Project

2023-04-29

#Loading and reading the cleaned csv dataset from the specified directory

```
data<- read.csv("C:/Users/vibar/Downloads/Drug_clean.csv")

#Numerically encoding the Type of Drug column
data$Status <- ifelse(data$Type == "RX", 0,
               ifelse(data$Type %in% c("OTC"), 1, 2))

head(data)
```

```
##                  Condition                        Drug EaseOfUse Effective
## 1 Acute Bacterial Sinusitis                 Amoxicillin  3.852353  3.655882
## 2 Acute Bacterial Sinusitis Amoxicillin-Pot Clavulanate  3.470000  3.290000
## 3 Acute Bacterial Sinusitis Amoxicillin-Pot Clavulanate  3.121429  2.962857
## 4 Acute Bacterial Sinusitis                  Ampicillin  2.000000  3.000000
## 5 Acute Bacterial Sinusitis                  Ampicillin  3.250000  3.000000
## 6 Acute Bacterial Sinusitis           Ampicillin Sodium  3.000000  3.000000
##             Form Indication     Price   Reviews Satisfaction Type Status
## 1        Capsule   On Label  12.59000  86.29412     3.197647   RX      0
## 2 Liquid (Drink)  Off Label 287.37000  43.00000     2.590000   RX      0
## 3         Tablet   On Label  70.60857 267.28571     2.248571   RX      0
## 4        Capsule   On Label  12.59000   1.00000     1.000000   RX      0
## 5         Tablet   On Label 125.24000  15.00000     3.000000   RX      0
## 6         Tablet  Off Label 143.21500   1.00000     3.000000   RX      0
```

#Checking for Missing and Duplicate Values

```
#Missing Values
if (any(is.na(data))) {
  print("There are missing values in the dataset.")
} else {
  print("There are no missing values in the dataset.")
}
```

```
## [1] "There are no missing values in the dataset."
```

```
#For Duplicates
if (any(duplicated(data))) {
  print("There are duplicate rows in the dataset.")
} else {
  print("There are no duplicate rows in the dataset.")
}
```

```
## [1] "There are no duplicate rows in the dataset."
```

#Descriptive Statistics For both the continuous variables "Effective and"Satisfaction"

```r
# Calculate mean
mean1<- mean(data$Effective)
cat("The mean of the Effectiveness variable is:", mean1, "\n")
```

```
## The mean of the Effectiveness variable is: 3.52563
```

```r
mean2<- mean(data$Satisfaction)
cat("The mean of the Satisfaction variable is:", mean2, "\n")
```

```
## The mean of the Satisfaction variable is: 3.192844
```

```r
# Calculate median
median1<- median(data$Effective)
cat("The median of the Effectiveness variable is:", median1, "\n")
```

```
## The median of the Effectiveness variable is: 3.6
```

```r
median2<- median(data$Satisfaction)
cat("The median of the Satisfaction variable is:", median2, "\n")
```

```
## The median of the Satisfaction variable is: 3.2
```

```r
# Calculate range
range1<- range(data$Effective)
cat("The range of the Effectiveness variable is:", range1, "\n")
```

```
## The range of the Effectiveness variable is: 1 5
```

```r
range2<- range(data$Satisfaction)
cat("The range of the Satisfaction variable is:", range2, "\n")
```

```
## The range of the Satisfaction variable is: 1 5
```

```r
# Calculate standard deviation
sd1<- sd(data$Effective)
cat("The sd of the Effectiveness variable is:", sd1, "\n")
```

