Q1)Extract a dataset of your choice and compute the correlation between any two variables and visualize the relationship using scatter plot.

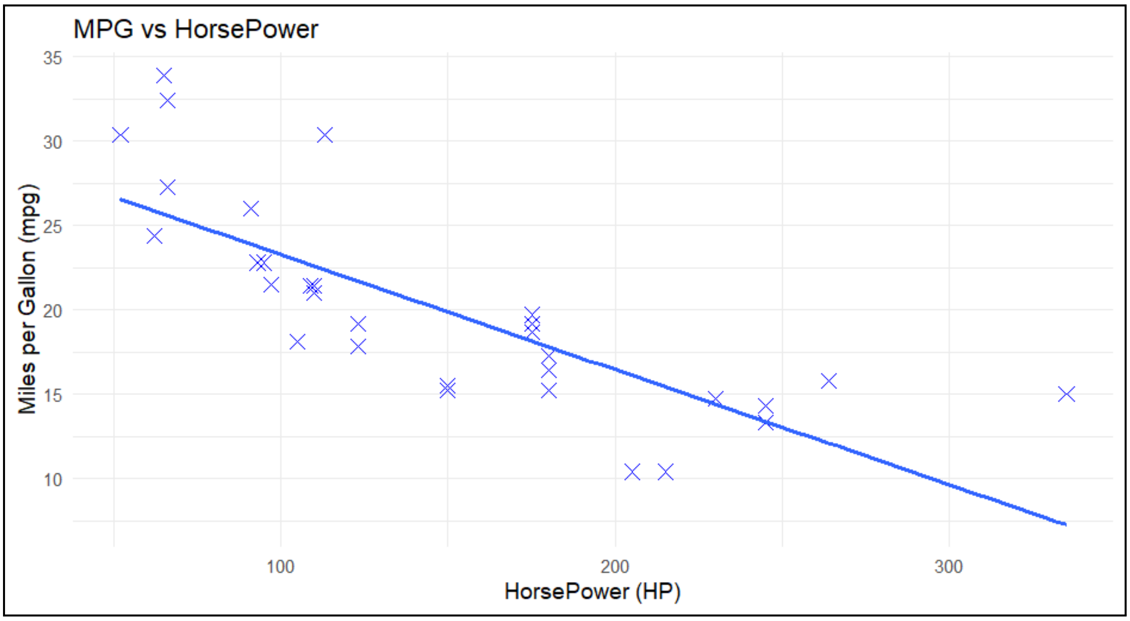
CODE:

```
# Load the built-in dataset
data(mtcars)

# View the dataset
head(mtcars)

# Compute correlation between mpg (Miles per Gallon) and hp (Horsepower)
correlation <- cor(mtcars$mpg, mtcars$hp)

# Print the correlation
print(paste("correlation between mpg and hp is:", round(correlation,2)))

# Create scatter plot with regression line
library(ggplot2)
ggplot(data = mtcars, aes(x = hp, y = mpg)) +
  geom_point(shape = 4, color = "blue", size = 3) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(
    title = "MPG vs HorsePower",
    x = "HorsePower (HP)",
    y = "Miles per Gallon (mpg)"
  ) +
  theme_minimal()
```

OUTPUT:

```
> # View the dataset
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
>
```

"correlation between mpg and hp is: -0.78"



