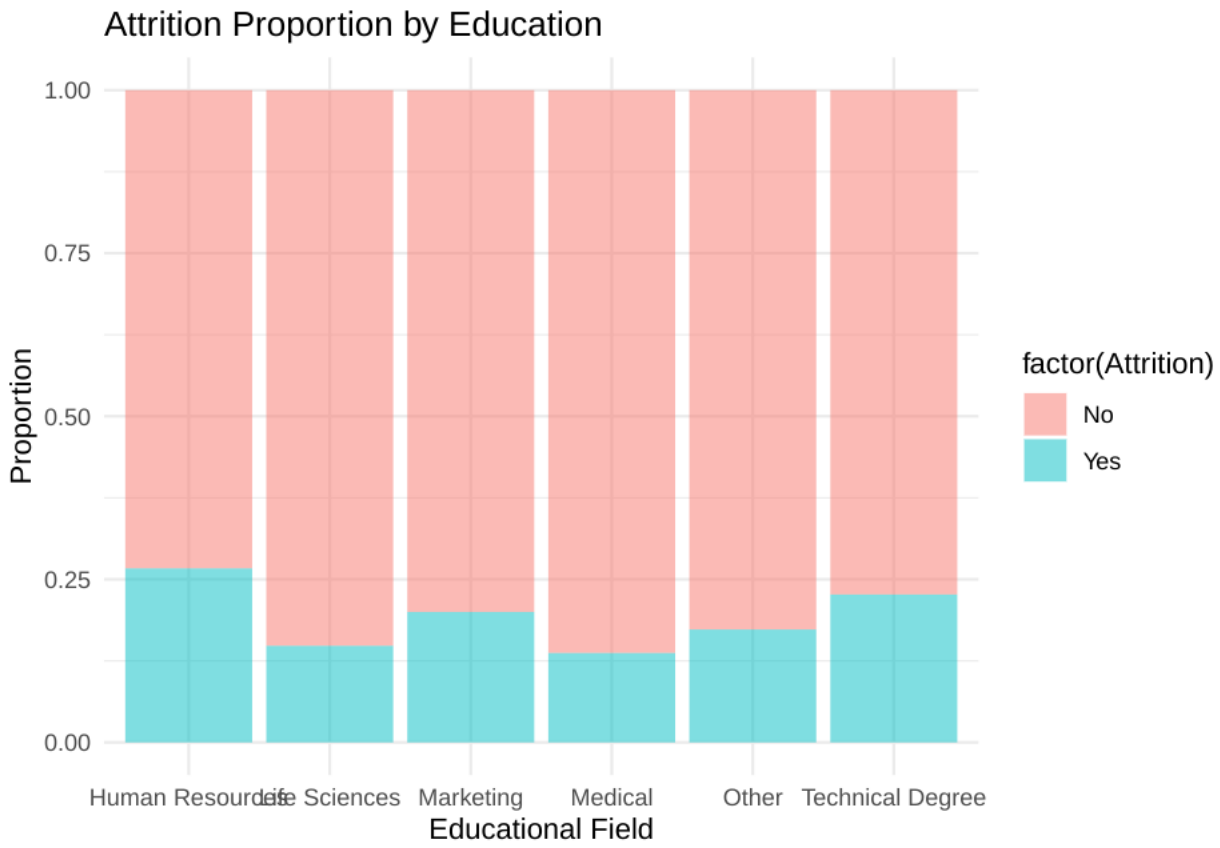
```
ggplot(data, aes(x = factor(Attrition), fill = factor(Department))) +
  geom_bar(position = "dodge", alpha = 0.5) +
  theme_minimal()
```



No clear correlation with Attrition

Education Field?

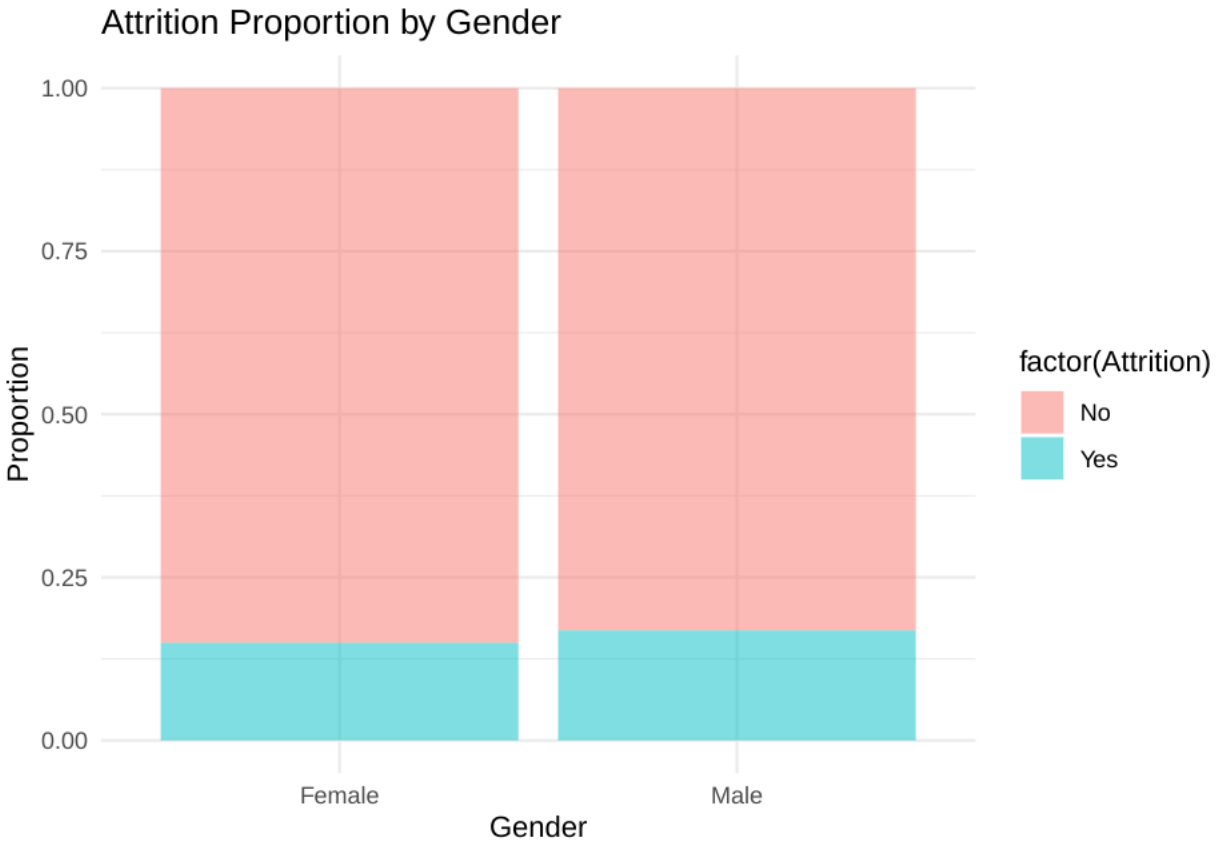
```
ggplot(data, aes(x = factor(EducationField), fill = factor(Attrition))) +
  geom_bar(position = "fill", alpha = 0.5) +
  labs(x = "Educational Field", y = "Proportion", title = "Attrition Proportion by Education") +
  theme_minimal()
```



Attrition ratio seems to change with educational field

Gender?

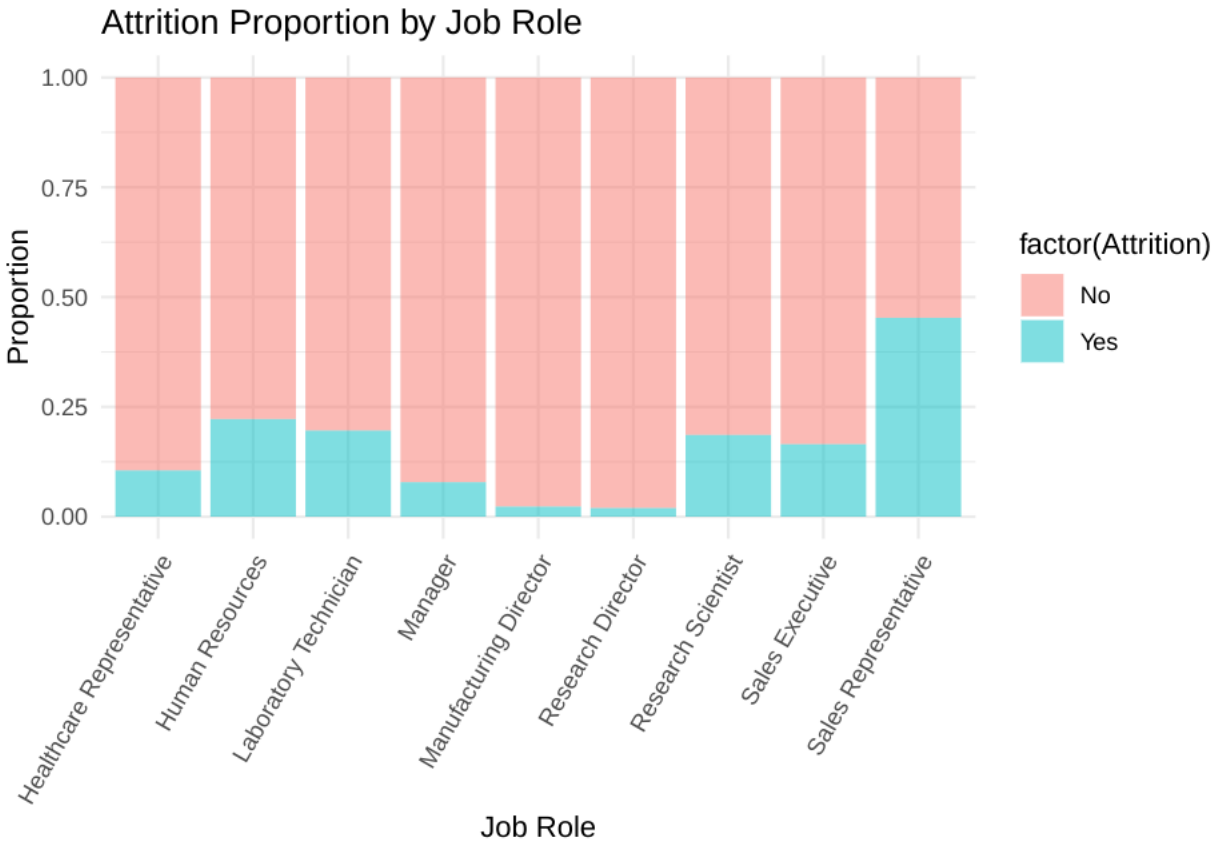
```
ggplot(data, aes(x = factor(Gender), fill = factor(Attrition))) +
  geom_bar(position = "fill", alpha = 0.5) +
  labs(x = "Gender", y = "Proportion", title = "Attrition Proportion by Gender") +
  theme_minimal()
```



Not much evidence of correlation between Attrition and Gender. Slightly higher proportion of attrition among males, but this does not appear significantly different.

Job Role?