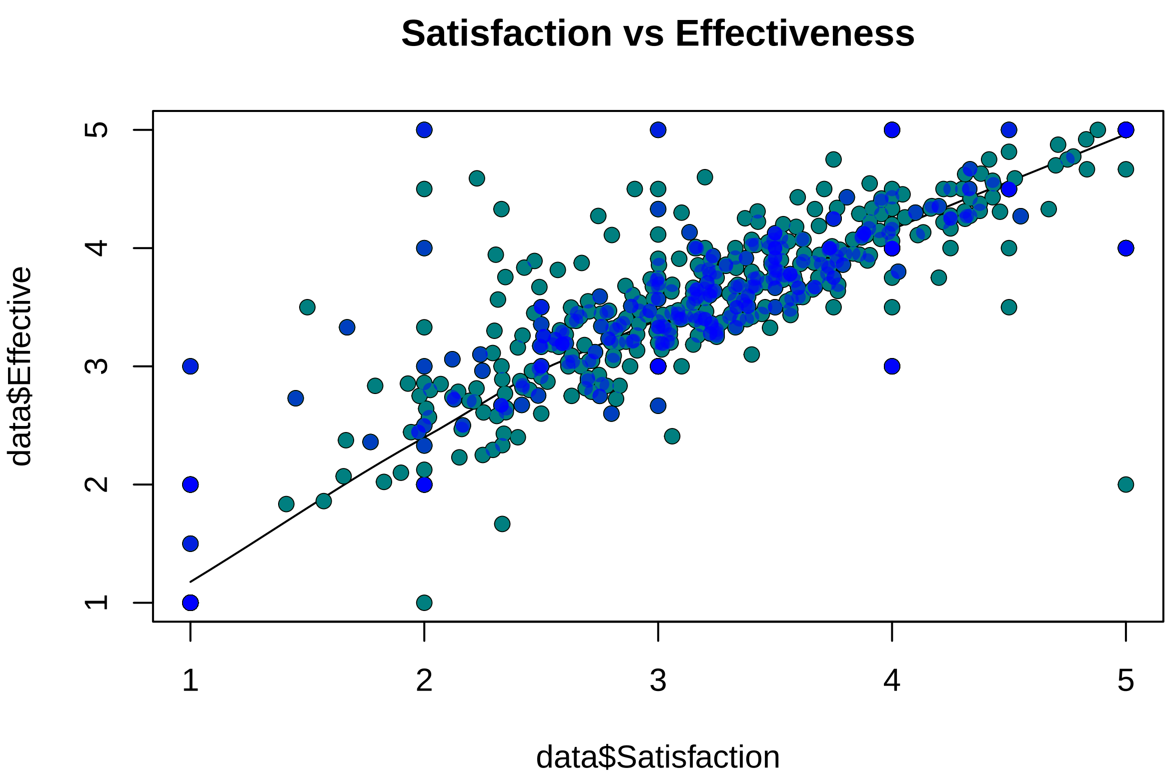
```
## The sd of the Effectiveness variable is: 0.9551967
```

```r
sd2<-sd(data$Satisfaction)
cat("The sd of the Satisfaction variable is:", sd2, "\n")
```

```
## The sd of the Satisfaction variable is: 1.030673
```

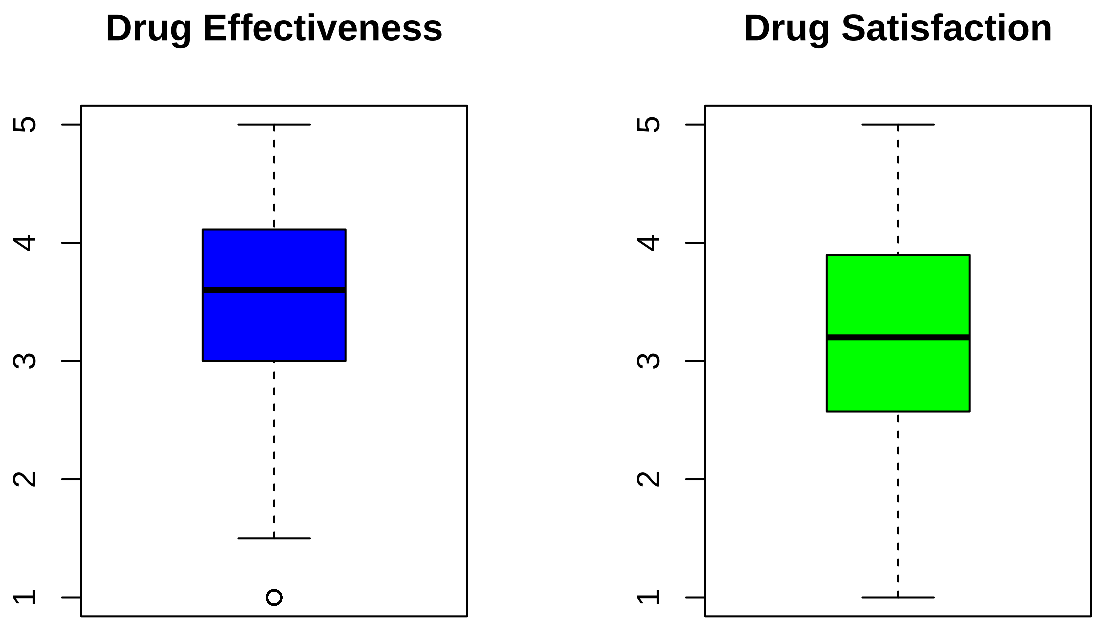
#Graphical analysis of the data Scatter Plot

```
scatter.smooth(x = data$Satisfaction, y = data$Effective, main = "Satisfaction vs Effectiveness")
points(x = data$Satisfaction, y = data$Effective, col = "green", pch = 16)
points(x = data$Satisfaction, y = data$Effective, col = rgb(0, 0, 1, alpha = 0.5), pch = 16)
```