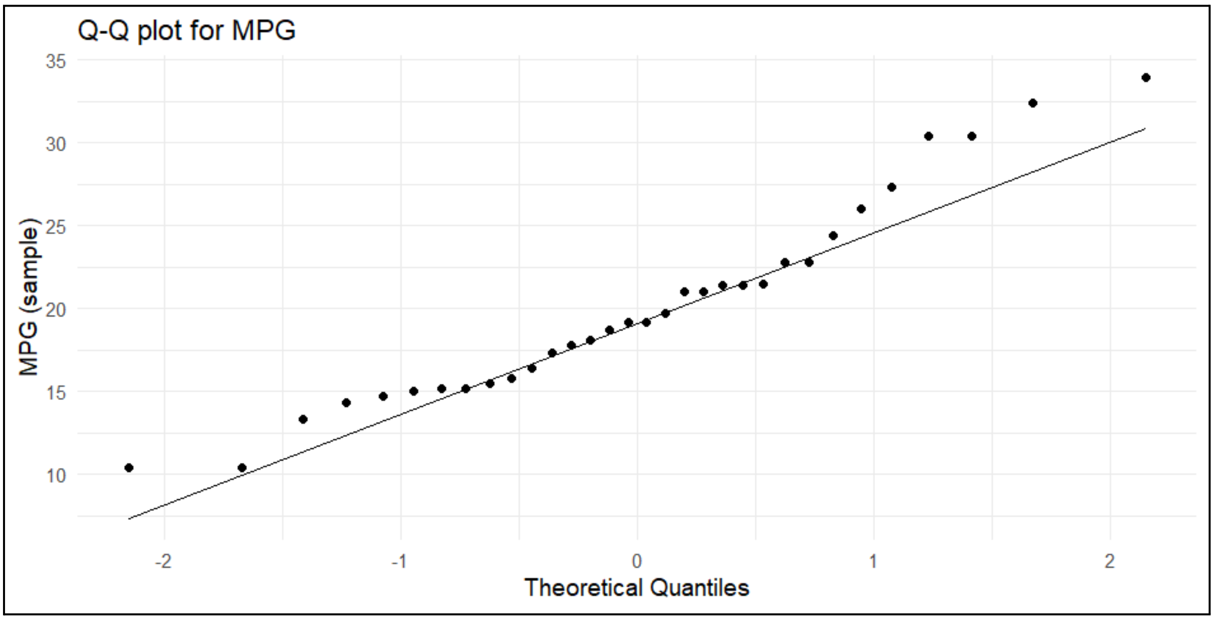
Q2)
Apply the Pearson correlation test on a dataset, show the normality of variables using Q-Q plot and interpret the results.

CODE:

```r
# Load the built-in dataset mtcars and display it
data()
mtcars

# Load ggplot2 library
library(ggplot2)

# Compute correlation between mpg (Miles per Gallon) and hp (Horsepower)
correlation <- cor(mtcars$mpg, mtcars$hp)

# Print the correlation
print(paste("correlation between mpg and hp is:", round(correlation,2)))

# Create Q-Q plot for normality check on mpg (miles per gallon) values
ggplot(data = mtcars, aes(sample = mpg)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Q-Q plot for MPG",
       x = "Theoretical Quantiles",
       y = "MPG (sample)") +
  theme_minimal()
```

OUTPUT:

```
        Pearson's product-moment correlation

data:  mtcars$mpg and mtcars$hp
t = -6.7424, df = 30, p-value = 1.788e-07
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.8852686 -0.5860994
sample estimates:
       cor
-0.7761684
```



Q3) Consider the Price quotes dataset and perform the following:
1. Generate the Summary statistics of the price quotes from Mary and Barry and interpret the results.
2. The standard deviation of Mary's price quotes is $11.05. The standard error of the mean of Mary's price quotes is $3.19. Both are measures of variability.
a) distinguish between these two numbers on the basis of how they are calculated and what they mean.
b) Provide an interpretation of each number.

CODE:

```
#Load the libraries
library(ggplot2)

# Load the dataset and summary statistics
data = read.csv("C:/Users/hp/Desktop/Stats Lab Dataset/pricequotes.csv")
print(summary(data))

#find the value of n
n_barry <- length(data$Barry_Price)
n_mary <- length(data$Mary_Price)

#find the standard deviation and mean
sd_barry <- sd(data$Barry_Price)
sd_mary <- sd(data$Mary_Price)
se_barry <- sd_barry/sqrt(n_barry)
se_mary <- sd_marry/sqrt(n_mary)
cat("Mary: SD=",round(sd_marry,2)," | SE = ",round(se_mary,2))
cat("Barry: SD=",round(sd_barry,2)," | SE = ",round(se_barry,2))

# Boxplot to compare distributions
ggplot(data,aes(x="Barry",y=Barry_Price))+
   geom_boxplot(fill="skyblue")+
   geom_boxplot(aes(x="Mary",y=Mary_Price),fill="lightgreen")+
   labs(title="BoxPlot of Price QUotes",x="Person",y="Price")

# Print interpretations
cat("\nInterpretation:\n")
cat("Standard Deviation (SD):", mary_sd, "- shows how much individual quotes from Mary vary from her average quote.\n")
cat("Standard Error of Mean (SEM):", mary_sem, "- shows how precise Mary's mean quote is, based on sample size.\n")
```