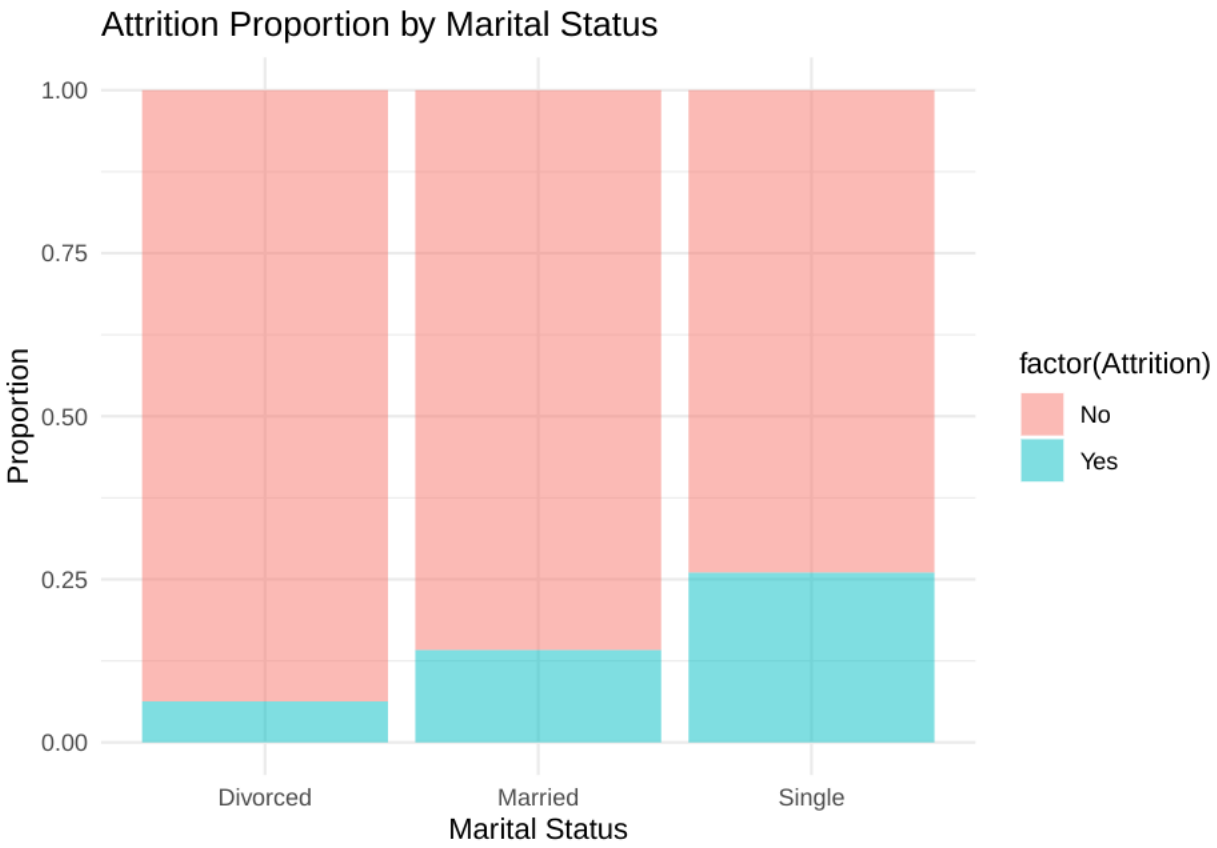
```
ggplot(data, aes(x = factor(JobRole), fill = factor(Attrition))) +
  geom_bar(position = "fill", alpha = 0.5) +
  labs(x = "Job Role", y = "Proportion", title = "Attrition Proportion by Job Role") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1))
```



There is consider variation in attrition frequency with respect to job role. In Sales Representative you have an almost 50-50 chance of quitting.

Marital Status?

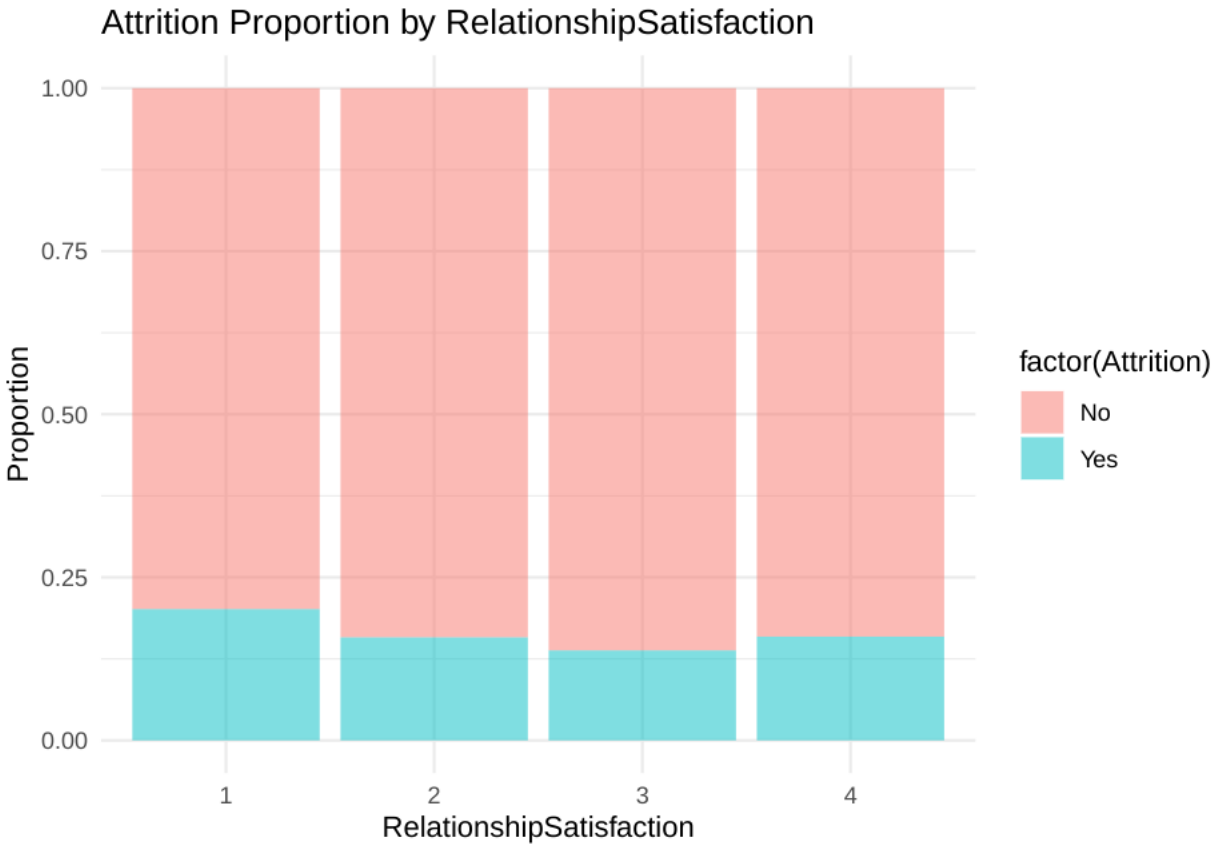
```
ggplot(data, aes(x = factor(MaritalStatus), fill = factor(Attrition))) +
  geom_bar(position = "fill", alpha = 0.5) +
  labs(x = "Marital Status", y = "Proportion", title = "Attrition Proportion by Marital Status") +
  theme_minimal()
```



Attrition proportion shows variation with MaritalStatus

Relationship Satisfaction?

```
ggplot(data, aes(x = factor(RelationshipSatisfaction), fill = factor(Attrition))) +
  geom_bar(position = "fill", alpha = 0.5) +
  labs(x = "RelationshipSatisfaction", y = "Proportion", title = "Attrition Proportion by RelationshipS
  theme_minimal()
```

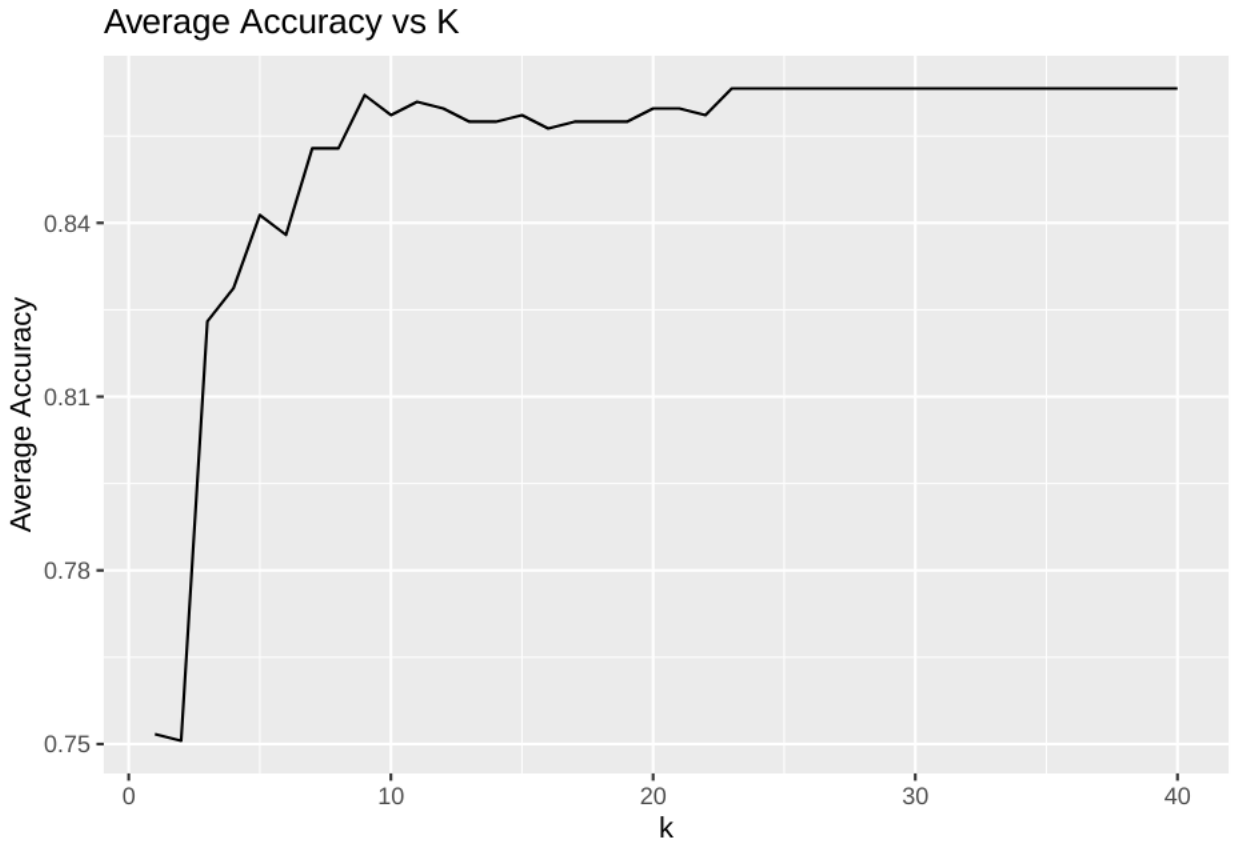


Some evidence of change in attrition with RelationshipSatisfaction (employee satisfaction with manager).