#Graphical analysis of the data Box-Plot Of the data

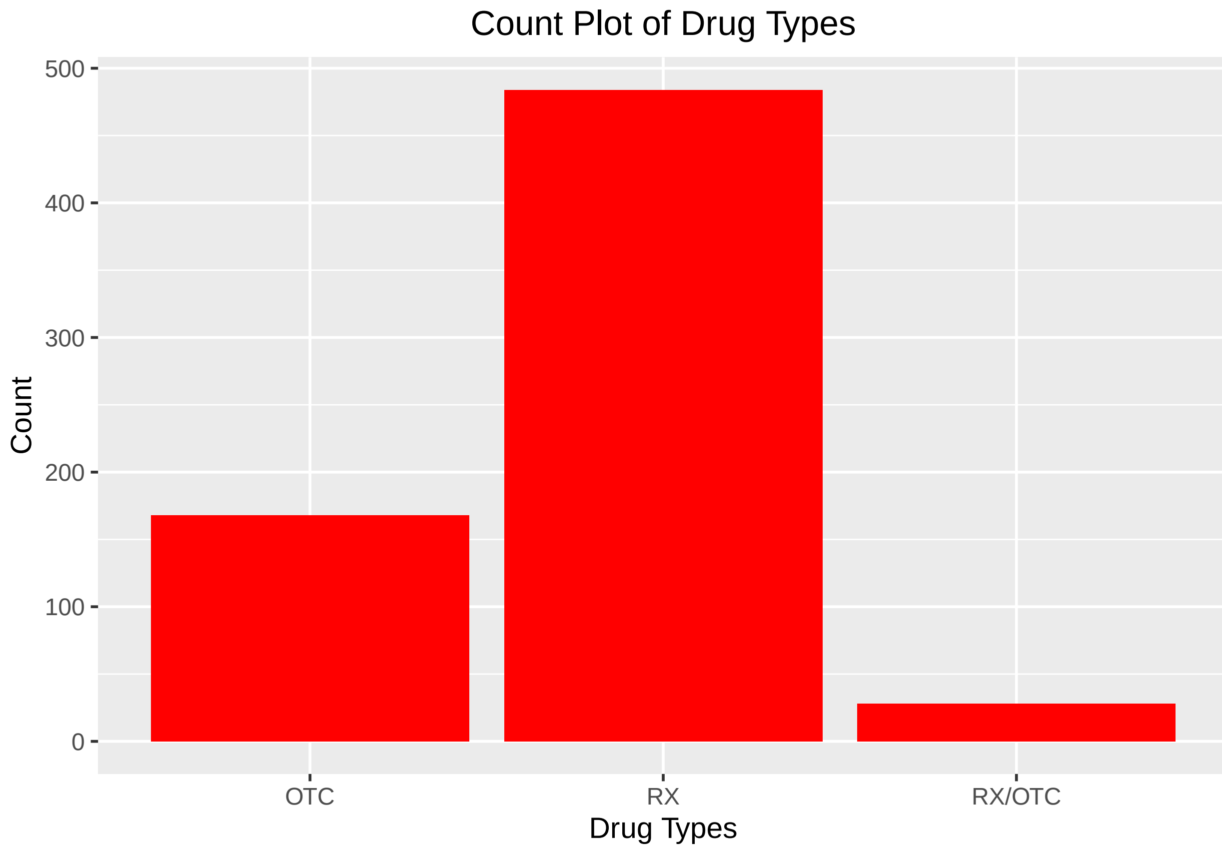
```
par(mfrow = c(1,2))
boxplot(data$Effective,main = "Drug Effectiveness", col = "blue")
boxplot(data$Satisfaction,main = "Drug Satisfaction", col = "green")
```