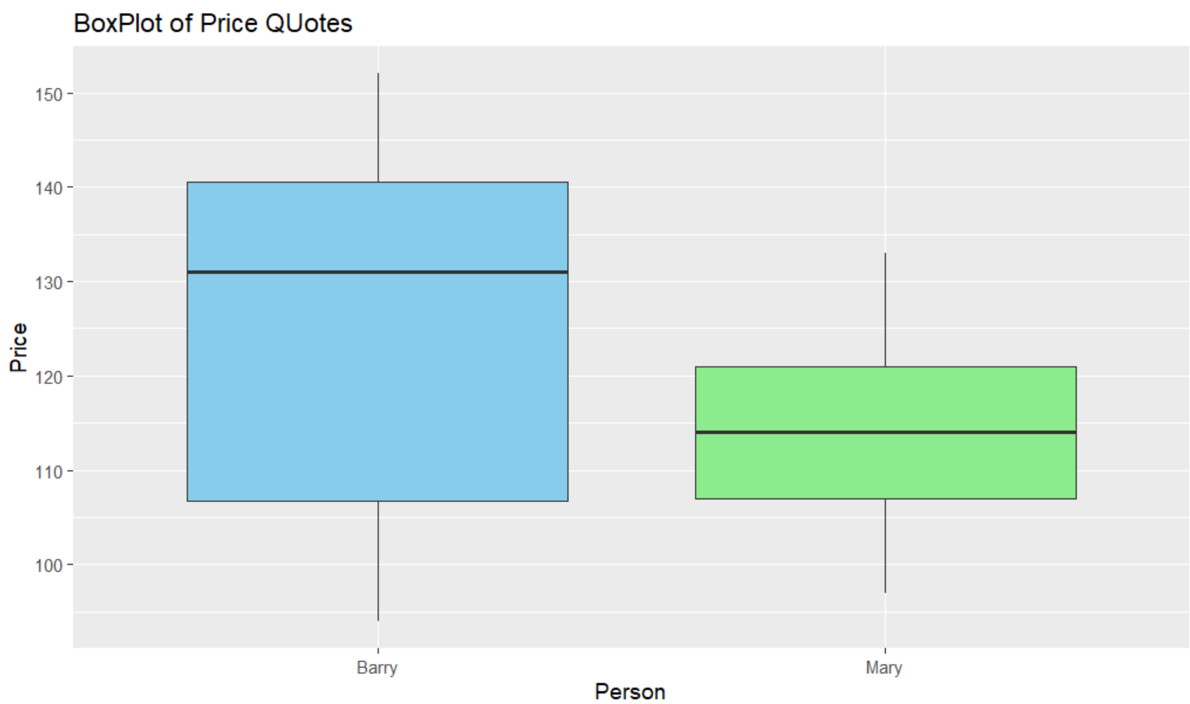
OUTPUT:

Order_Number	Barry_Price	Mary_Price
Min. : 1.00	Min. : 94.0	Min. : 97.0
1st Qu.: 3.75	1st Qu.:106.8	1st Qu.:107.0
Median : 6.50	Median :131.0	Median :114.0
Mean : 6.50	Mean :124.3	Mean :114.8
3rd Qu.: 9.25	3rd Qu.:140.5	3rd Qu.:121.0
Max. :12.00	Max. :152.0	Max. :133.0

Mary: SD= 11.05 | SE = 3.19

Barry: SD= 20.7 | SE = 5.98

Interpretation:
Standard Deviation (SD): 11.05 - shows how much individual quotes from Mary vary from her average quote.
Standard Error of Mean (SEM): 3.19 - shows how precise Mary's mean quote is, based on sample size.