KNN Construction - K Selection

```
# Full Model




data3 <- referencedata %>%
  dummy_cols(select_columns = c("MaritalStatus", "EducationField", "BusinessTravel", "OverTime", "Depar
  select(-c("MaritalStatus", "Department", "JobSatisfaction", "EducationField", "Gender", "JobRole", "B


set.seed(4)
iterations <- 5
numks <- 40
percentsplit <- 0.8

# Identify the index of the 'Attrition' column
attrition_col <- which(names(data3) == "Attrition")
results_df <- data.frame()

for (j in 1:iterations) {
  trainIndices <- sample(1:nrow(data3), round(percentsplit * nrow(data3)))
  train <- data3[trainIndices, ]
  test <- data3[-trainIndices, ]
```

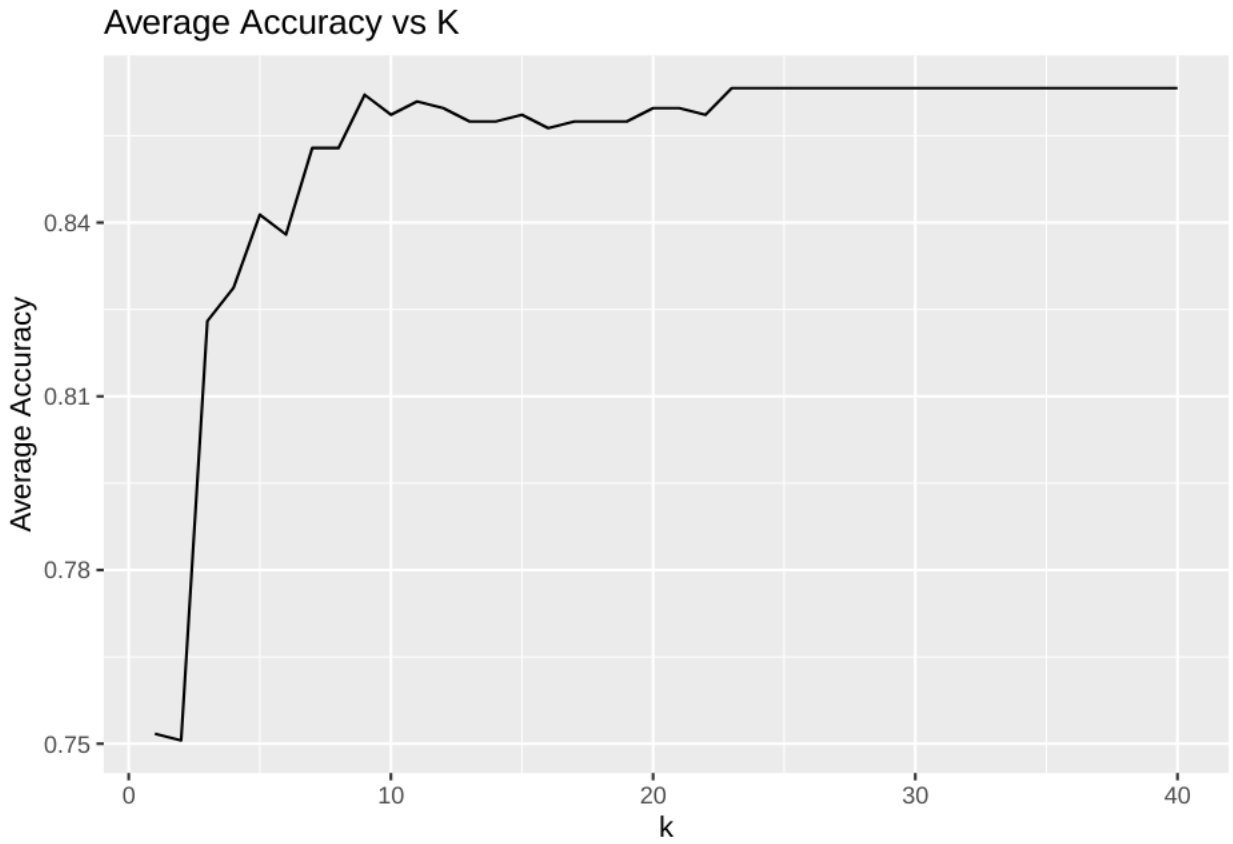
```r
  # Select all explanatory variables except 'Attrition'
  train_features <- train[, -attrition_col]
  test_features <- test[, -attrition_col]

  for (i in 1:numks) {
    classifications <- knn(train_features, test_features, train$Attrition, prob = TRUE, k = i)
    CM <- confusionMatrix(table(classifications, test$Attrition))

    # Append the accuracy and k value to the results data frame
    results_df <- rbind(results_df, data.frame(accuracy = CM$overall["Accuracy"], k = i))
  }
}


avg_accuracies <- results_df %>%
  group_by(k) %>%
  summarise(Average_Accuracy = mean(accuracy))

# Plotting average accuracies vs k
ggplot(avg_accuracies, aes(x = k, y = Average_Accuracy)) +
  geom_line() +
  labs(title = "Average Accuracy vs K", x = "k", y = "Average Accuracy")
```



Highest Accuracy at k=15

```r
# Reduced Model
```

```r
data4 <- referencedata %>%
  dummy_cols(select_columns = c("MaritalStatus", "EducationField", "Department", "Gender", "JobRole")) 
  select(-c("MaritalStatus", "Department", "JobSatisfaction", "EducationField", "Gender", "JobRole", "B


set.seed(4)
iterations <- 5
numks <- 40
percentsplit <- 0.8

# Identify the index of the 'Attrition' column
attrition_col <- which(names(data4) == "Attrition")
results_df <- data.frame()

for (j in 1:iterations) {
  trainIndices <- sample(1:nrow(data4), round(percentsplit * nrow(data4)))
  train <- data4[trainIndices, ]
  test <- data4[-trainIndices, ]

  # Select all explanatory variables except 'Attrition'
  train_features <- train[, -attrition_col]
  test_features <- test[, -attrition_col]

  for (i in 1:numks) {
    classifications <- knn(train_features, test_features, train$Attrition, prob = TRUE, k = i)
    CM <- confusionMatrix(table(classifications, test$Attrition))

    # Append the accuracy and k value to the results data frame
    results_df <- rbind(results_df, data.frame(accuracy = CM$overall["Accuracy"], k = i))
  }
}


avg_accuracies <- results_df %>%
  group_by(k) %>%
  summarise(Average_Accuracy = mean(accuracy))

# Plotting average accuracies vs k
ggplot(avg_accuracies, aes(x = k, y = Average_Accuracy)) +
  geom_line() +
  labs(title = "Average Accuracy vs K", x = "k", y = "Average Accuracy")
```



Reduced Model has best accuracy at k=9

Internal Cross Validation

```
# Full Model Internal Cross Validation


classifications<-knn.cv(data3[-2], data4$Attrition, k=15)

confusionMatrix(classifications, data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  725 135
##        Yes   5   5
##
##                Accuracy : 0.8391
##                  95% CI : (0.8129, 0.8629)
##     No Information Rate : 0.8391
##     P-Value [Acc > NIR] : 0.5225
##
##                   Kappa : 0.0462
##
##  Mcnemar's Test P-Value : <2e-16
```

```
##
##             Sensitivity : 0.99315
##             Specificity : 0.03571
##          Pos Pred Value : 0.84302
##          Neg Pred Value : 0.50000
##              Prevalence : 0.83908
##          Detection Rate : 0.83333
##    Detection Prevalence : 0.98851
##       Balanced Accuracy : 0.51443
##
##        'Positive' Class : No
##
```

```
# Reduced Model Internal Cross Validation

classifications<-knn.cv(data3[-2], data4$Attrition, k=9)

confusionMatrix(classifications, data$Attrition)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  724 133
##        Yes   6   7
##
##                Accuracy : 0.8402
##                  95% CI : (0.8142, 0.864)
##     No Information Rate : 0.8391
##     P-Value [Acc > NIR] : 0.4858
##
##                   Kappa : 0.066
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.9918
##             Specificity : 0.0500
##          Pos Pred Value : 0.8448
##          Neg Pred Value : 0.5385
##              Prevalence : 0.8391
##          Detection Rate : 0.8322
##    Detection Prevalence : 0.9851
##       Balanced Accuracy : 0.5209
##
##        'Positive' Class : No
##
```

Naive Bayes

```
# Full Model Naive Bayes Preidction Model
```

```r
set.seed(4)

iterations <- 100
masterAcc <- numeric(iterations)
meanSensitivity <- numeric(iterations)
meanSpecificity <- numeric(iterations)
meanAccuracy <- numeric(iterations)
splitPerc <- 0.8

for (j in 1:iterations) {
  trainIndices <- sample(1:dim(referencedata)[1], round(splitPerc * dim(referencedata)[1]))
  train <- referencedata[trainIndices, ]
  test <- referencedata[-trainIndices, ]

  model <- naiveBayes(Attrition ~ Age + BusinessTravel + DailyRate + Department + DistanceFromHome + Ed

    predictions <- predict(model, test[, c("Age", "BusinessTravel", "DailyRate", "Department", "Distanc


  # Calculate accuracy
  CM <- confusionMatrix(predictions, test$Attrition)
  masterAcc[j] <- CM$overall["Accuracy"]

  # Calculate sensitivity and specificity
  sensitivity <- CM$byClass["Sensitivity"]
  specificity <- CM$byClass["Specificity"]

  meanSensitivity[j] <- sensitivity
  meanSpecificity[j] <- specificity

  # Store mean accuracy
  meanAccuracy[j] <- CM$overall["Accuracy"]
}

# Calculate mean sensitivity, mean specificity, and mean accuracy across iterations
meanSensitivityOverall <- mean(meanSensitivity)
meanSpecificityOverall <- mean(meanSpecificity)
meanAccuracyOverall <- mean(meanAccuracy)

# Results
meanSensitivityOverall
```

```
## [1] 0.8958887
```

```r
meanSpecificityOverall
```

```
## [1] 0.5515502
```

```r
meanAccuracyOverall
```

```
## [1] 0.8391379
```

```r
# Reduced Model Naive Bayes

set.seed(4)
```

```r
iterations <- 100
masterAcc <- numeric(iterations)
meanSensitivity <- numeric(iterations)
meanSpecificity <- numeric(iterations)
meanAccuracy <- numeric(iterations)
splitPerc <- 0.8

for (j in 1:iterations) {
  trainIndices <- sample(1:dim(referencedata)[1], round(splitPerc * dim(referencedata)[1]))
  train <- referencedata[trainIndices, ]
  test <- referencedata[-trainIndices, ]

  model <- naiveBayes(Attrition ~EducationField+Department+Gender+JobRole, data = train)

    predictions <- predict(model, test[, c("EducationField", "Department", "Gender", "JobRole")], type =


  # Calculate accuracy
  CM <- confusionMatrix(predictions, test$Attrition)
  masterAcc[j] <- CM$overall["Accuracy"]

  # Calculate sensitivity and specificity
  sensitivity <- CM$byClass["Sensitivity"]
  specificity <- CM$byClass["Specificity"]

  meanSensitivity[j] <- sensitivity
  meanSpecificity[j] <- specificity

  # Store mean accuracy
  meanAccuracy[j] <- CM$overall["Accuracy"]
}

# Calculate mean sensitivity, mean specificity, and mean accuracy across iterations
meanSensitivityOverall <- mean(meanSensitivity)
meanSpecificityOverall <- mean(meanSpecificity)
meanAccuracyOverall <- mean(meanAccuracy)

# Results
meanSensitivityOverall
```

```
## [1] 0.9643062
```

```r
meanSpecificityOverall
```

```
## [1] 0.120365
```

```r
meanAccuracyOverall
```

```
## [1] 0.8254598
```