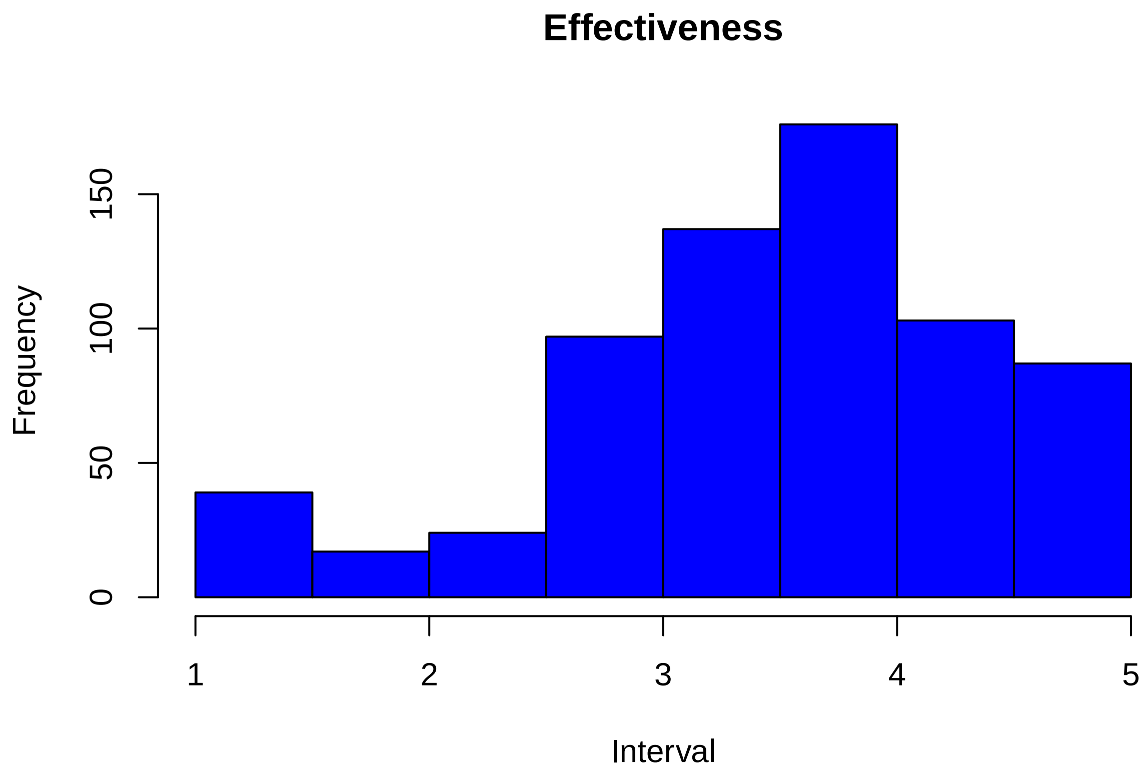
install.packages("moments") library(moments)