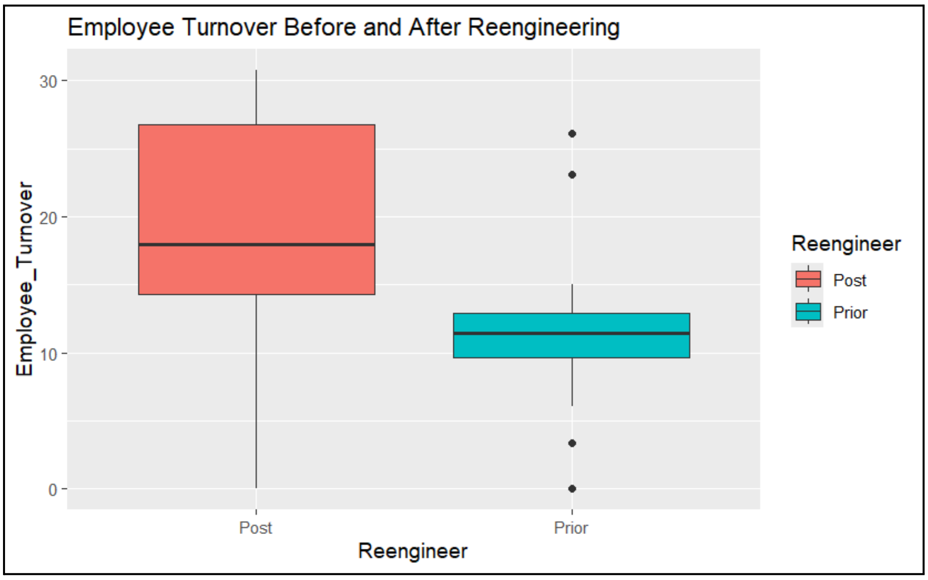
Q4) Consider the Treatment Facility dataset and perform the following:
1. Generate the Summary statistics of the Treatment facility and interpret the results.
2. Determine what effect the reengineering effort had on the incidence behavioral problems and staff turnover.

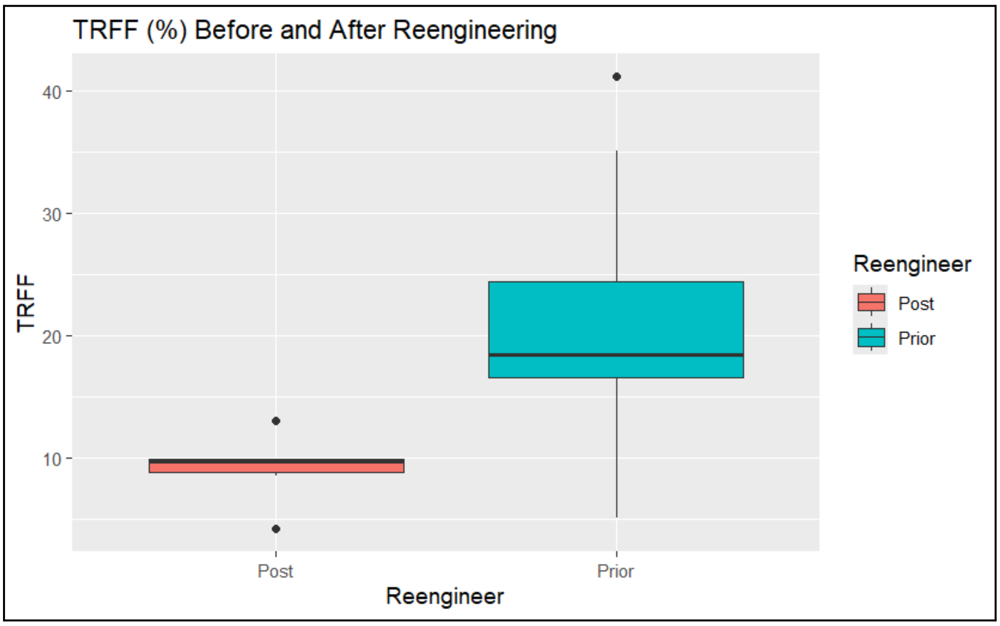
CODE:

```
#Libraries used
library(dplyr)
library(ggplot2)

#Load the dataset
df<- read.csv("treatmentfacility.csv")
df$Reengineer <- factor(df$Reengineer,levels=c("Prior","Post"))

#Print the summary of dataset
summary_stats = df %>%
group_by(Reengineer) %>%
summarize(
   n=n(),
   mean_turnover = mean(Employee_Turnover),
   sd_turnover = sd(Employee_Turnover),
   mean_TRFF = mean(TRFF),
   sd_TRFF = sd(TRFF),
   mean_CI = mean(CI),
   sd_CI = sd(CI)
)
print(summary_stats)


#Plot the graphs to show the reengineering effect
ggplot(df,aes(x=Reengineer,y=Employee_Turnover,fill=Reengineer))+
  geom_boxplot()+
  labs(title = "Employee Turnover Before and Afterr Rengineering")

ggplot(df,aes(x=Reengineer,y=TRFF,fill=Reengineer))+
  geom_boxplot()+
  labs(title="TRFF Before and After Engineering")d
```

OUTPUT:

Reengineer	n	mean_turnover	sd_turnover	mean_TRFF	sd_TRFF	mean_CI	sd_CI
1 Prior	13	11.7	7.04	20.5	10.4	53.9	48.7
2 Post	7	18.7	10.6	9.23	2.63	23.3	7.81





Q5)Consider the Baggage complaints dataset and perform the following:
1. Generate the Summary statistics and interpret the results.
2. Compare the baggage complaints for three airlines: American Eagle, Hawaiian, and United. Which airline has the best record? The worst? Are complaints getting better or worse over time? Are there other factors, such as destinations, seasonal effects or the volume of travelers that affect baggage performance?