We will move forward making our Attrition predictions with the full Naive Bayes model

Predictions with Full Naive Bayes Model

```
CaseStudy2CompSet_No_Attrition <- read.csv("/cloud/project/CaseStudy2CompSet No Attrition.csv", header =

noattrition<-data.frame(CaseStudy2CompSet_No_Attrition)


noattrition$Attrition<-0


  model <- naiveBayes(Attrition ~ Age + BusinessTravel + DailyRate + Department + DistanceFromHome + Edu

    predictions <- predict(model, noattrition[, c("Age", "BusinessTravel", "DailyRate", "Department", "


noattrition$Attrition <- predictions

attritionpredictions <- noattrition %>% select(c("ID", "Attrition"))



write.csv(attritionpredictions, "CaseStudy2PredictionsLaskow_Attrition.csv", row.names = FALSE)

summary(attritionpredictions$Attrition)
```

```
##  No Yes
## 253  47
```

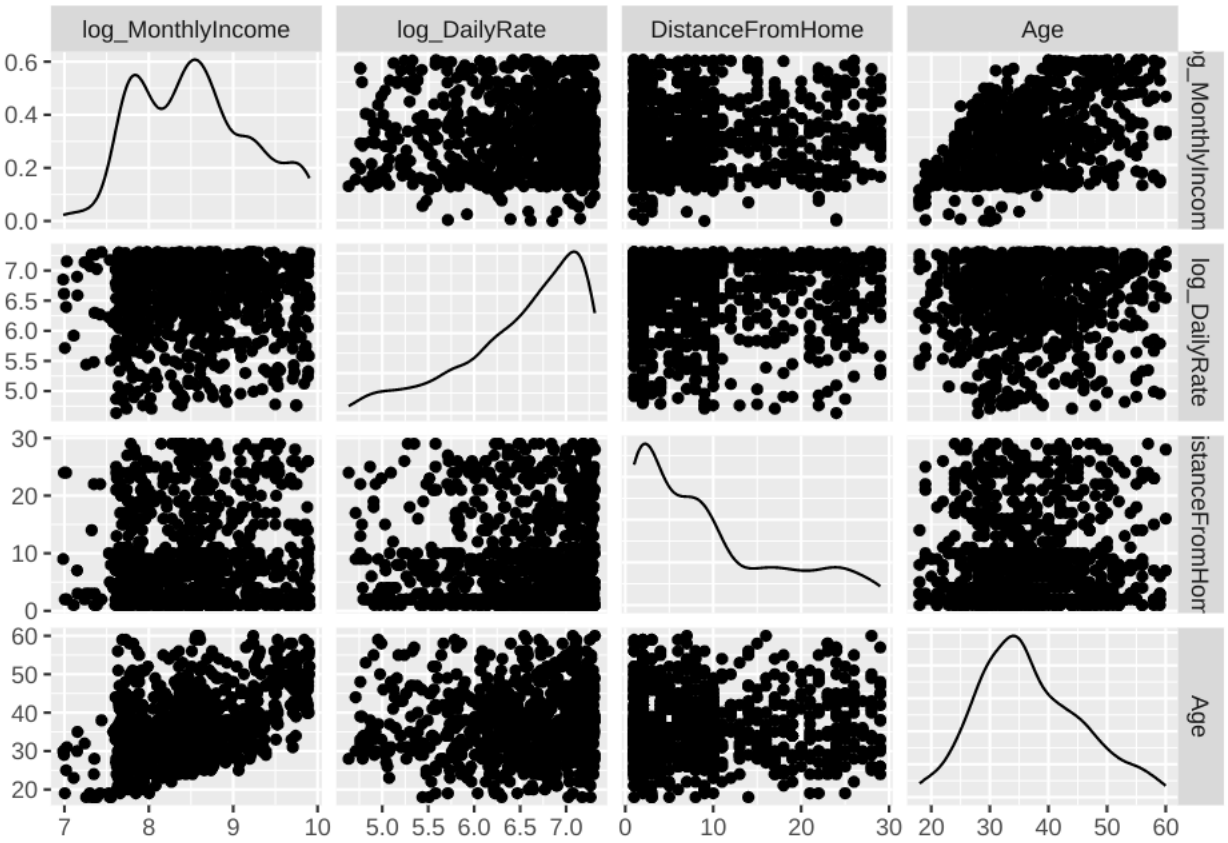
Predicted: 253 not quit, 47 quit

Part 2: Regression and Salary (MonthlyIncome)

```
# Bring in the reference dataset again


attritiondata_original<-read.table(textConnection(getURL(
  "https://s3.us-east-2.amazonaws.com/msds.ds.6306.2/CaseStudy2-data.csv"
)), sep=",", header=TRUE)

referencedata<-data.frame(attritiondata_original)

for (i in names(referencedata)) {
  if (class(referencedata[[i]]) == "character") {
    referencedata[[i]] <- factor(referencedata[[i]])
  }
}
```

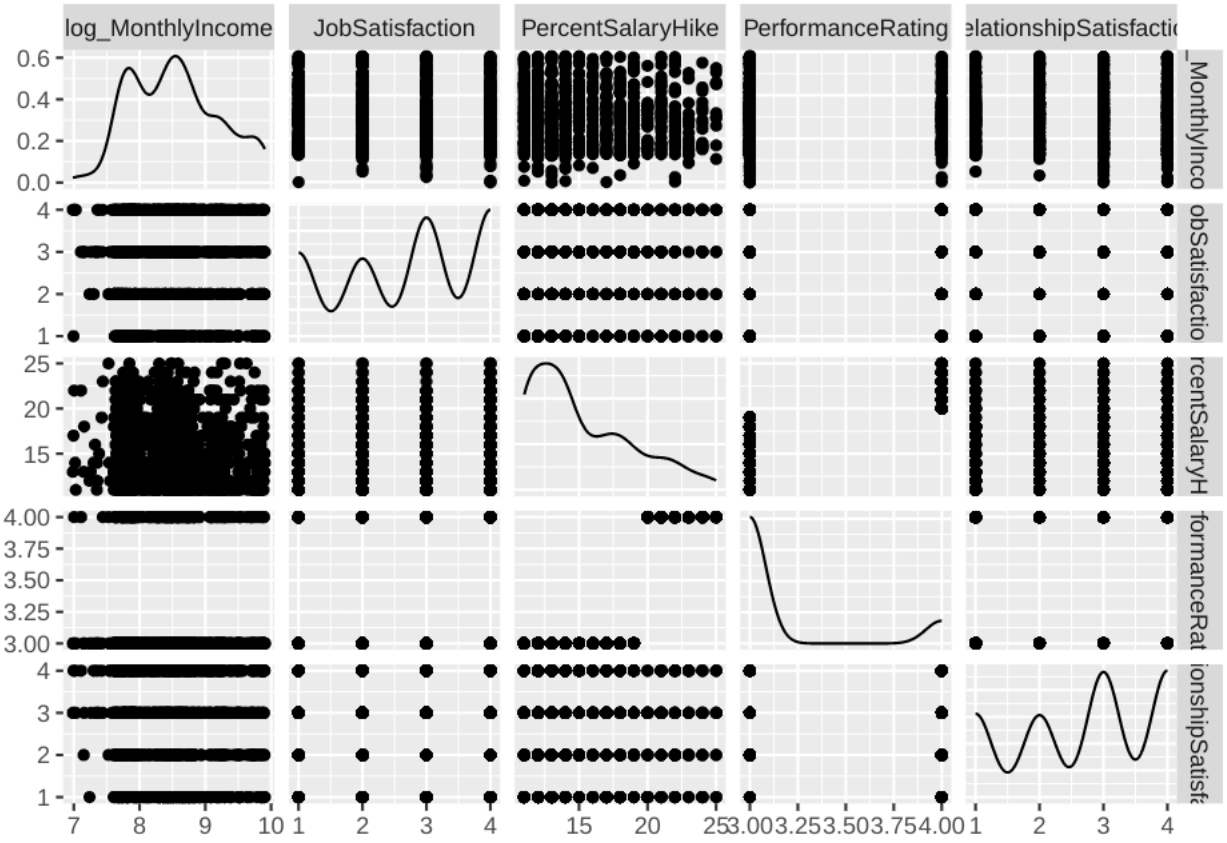
We need to start with some correlation matrices in order to visualize variables related to Salary (hereafter referred to as MonthlyIncome)

```
# Let's go through bit by bit and visualize the relationship to MonthlyRate. Numeric Variables first.


# Batch1:

referencedata$log_MonthlyIncome<-log(referencedata$MonthlyIncome)


referencedata$log_DailyRate<-log(referencedata$DailyRate)

referencedata$log_Age<-log(referencedata$Age)



numeric_columns1 <- c("log_MonthlyIncome", "log_DailyRate", "DistanceFromHome", "Age")
trainnew <- referencedata[, numeric_columns1]


# Visualizing correlation matrix

ggpairs(trainnew, upper = list(continuous = "points"))
```