#Graphical analysis of the data Histogram

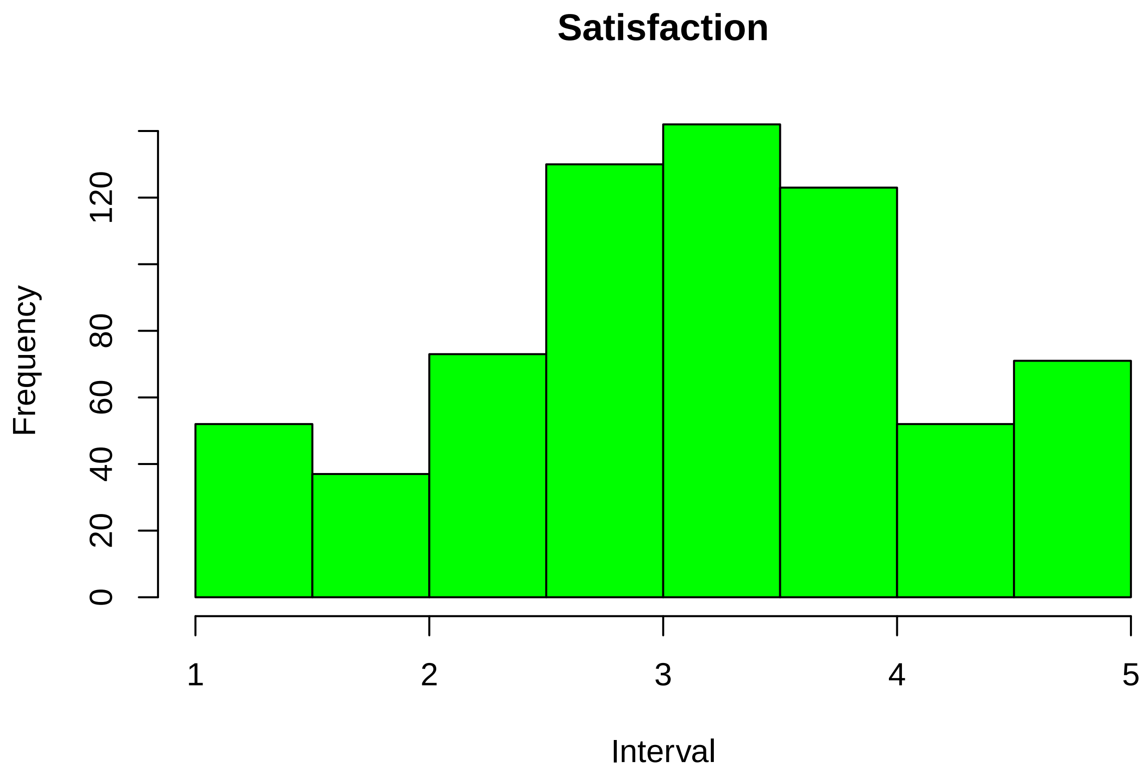
```
#Count for Drug Type
library(ggplot2)
plot <- ggplot(data, aes(x = Type)) +
  geom_bar(fill = "red") +
  labs(title = "Count Plot of Drug Types", x = "Drug Types", y = "Count") +
  theme(plot.title = element_text(hjust = 0.5))

# Display the plot
print(plot)
```




```r
#Histogram for Effective
hist(data$Effective, col = "blue",
     xlab = "Interval", main = "Effectiveness")
```




```
#Histogram for Satisfaction
hist(data$Satisfaction, col = "green",
     xlab = "Interval", main = "Satisfaction")
```