CODE:

```
# Load the required libraries
library(readr)
library(ggplot2)

#Load and adjust the dataset
df <- read.csv("baggagecomplaints.csv")
df <- df %>%
  mutate(Rate = 100 * Baggage/Enplaned )
print(summary(df[c("Baggage","Rate")]))

#Print summary of dataset
summary_airline = df %>%
  group_by(Airline) %>%
  summarize(
    total_months = n(),
    total_passangers = sum(Enplaned),
    mean_complaints = mean(Baggage),
    median_complaints = median(Baggage),
    sd_complaints = sd(Baggage),
    mean_rate = mean(Rate),
    median_rate = median(Rate),
    sd_rate = sd(Rate),
    min_rate = min(Rate),
    max_rate = max(Rate)
  )
print(summary_airline,n=Inf,width = Inf)

 average complaints per year for each of the selected airlines
yearly_avg <- df%>%
  group_by(Year,Airline) %>%
  summarise(
    avg_complaints = mean(Baggage),
    .groups="drop" )
```

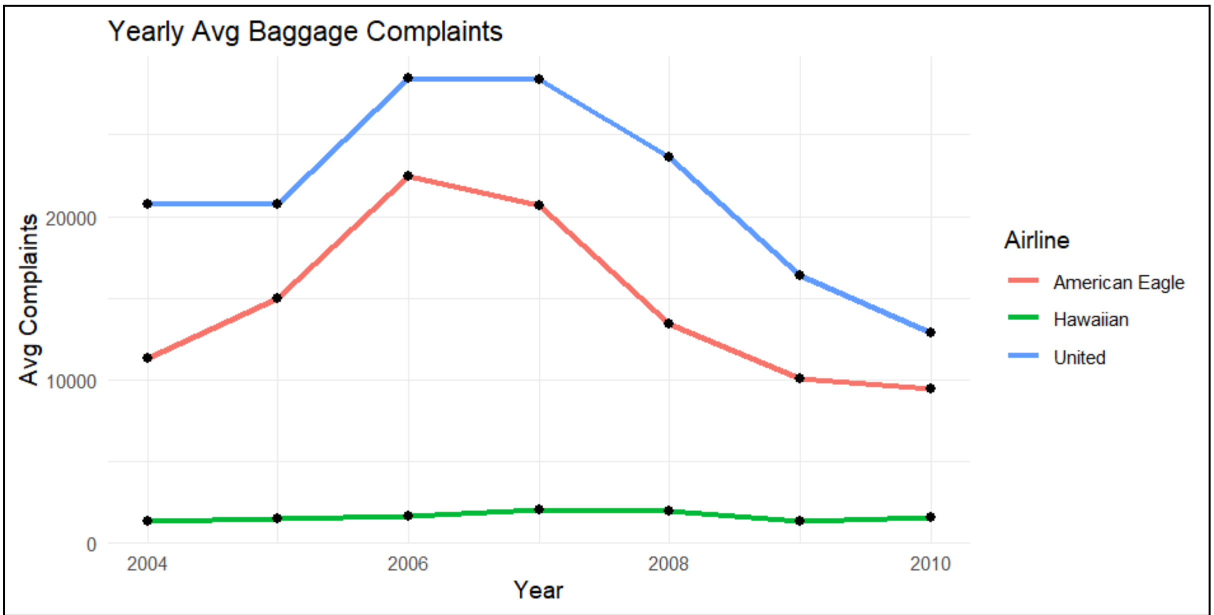
#Plot the graph to compare baggage complaints

```
ggplot(yearly_avg, aes(x=Year,y=avg_complaints,color=Airline))+
  geom_line(linewidth=1.2)+
  geom_point(size=2,color="black")+
  theme_minimal()+
  labs(
    title="Yearly Avg Baggage Complaints",
    x="Year",
    y="Avg Complaints"
  )
```