Some evidence of Age relating to MonthlyIncome

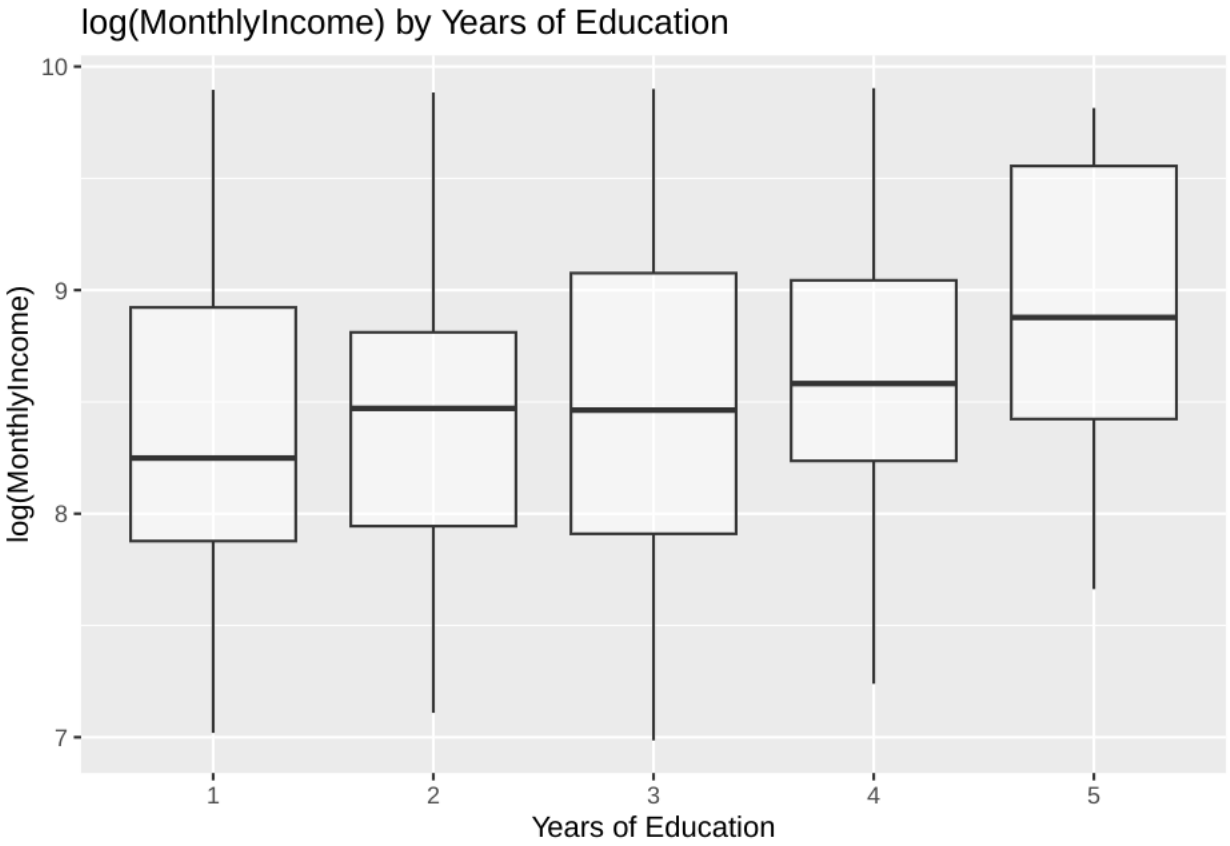
```
# Batch2:


referencedata$log_PSH<-log(referencedata$PercentSalaryHike)


numeric_columns1 <- c("log_MonthlyIncome", "JobSatisfaction", "PercentSalaryHike", "PerformanceRating",

trainnew <- referencedata[, numeric_columns1]


# Visualizing correlation matrix

ggpairs(trainnew, upper = list(continuous = "points"))
```



No clear relationship between any numeric variables and MonthlyIncome. Let's look a little closer at some of the variables we'd expect to be correlation just to be sure.

MonthlyIncome Distribution by Education (in years)

```
ggplot(referencedata, aes(x = factor(Education), y = log(MonthlyIncome))) +
  geom_boxplot(alpha = 0.5)+labs(title="log(MonthlyIncome) by Years of Education", x="Years of Educatio
```



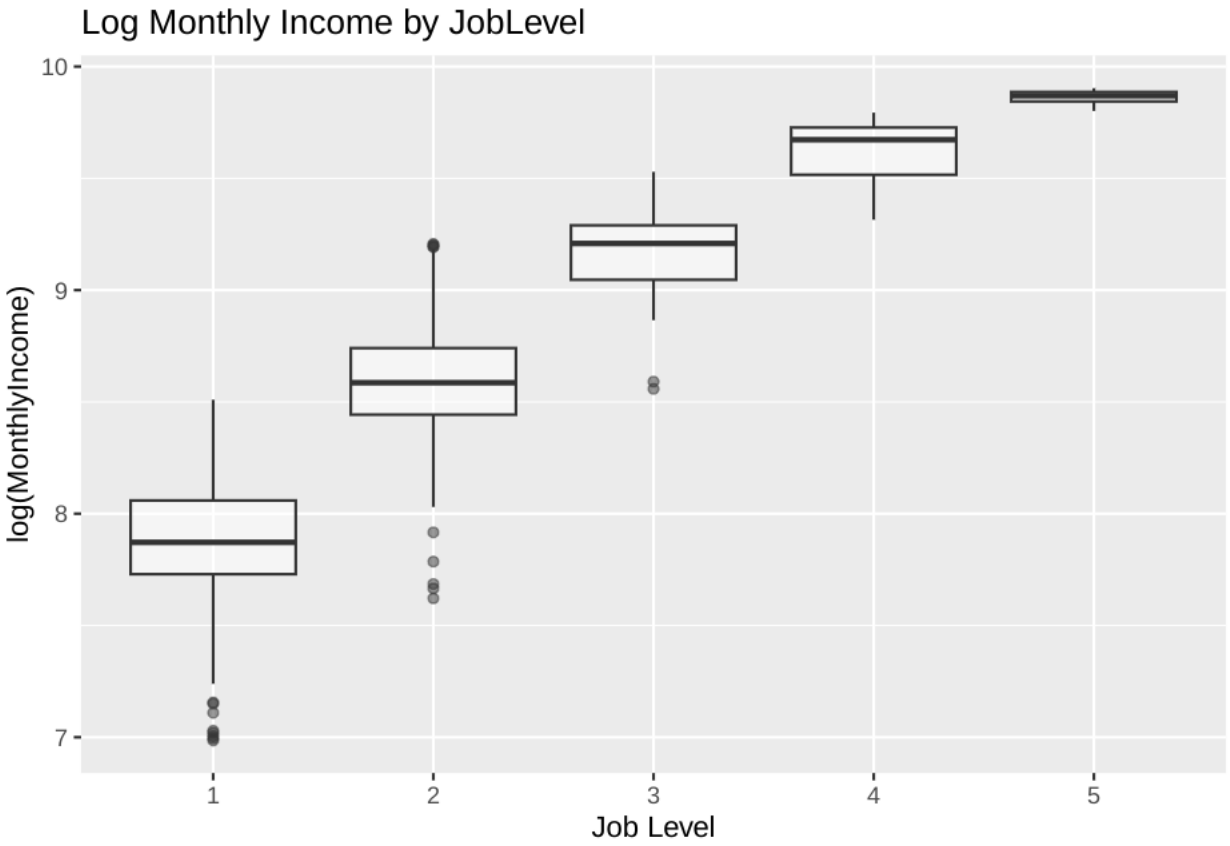

```
anova<-aov(log(MonthlyIncome)~factor(Education), data=referencedata)

summary(anova)
```

```
##                    Df Sum Sq Mean Sq F value   Pr(>F)
## factor(Education)   4    9.1  2.2843   5.324 0.000308 ***
## Residuals         865  371.2  0.4291
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see some evidence of mean log_MonthlyIncome changing with Education.

MonthlyRate Distribution by JobRole

```
ggplot(referencedata, aes(x = factor(JobLevel), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5)+labs(title="Log Monthly Income by JobLevel", x="Job Level", y="log(MonthlyI
```



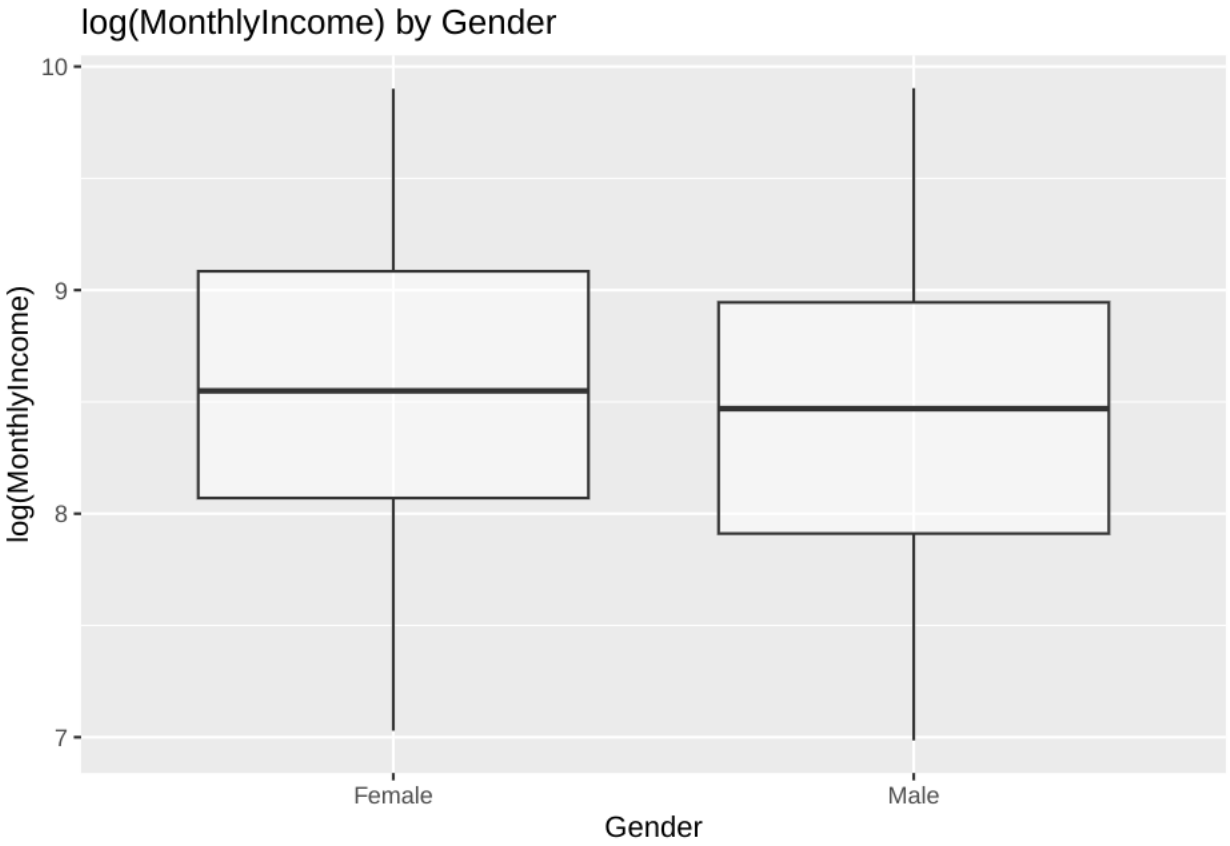

```
anova<-aov(log(MonthlyIncome)~factor(JobLevel), data=referencedata)

summary(anova)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## factor(JobLevel)   4  332.7   83.19    1513 <2e-16 ***
## Residuals        865   47.5    0.05
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Considerable evidence of correlation between JobLevel and log_MonthlyIncome

How about gender?

```
ggplot(referencedata, aes(x = factor(Gender), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5)+labs(title="log(MonthlyIncome) by Gender", x="Gender", y="log(MonthlyIncome
```