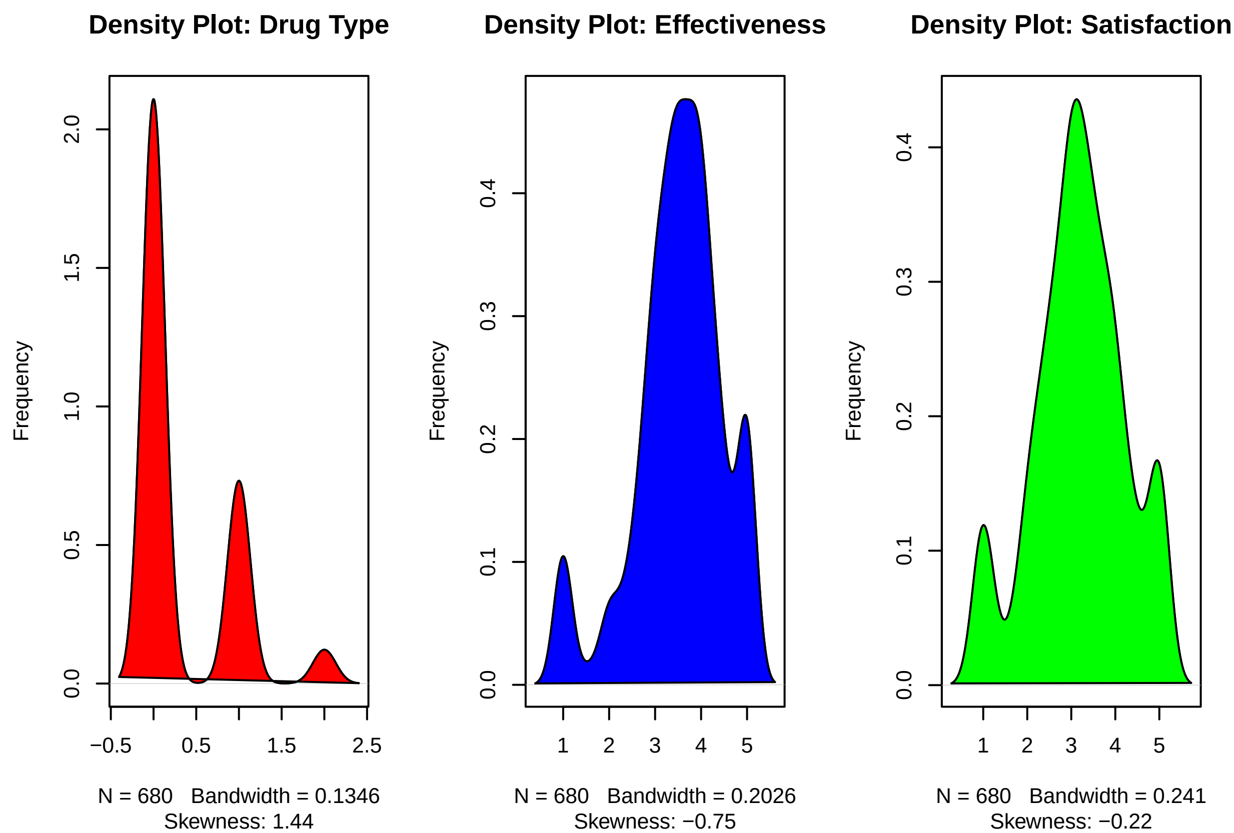


#Graphical analysis of the data Density Plot

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.2.3
```

```
par(mfrow=c(1, 3))
plot(density(data$Status), main="Density Plot: Drug Type", ylab="Frequency", sub=paste("Skewness:", rou
polygon(density(data$Status), col="red")

plot(density(data$Effective), main="Density Plot: Effectiveness", ylab="Frequency", sub=paste("Skewness
polygon(density(data$Effective), col="blue")

plot(density(data$Satisfaction), main="Density Plot: Satisfaction", ylab="Frequency", sub=paste("Skewne
polygon(density(data$Satisfaction), col="green")
```



#Correlation Test between the variables

```
# Between Status and Effectiveness
cor(data$Status,data$Effective)
```

```
## [1] 0.05501989
```

```
# Between Status and Satisfaction
cor(data$Status,data$Satisfaction)
```

```
## [1] 0.1580571
```

#Shapiro-Wilk Test

```
# Perform Shapiro-Wilk test for Effective by Status
shapiro.test(data$Effective)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Effective
## W = 0.93879, p-value = 4.219e-16
```

```r
# Perform Shapiro-Wilk test for Satisfaction by Status
shapiro.test(data$Satisfaction)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  data$Satisfaction
## W = 0.96768, p-value = 4.241e-11
```

#Kruskal-Wallis Test

```r
# Perform Kruskal-Wallis test for Effective by Status
kruskal.test(Effective ~ Status, data = data)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Effective by Status
## Kruskal-Wallis chi-squared = 17.096, df = 2, p-value = 0.0001939
```

```r
# Perform Kruskal-Wallis test for Satisfaction by Status
kruskal.test(Satisfaction ~ Status, data = data)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Satisfaction by Status
## Kruskal-Wallis chi-squared = 39.132, df = 2, p-value = 3.181e-09
```

#Calculating the Skewness and the Kurtosis for the Continous Variables

```r
# Calculate skewness
skew <- skewness(data$Satisfaction)
cat("The skewness of the Satisfaction variable is:", skew, "\n")
```