OUTPUT:

```
> print(summary(df[c("Baggage","Rate")]))
    Baggage           Rate       
 Min.   : 1033   Min.   :0.1606  
 1st Qu.: 1910   1st Qu.:0.3080  
 Median :12224   Median :0.4208  
 Mean   :12614   Mean   :0.5914  
 3rd Qu.:19359   3rd Qu.:0.7872  
 Max.   :41787   Max.   :1.9321  
```

```
> print(summary_airline,n=Inf,width = Inf)
# A tibble: 3 × 11
  Airline         total_months total_passangers mean_complaints median_complaints
  <chr>                  <int>            <dbl>           <dbl>             <dbl>
1 American Eagle            84        117324946          14619.             13111
2 Hawaiian                  84         49910630           1622.              1516.
3 United                    84        388139830          21600.             19986.
  sd_complaints mean_rate median_rate sd_rate min_rate max_rate
          <dbl>     <dbl>       <dbl>   <dbl>    <dbl>    <dbl>
1         5696.     1.03        0.954  0.341     0.543    1.93 
2          424.     0.277       0.277  0.0669    0.161    0.402
3         7830.     0.464       0.421  0.150     0.241    0.907
```



Q6)Consider the Medical Malpractice dataset and perform the following.
1. Using descriptive statistics and graphical displays, explore claim payment amounts, and identify factors that appear to influence the amount of the payment.
2. Use the data set to answer the following questions: What percentage of the sample involved Anesthesiologists? Dermatologists? Orthopedic surgeons? Is there any relationship between age of the patient and size of the payment?

CODE:
```
# Load required libraries
library(ggplot2)
library(dplyr)
library(readr)

# Load the dataset and find the summary statistics
data <- read.csv('medicalmalpractice.csv')
summary(data$Amount)

# 2. Histogram of claim amount (log scale)
ggplot(data, aes(x = log10(Amount))) +
  geom_histogram(fill = "lightblue", bins = 20) +
  labs(title = "Histogram of Claim Amounts (Log Scale)",
x = "Log Amount",
y = "Frequency")

# 3. Boxplot for top 3 specialties
top3_specialty <- data %>%
                  count(Specialty,name="n") %>%
                  slice_max(n,n=3) %>%
                  pull(Specialty)

data %>%
  filter(Specialty %in% top3_specialty) %>%
ggplot(aes(x=Specialty,y=Amount,fill=Specialty))+
  geom_boxplot()+
  coord_flip()+
  theme_minimal()


# 4. Percentage of specific specialties
total = length(data$Specialty)
spec_percent <- data %>%
group_by(Specialty) %>%
summarise(
n=n(),
pct = 100*n/total ) %>%
```

```
filter(Specialty %in% c("Anesthesiology","Dermatology","Orthopedic Surgery"))
spec_percent

# 5. Correlation between Age and Amount
cor.test(data$Age, data$Amount
```

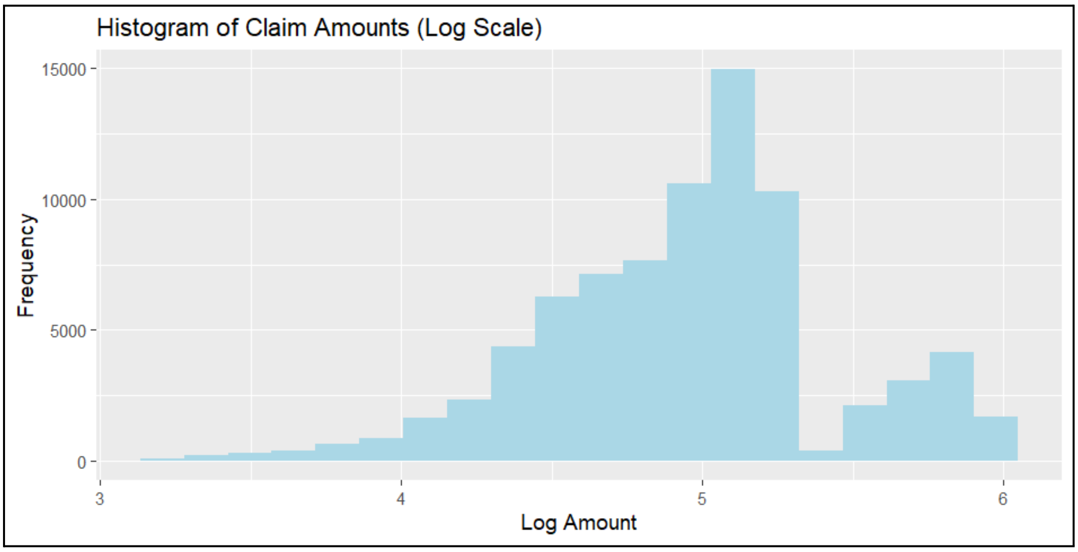
OUTPUT:

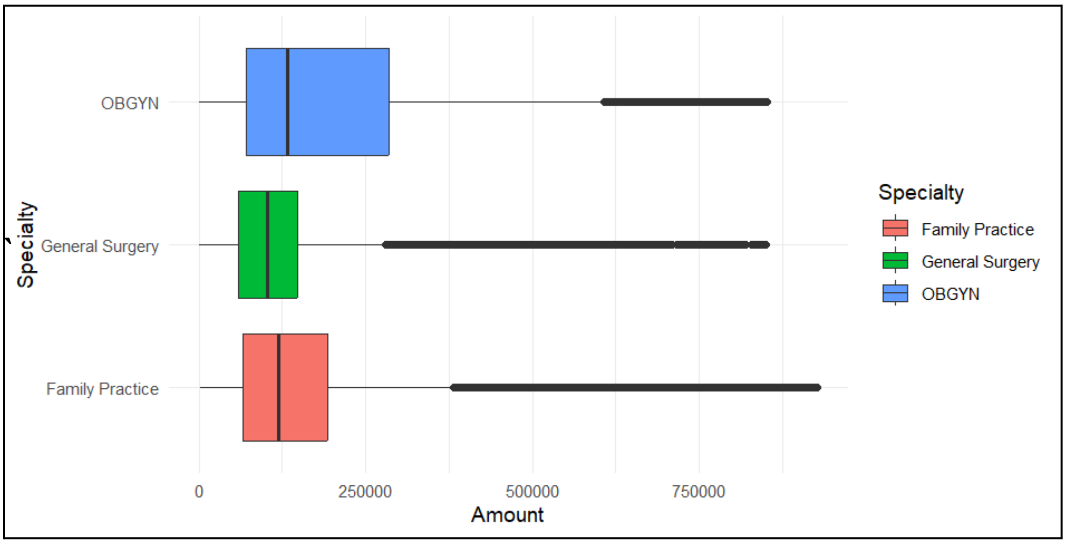
Claim Payment Amounts:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1576	43670	98131	157485	154675	926411

Percentage of Claims:
Anesthesiology: 11.02 %
Dermatology: 1.75 %
Orthopedic Surgery: 9.18 %

Age vs. Amount Correlation:
Correlation: -0.105
P-value: 4.765828e-194





Q9) Consider the scores of ten students in SMIP and DBMS and Compute the Spearman rank correlation and Interpret the results using Python programming.

SMIP	70	46	94	34	20	86	18	12	56	64	42
DBMS	60	66	90	46	16	98	24	08	32	54	62

CODE:

```
import scipy.stats as stats

# Scores of 10 students
smip_scores = [70, 46, 94, 34, 20, 86, 18, 12, 56, 64]
dbms_scores = [60, 66, 90, 46, 16, 98, 24, 8, 32, 54]

# Spearman Rank Correlation
correlation, p_value = stats.spearmanr(smip_scores, dbms_scores)

print(f"Spearman Rank Correlation Coefficient: {correlation:.4f}")
print(f"P-value: {p_value:.4f}")
```

OUTPUT:

Spearman Rank Correlation Coefficient: 0.8788
P-value: 0.0008
There is a statistically significant correlation between SMIP and DBMS scores.