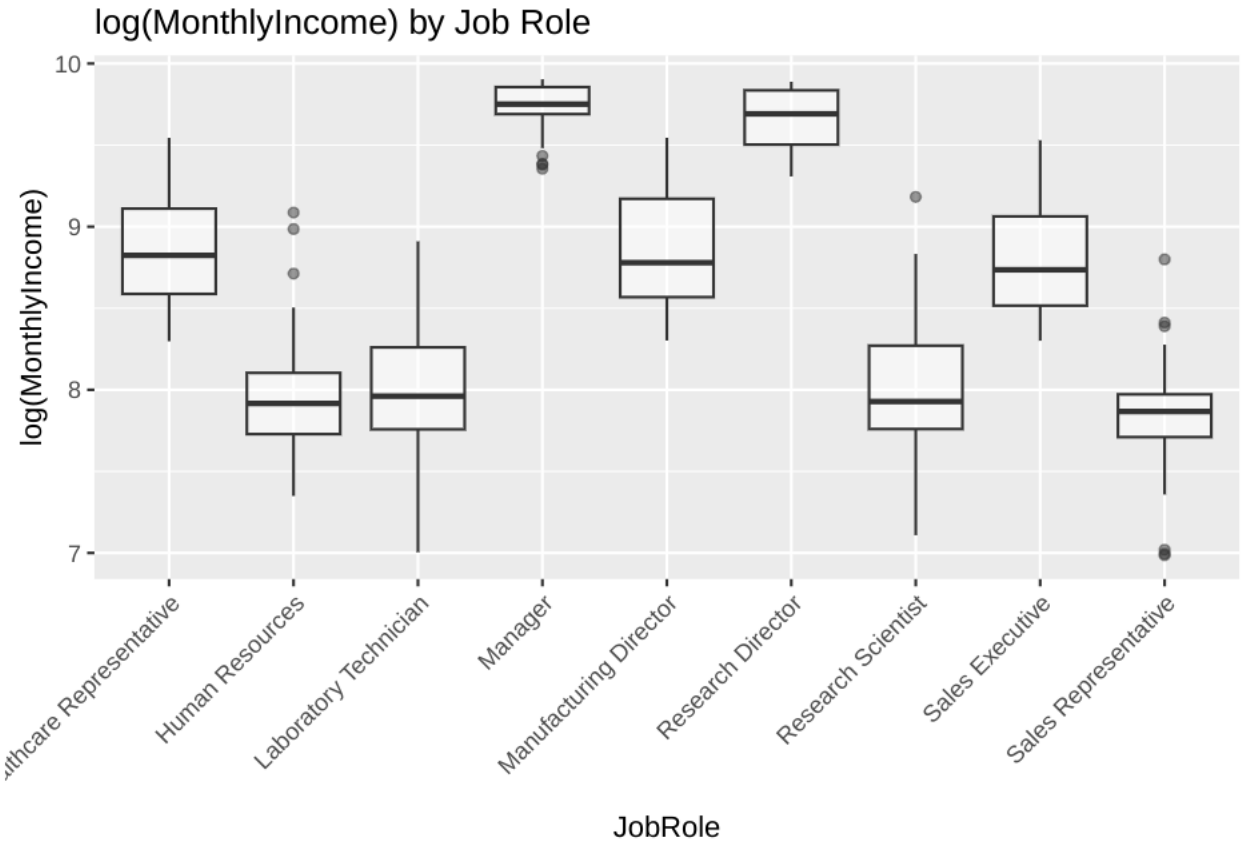

```
anova<-aov(log_MonthlyIncome~Gender, data=referencedata)

summary(anova)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Gender        1    1.9   1.852   4.247 0.0396 *
## Residuals   868  378.4   0.436
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There's some evidence that median MonthlyIncome differs with Gender. We might want to explore this later.

What about JobRole?

```
ggplot(referencedata, aes(x = factor(JobRole), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5) +
  labs(title = "log(MonthlyIncome) by Job Role", x = "JobRole", y = "log(MonthlyIncome)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



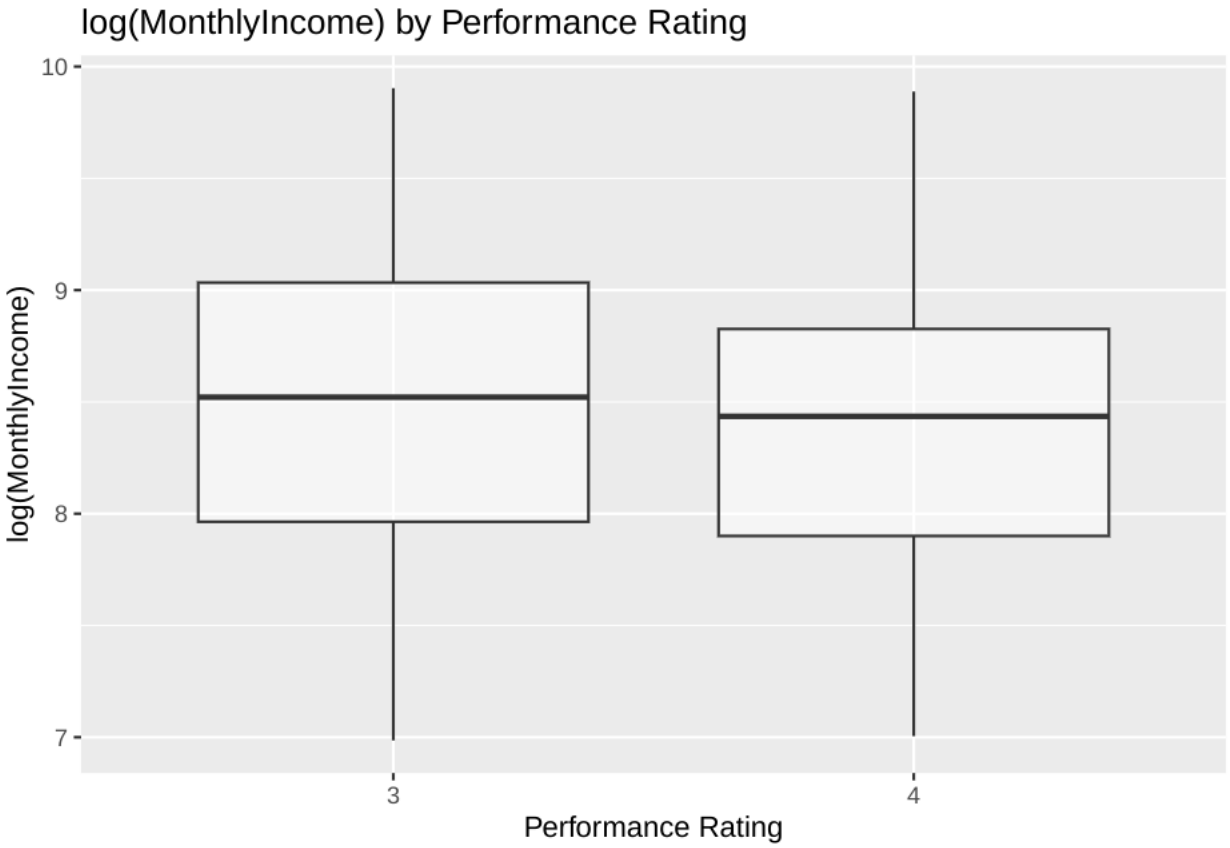

```
anova<-aov(log_MonthlyIncome~JobRole, data=referencedata)

summary(anova)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## JobRole       8 288.03   36.00     336 <2e-16 ***
## Residuals   861  92.26    0.11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here we see a wide range of MonthlyRate distributions. Definitely worth including.

What about performance rating?

```
ggplot(referencedata, aes(x = factor(PerformanceRating), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5)+
  labs(title = "log(MonthlyIncome) by Performance Rating", x = "Performance Rating", y = "log(MonthlyIn
```



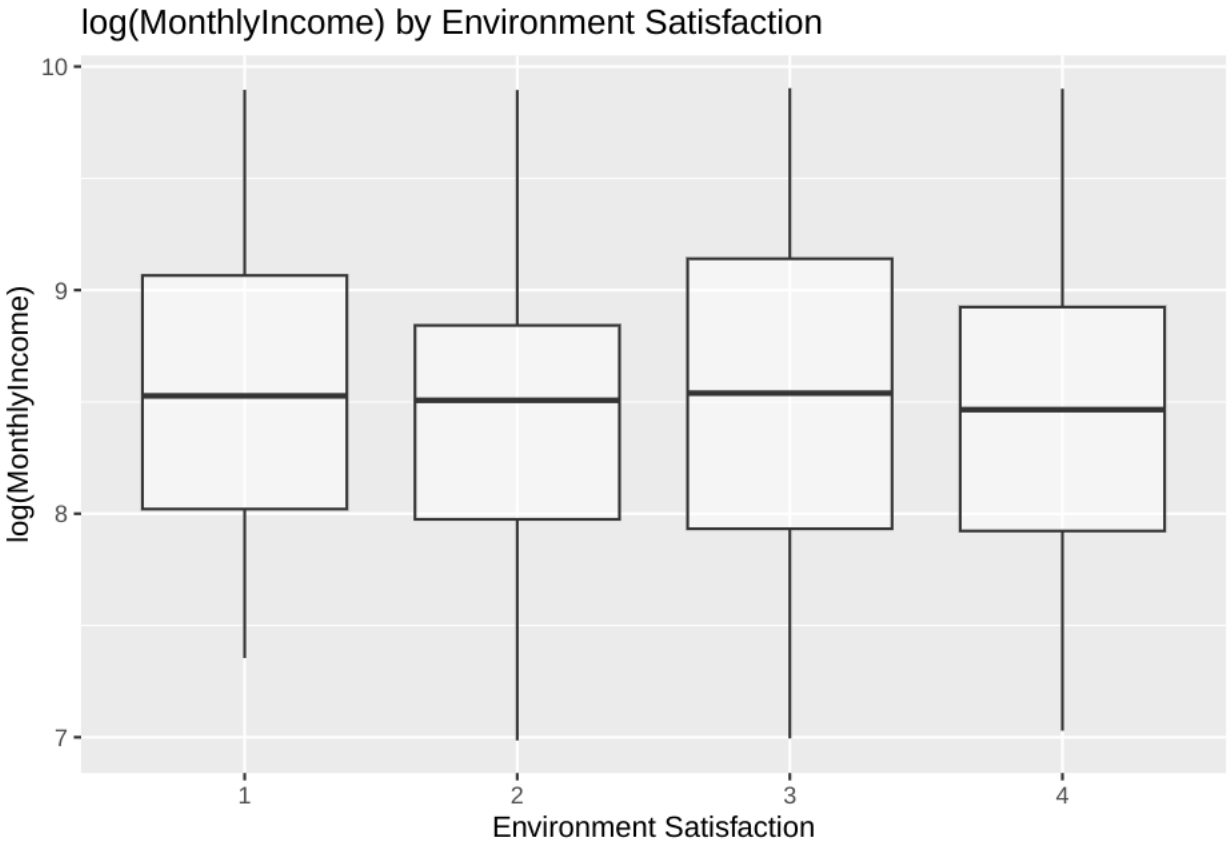

```
anova<-aov(log_MonthlyIncome~PerformanceRating, data=referencedata)

summary(anova)
```

```
##                     Df Sum Sq Mean Sq F value Pr(>F)
## PerformanceRating    1    0.7  0.7054   1.613  0.204
## Residuals          868  379.6  0.4373
```

No evidence of difference in median income with rating of 3 vs rating of 4

What about EnvironmentSatisfaction?

```
ggplot(referencedata, aes(x = factor(EnvironmentSatisfaction), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5)+
  labs(title = "log(MonthlyIncome) by Environment Satisfaction", x = "Environment Satisfaction", y = "l
```




```
anova<-aov(log_MonthlyIncome~factor(EnvironmentSatisfaction), data=referencedata)

summary(anova)
```

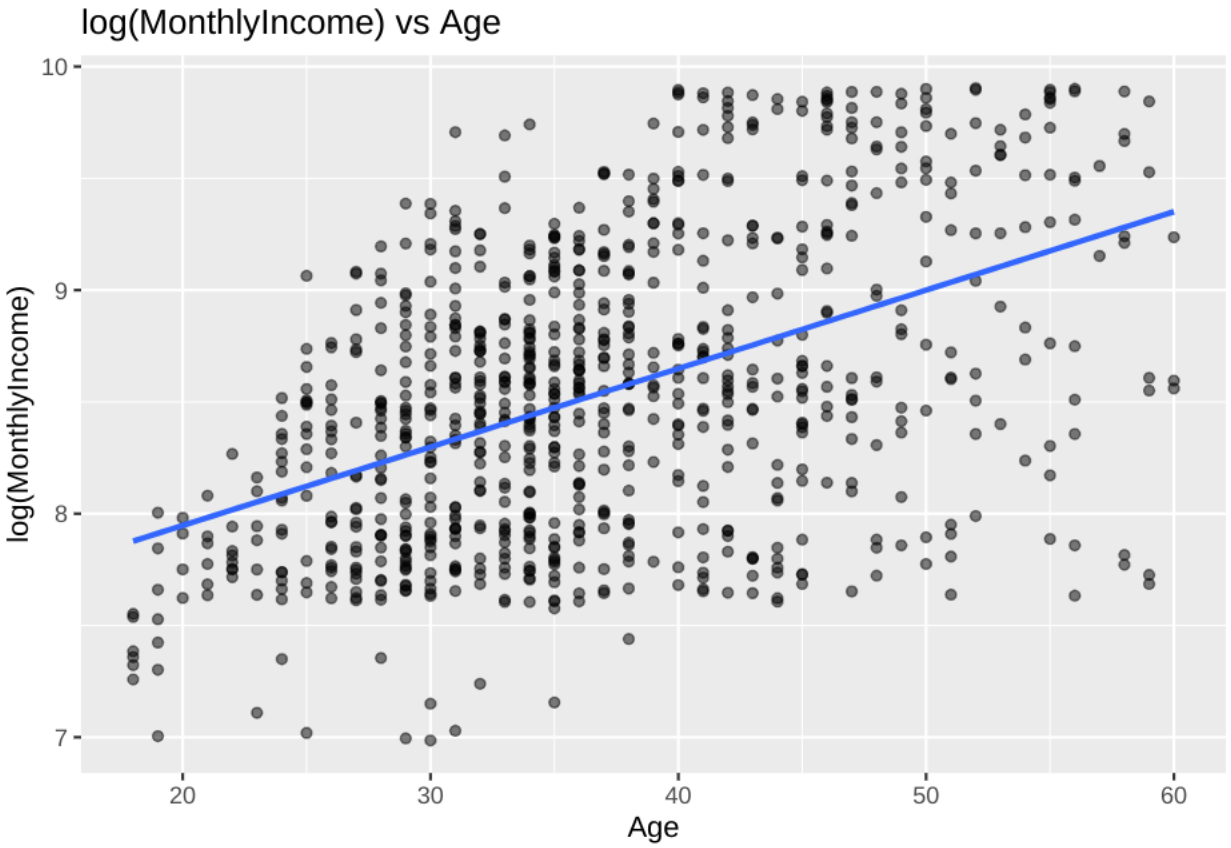
```
##                                Df Sum Sq Mean Sq F value Pr(>F)
## factor(EnvironmentSatisfaction)  3    0.7  0.2334   0.532   0.66
## Residuals                      866  379.6  0.4383
```

Not sufficient evidence to reject null that all median incomes are equal with respect to EnvironmentalSatisfaction

Age

```
ggplot(referencedata, aes(x = Age, y = log_MonthlyIncome)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE)+
  labs(title = "log(MonthlyIncome) vs Age", x = "Age", y = "log(MonthlyIncome)")
```

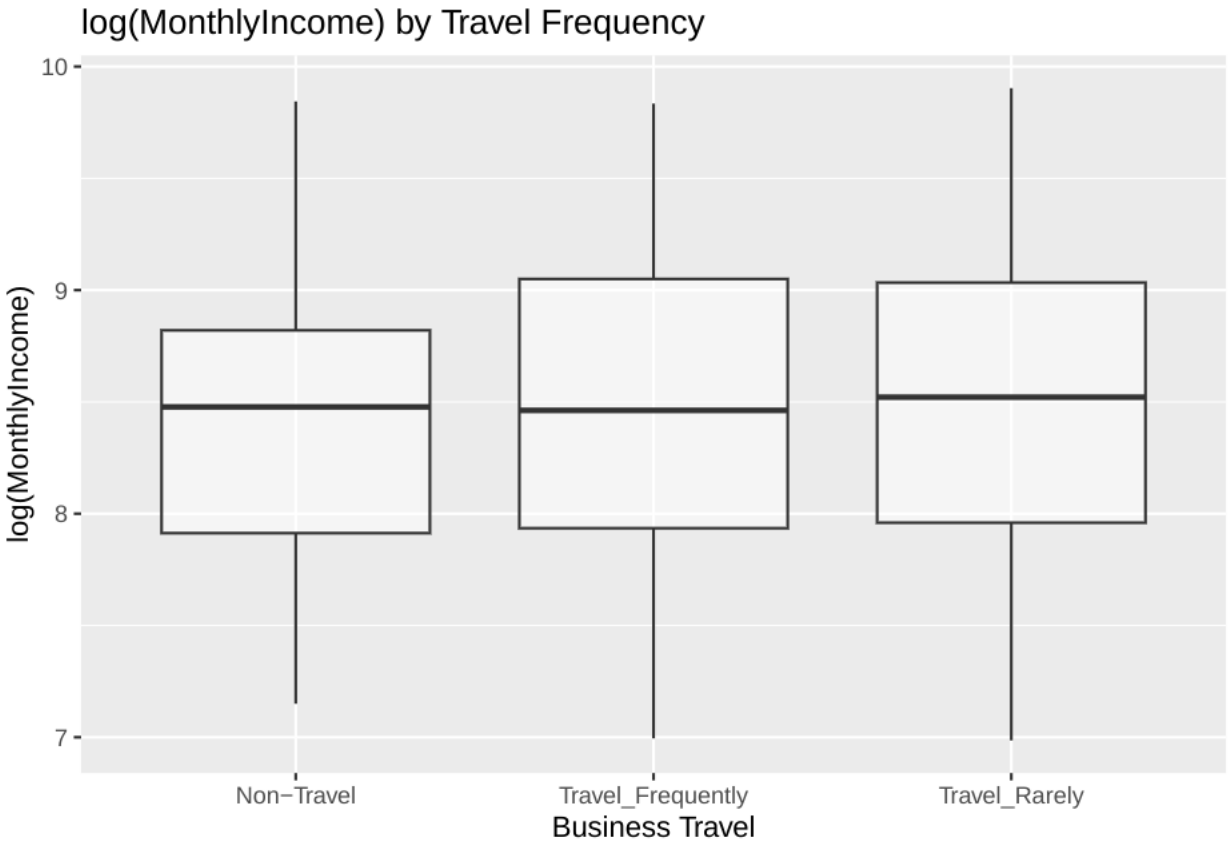
```
## `geom_smooth()` using formula = 'y ~ x'
```



Positive correlation between age and monthly income

BusinessTravel

```
ggplot(referencedata, aes(x = factor(BusinessTravel), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5)+
  labs(title = "log(MonthlyIncome) by Travel Frequency", x = "Business Travel", y = "log(MonthlyIncome)
```



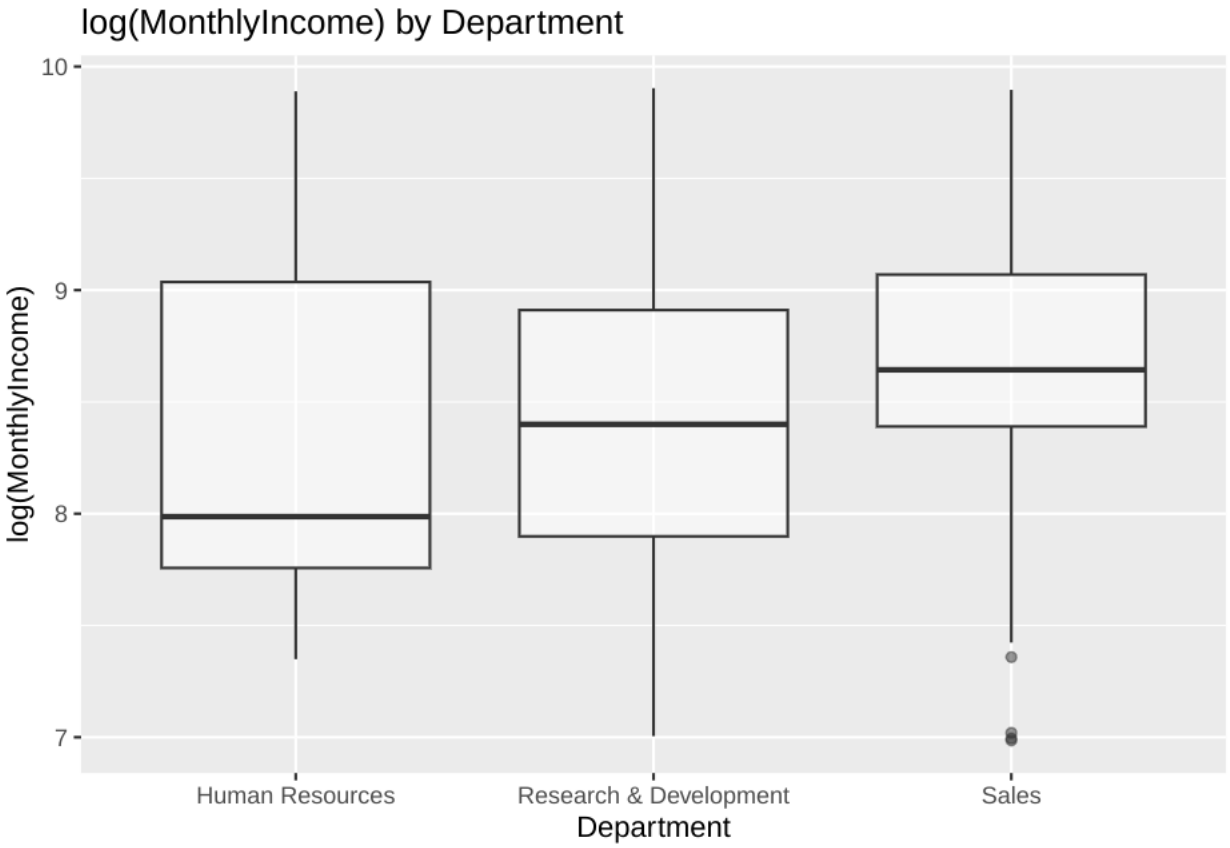

```
anova<-aov(log_MonthlyIncome~factor(BusinessTravel), data=referencedata)

summary(anova)
```

```
##                         Df Sum Sq Mean Sq F value Pr(>F)
## factor(BusinessTravel)   2    1.2  0.5878   1.344  0.261
## Residuals              867  379.1  0.4373
```

Not sufficient evidence of correlation between median MonthlyRate and BusinessTravel. We'll leave this out.

Department

```
ggplot(referencedata, aes(x = factor(Department), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5)+
  labs(title = "log(MonthlyIncome) by Department", x = "Department", y = "log(MonthlyIncome)")
```



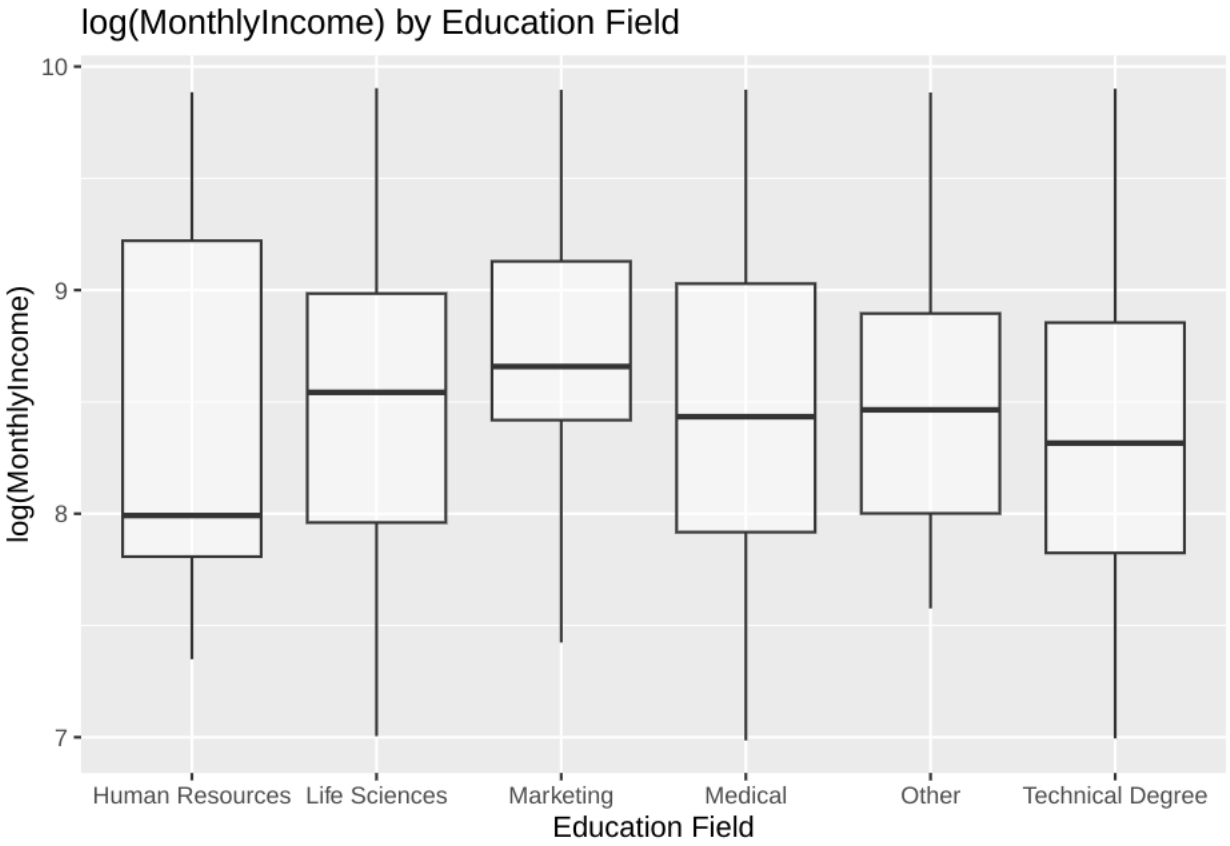

```
anova<-aov(log_MonthlyIncome~factor(Department), data=referencedata)

summary(anova)
```

```
##                      Df Sum Sq Mean Sq F value   Pr(>F)
## factor(Department)    2    7.0   3.478   8.078 0.000334 ***
## Residuals           867  373.3   0.431
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Strong evidence of difference in median MonthlyIncome by Department

EducationField

```
# BusinessTravel


ggplot(referencedata, aes(x = factor(EducationField), y = log_MonthlyIncome)) +
  geom_boxplot(alpha = 0.5)+
  labs(title = "log(MonthlyIncome) by Education Field", x = "Education Field", y = "log(MonthlyIncome)"
```




```
anova<-aov(log_MonthlyIncome~factor(EducationField), data=referencedata)

summary(anova)
```

```
##                          Df Sum Sq Mean Sq F value Pr(>F)
## factor(EducationField)    5    5.1  1.0246    2.36 0.0386 *
## Residuals               864  375.2  0.4342
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Evidence that median MonthlyIncome changes with EducationField

Final Variables to Use:

Age Gender Education JobRole JobLevel Department EducationField

K-Fold Cross Validation of Regression Model

```
# Cross-Validation Method using trainControl and train functions in Caret package
ctrl<-trainControl(method="cv", number=5)

# Full Model

# Cross-Validation Method using trainControl and train functions in Caret package
ctrl<-trainControl(method="cv", number=5)


# Regression Model

modelreduced <- train(MonthlyIncome ~ Age + Gender + Education + JobRole + JobLevel + Department + Educ
```

```
               data = referencedata,
               method = "lm",
               trControl = ctrl)


summary(modelreduced)
```

```
##
## Call:
## lm(formula = .outcome ~ ., data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3530.3  -659.7   -25.1   648.8  4145.7
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       -698.683    562.644  -1.242  0.21466
## Age                                 14.257      4.844   2.943  0.00334 **
## GenderMale                         114.585     75.318   1.521  0.12855
## Education                          -32.638     37.444  -0.872  0.38365
## `JobRoleHuman Resources`          -272.480    521.111  -0.523  0.60119
## `JobRoleLaboratory Technician`    -538.170    172.301  -3.123  0.00185 **
## JobRoleManager                    4140.452    285.109  14.522  < 2e-16 ***
## `JobRoleManufacturing Director`     85.077    170.271   0.500  0.61745
## `JobRoleResearch Director`        3960.652    219.013  18.084  < 2e-16 ***
## `JobRoleResearch Scientist`       -253.057    172.189  -1.470  0.14203
## `JobRoleSales Executive`           248.204    362.191   0.685  0.49335
## `JobRoleSales Representative`      -38.207    395.340  -0.097  0.92303
## JobLevel                          3032.087     70.020  43.303  < 2e-16 ***
## `DepartmentResearch & Development`  23.163    482.776   0.048  0.96174
## DepartmentSales                   -351.012    493.991  -0.711  0.47755
## `EducationFieldLife Sciences`      159.364    373.848   0.426  0.67001
## EducationFieldMarketing            111.726    395.986   0.282  0.77790
## EducationFieldMedical               77.864    374.565   0.208  0.83537
## EducationFieldOther                 95.246    399.708   0.238  0.81171
## `EducationFieldTechnical Degree`   113.989    390.109   0.292  0.77021
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1082 on 850 degrees of freedom
## Multiple R-squared:  0.9458, Adjusted R-squared:  0.9446
## F-statistic:   781 on 19 and 850 DF,  p-value: < 2.2e-16
```

```
# Let's build a test and training set to see the RMSE and MSPE of the full model

set.seed(3)
mspe_values <- numeric(100)
rmse_values <- numeric(100)

for (i in 1:100) {
  TrainObs <- sample(seq(1, dim(referencedata)[1]), round(0.75 * dim(referencedata)[1]), replace = FALS

  trainset <- referencedata[TrainObs, ]
```

```
  testset <- referencedata[-TrainObs, ]

  fullmodel <- lm(MonthlyIncome ~ Age + Gender + Education + JobRole + JobLevel + Department + Educatio
                  data = trainset)

  fullmodel_preds <- predict(fullmodel, newdata = testset)

  # Calculate MSPE for this iteration
  mspe_values[i] <- mean((testset$MonthlyIncome - fullmodel_preds)^2)

  # Calculate RMSE for this iteration
  rmse_values[i] <- sqrt(mean((testset$MonthlyIncome - fullmodel_preds)^2))
}

# Calculate average MSPE and RMSE over 20 iterations
average_mspe <- mean(mspe_values)
average_rmse <- mean(rmse_values)
average_mspe
```

```
## [1] 1218214
```

```
average_rmse
```

```
## [1] 1102.52
```

```
# Reduced model



mspe_values <- numeric(100)
rmse_values <- numeric(100)

for (i in 1:100) {
  TrainObs <- sample(seq(1, dim(referencedata)[1]), round(0.75 * dim(referencedata)[1]), replace = FALS

  trainset <- referencedata[TrainObs, ]
  testset <- referencedata[-TrainObs, ]

   modelreduced <- lm(MonthlyIncome ~ Age + Gender + Education + JobRole + JobLevel + Department + Educ
               data = trainset)

  fullmodel_preds <- predict(fullmodel, newdata = testset)

  # Calculate MSPE for this iteration
  mspe_values[i] <- mean((testset$MonthlyIncome - fullmodel_preds)^2)

  # Calculate RMSE for this iteration
  rmse_values[i] <- sqrt(mean((testset$MonthlyIncome - fullmodel_preds)^2))
}

# Calculate average MSPE and RMSE over 20 iterations
average_mspe <- mean(mspe_values)
average_rmse <- mean(rmse_values)
average_mspe
```

```
## [1] 1127346
```

```
average_rmse
```

```
## [1] 1060.646
```

Reduced model has a lower MSPE by about 73000. Adjusted R Squares are close, as are the RMSE values. We will move forward with the reduced model.

MonthlyIncome Predictions

```
nosalaries$MonthlyIncome<-NA

model<-lm(MonthlyIncome ~ Age + factor(Gender) + Education + factor(JobRole) + JobLevel + factor(Departr



predictions<-predict(model, newdata=nosalaries)

nosalaries$MonthlyIncome<-predictions

salarypredictions<-nosalaries%>%select(c("ID", "MonthlyIncome"))

write.csv(salarypredictions,"Case2PredictionsLaskow_Salaries.csv", row.names=TRUE)
```

```
head(salarypredictions)
```

```
##    ID MonthlyIncome
## 1 871      6019.149
## 2 872      2700.731
## 3 873     12206.774
## 4 874      2475.061
## 5 875      2315.264
## 6 876      5662.971
```

```
summary(salarypredictions$MonthlyIncome)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2181    2621    5773    6217    6277   19542
```

Lastly, an RShiny app is available at the following link:

https://886kdw-joel.shinyapps.io/Employee_Demographics/

This app enables user to explore age distribution by department.