```
## The skewness of the Satisfaction variable is: -0.2169119
```

```r
# Calculate kurtosis
kurt <- kurtosis(data$Satisfaction)
cat("The kurtosis of the Satisfaction variable is:", kurt, "\n")
```

```
## The kurtosis of the Satisfaction variable is: -0.2403637
```

```r
# Calculate skewness
skew <- skewness(data$Effective)
cat("The skewness of the Effective variable is:", skew, "\n")
```

```
## The skewness of the Effective variable is: -0.7535869
```

```r
# Calculate kurtosis
kurt <- kurtosis(data$Effective)
cat("The kurtosis of the Effective variable is:", kurt, "\n")
```

```
## The kurtosis of the Effective variable is: 0.6951788
```

#Building the Generalized Linear Model Effectiveness and Satisfaction Combined

```r
model1 <- glm(Status ~ Effective + Satisfaction, data = data)
summary(model1)
```

```
##
## Call:
## glm(formula = Status ~ Effective + Satisfaction, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6669  -0.3514  -0.2676   0.5223   1.7378
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.24173    0.07897   3.061  0.00229 **
## Effective    -0.18926    0.04321  -4.380 1.38e-05 ***
## Satisfaction  0.23644    0.04005   5.904 5.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.2888043)
##
##     Null deviance: 206.21  on 679  degrees of freedom
## Residual deviance: 195.52  on 677  degrees of freedom
## AIC: 1090.2
##
## Number of Fisher Scoring iterations: 2
```

#Proceeding ahead with Mutlinomial Logistic Regression owing to the high AIC by GLM

#Building the Multinomial Logistic Regression Model Effectiveness and Satisfaction Combined

```r
library(nnet)
model2 <- multinom(Status ~ Effective + Satisfaction, data = data)
```

```
## # weights:  12 (6 variable)
## initial  value 747.056356
## iter  10 value 463.068247
## iter  20 value 460.538397
## iter  20 value 460.538394
## iter  20 value 460.538394
## final  value 460.538394
## converged
```

```r
summary(model2)
```

```
## Call:
## multinom(formula = Status ~ Effective + Satisfaction, data = data)
##
## Coefficients:
##   (Intercept) Effective Satisfaction
## 1   -1.918732 -1.018603     1.359812
## 2   -2.176566 -1.125653     1.014652
##
## Std. Errors:
##   (Intercept) Effective Satisfaction
## 1   0.3810441 0.2491760    0.2321839
## 2   0.6419704 0.4576465    0.4438286
##
## Residual Deviance: 921.0768
## AIC: 933.0768
```

#Multinomial Logistic Regression performs better when compared to GLM on Residual Deviance and AIC

#Building the Multinomial Logistic Regression Model Between Satisfaction and Status

```r
library(nnet)
model3 <- multinom(Status ~ Satisfaction, data = data)
```

```
## # weights:  9 (4 variable)
## initial  value 747.056356
## iter  10 value 471.863160
## final  value 471.862290
## converged
```

```r
summary(model3)
```

```
## Call:
## multinom(formula = Status ~ Satisfaction, data = data)
##
## Coefficients:
##   (Intercept) Satisfaction
## 1   -2.831545   0.53271944
## 2   -2.978803   0.04181026
##
## Std. Errors:
##   (Intercept) Satisfaction
## 1   0.3440076   0.09603717
## 2   0.6241925   0.19116386
##
## Residual Deviance: 943.7246
## AIC: 951.7246
```

##Building the Multinomial Logistic Regression Model Between Effectiveness and Status

```r
library(nnet)
model4 <- multinom(Status ~ Effective, data = data)
```

```
## # weights:  9 (4 variable)
## initial  value 747.056356
## iter  10 value 483.491500
## final  value 483.491484
## converged
```

```r
summary(model4)
```

```
## Call:
## multinom(formula = Status ~ Effective, data = data)
##
## Coefficients:
##   (Intercept)  Effective
## 1   -2.118127  0.2944319
## 2   -2.245028 -0.1788270
##
## Std. Errors:
##   (Intercept) Effective
## 1   0.3824528 0.1014545
## 2   0.6561624 0.1896376
##
## Residual Deviance: 966.983
## AIC: 974.983
```