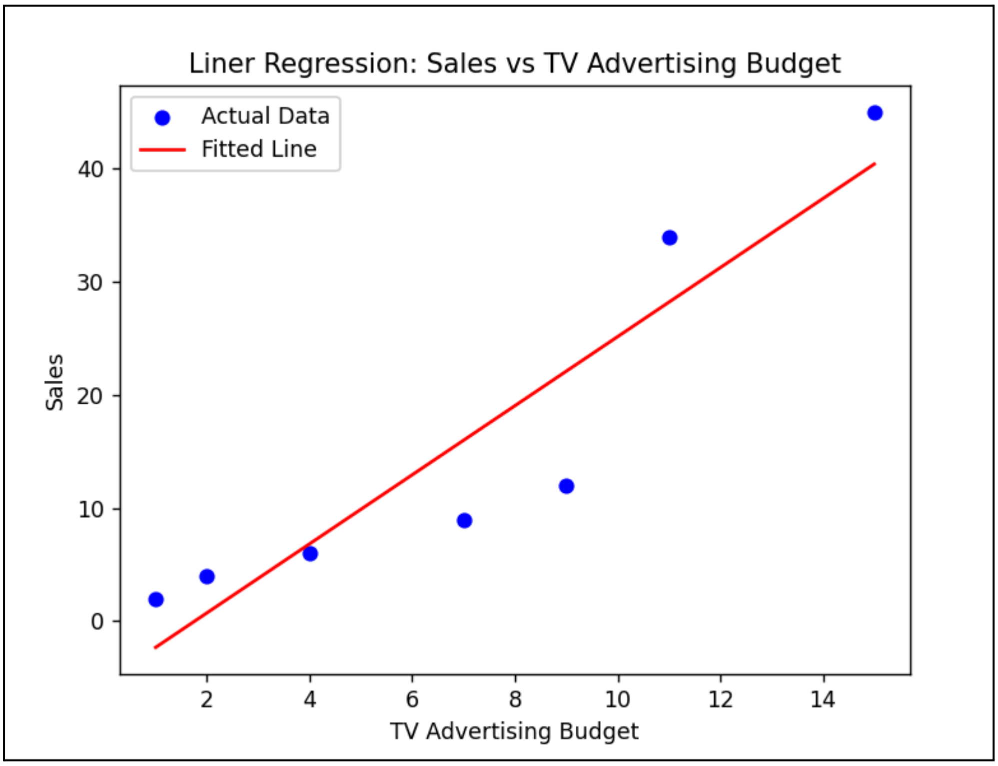
Q10) Develop a Python code to build a simple Linear Regression model to predict sales units based on the advertising budget spent on TV. Display the statistical summary of the model.

CODE:

```
#Load the required libraries
import pandas as pd
import statsmodels.api as sm
import matplotlib.pyplot as plt

#Load the dataset
data = {
   'TV': [1,2,4,7,9,11,15],
   'Sales': [2,4,6,9,12,34,45]
}
df = pd.DataFrame(data)

#Define (X) and (Y) variables
X = df['TV']
Y = df['Sales']

#Add constant and fit the regression model
X = sm.add_constant(X)
# print(X)
model = sm.OLS(Y,X).fit()

#Print summary
print(model.summary())

#Plot the graph
plt.scatter(df['TV'],df['Sales'], color='blue', label='Actual Data')
plt.plot(df['TV'],model.predict(X),color='red', label='Fitted Line')
plt.title("Liner Regression: Sales vs TV Advertising Budget")
plt.xlabel("TV Advertising Budget")
plt.ylabel("Sales")
plt.legend()
plt.show()
```

OUTPUT:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.859
Model:                            OLS   Adj. R-squared:                  0.831
Method:                 Least Squares   F-statistic:                     30.44
Date:                Thu, 10 Jul 2025   Prob (F-statistic):            0.00268
Time:                        21:22:58   Log-Likelihood:                -22.239
No. Observations:                   7   AIC:                             48.48
Df Residuals:                       5   BIC:                             48.37
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -5.3636      4.661     -1.151      0.302     -17.345       6.617
TV             3.0519      0.553      5.518      0.003       1.630       4.474
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   1.357
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.958
Skew:                          -0.709   Prob(JB):                        0.619
Kurtosis:                       1.873   Cond. No.                         15.3
==============================================================================
```



Q11) Consider the Australian Drug Sales dataset and develop a Python code to perform Time Series Analysis and visualize using plots.

CODE:

```
#Load the libraries
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.holtwinters import ExponentialSmoothing

#Load the dataset
df = pd.read_csv("AustraliaDrugSales.csv",parse_dates=['date'])
df.set_index('date', inplace=True)
df.index.freq = 'MS'

#create and fit a time series forecasting model
model = ExponentialSmoothing(
    df['value'],
    trend='add',
    seasonal='add',
    seasonal_periods=12
).fit()
forecast = model.forecast(24)

#Plot the graph
plt.plot(df.index,df['value'],label='Observerd',color='blue')
plt.plot(model.fittedvalues.index, model.fittedvalues, label='Fitted', color='red')
plt.plot(forecast.index, forecast,label='Forecast',color='green')
plt.legend()
plt.title("Monthly Drug Sales in Australia")
plt.xlabel("Date")
plt.ylabel("Sales")
plt.show()
```

OUTPUT: