Applied data project (or Assignment 1)

Name:Pratham Radhakrishna

ID Number: 3997064

Task 1

Descriptive Statistics

```
# Loading the package
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
#setting the file directory
setwd("C:/Users/Admin/Desktop/applied analytics")
# Loading S&P 500 dataset
sp500_data <- read.csv("S&P 500.csv", header = TRUE, stringsAsFactors = FALSE)
# Renaming column from 'ï..Date' to 'Date'
colnames(sp500_data)[colnames(sp500_data) == "ï..Date"] <- "Date"
# Loading Bitcoin dataset
bitcoin_data <- read.csv("BTC-USD.csv", header = TRUE, na.strings = "", stringsAsFactors = FA
LSE, strip.white = TRUE)
```

```
# Displaying the first few rows of each dataset
head(sp500_data)
```

```
##        Date    Price
## 1 2/04/2019 2,867.24
## 2 3/04/2019 2,873.40
## 3 4/04/2019 2,879.39
## 4 5/04/2019 2,892.74
## 5 8/04/2019 2,895.77
## 6 9/04/2019 2,878.20
```

```
head(bitcoin_data)
```

```
##        Date Close.price.adjusted  X X.1 X.2 X.3 X.4 X.5 X.6 X.7 X.8 X.9
## 1 2/04/2019             4879.878 NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 2 3/04/2019             4973.022 NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 3 4/04/2019             4922.799 NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 4 5/04/2019             5036.681 NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 5 6/04/2019             5059.817 NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 6 7/04/2019             5198.897 NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
```

```
# Checking the structure and summary of the datasets
str(sp500_data)
```

```
## 'data.frame':    1258 obs. of  2 variables:
##  $ Date : chr  "2/04/2019" "3/04/2019" "4/04/2019" "5/04/2019" ...
##  $ Price: chr  "2,867.24" "2,873.40" "2,879.39" "2,892.74" ...
```

```
summary(sp500_data)
```

```
##      Date              Price
##  Length:1258        Length:1258
##  Class :character   Class :character
##  Mode  :character   Mode  :character
```

```
str(bitcoin_data)
```

```
## 'data.frame':    1827 obs. of  12 variables:
##  $ Date                : chr  "2/04/2019" "3/04/2019" "4/04/2019" "5/04/2019" ...
##  $ Close.price.adjusted: num  4880 4973 4923 5037 5060 ...
##  $ X                   : logi  NA NA NA NA NA NA ...
##  $ X.1                 : logi  NA NA NA NA NA NA ...
##  $ X.2                 : logi  NA NA NA NA NA NA ...
##  $ X.3                 : logi  NA NA NA NA NA NA ...
##  $ X.4                 : logi  NA NA NA NA NA NA ...
##  $ X.5                 : logi  NA NA NA NA NA NA ...
##  $ X.6                 : logi  NA NA NA NA NA NA ...
##  $ X.7                 : logi  NA NA NA NA NA NA ...
##  $ X.8                 : logi  NA NA NA NA NA NA ...
##  $ X.9                 : logi  NA NA NA NA NA NA ...
```

```
summary(bitcoin_data)
```

```
##      Date           Close.price.adjusted    X             X.1
##  Length:1827        Min.   : 4880        Mode:logical   Mode:logical
##  Class :character   1st Qu.:10580        NA's:1827      NA's:1827
##  Mode  :character   Median :25576
##                     Mean   :27098
##                     3rd Qu.:39761
##                     Max.   :73084
##    X.2            X.3            X.4            X.5            X.6
##  Mode:logical   Mode:logical   Mode:logical   Mode:logical   Mode:logical
##  NA's:1827      NA's:1827      NA's:1827      NA's:1827      NA's:1827
##
##
##
##
##    X.7            X.8            X.9
##  Mode:logical   Mode:logical   Mode:logical
##  NA's:1827      NA's:1827      NA's:1827
##
##
##
##
```

```
#cleaning data for bitcoin coin dataset
# Removing columns with only NA values
bitcoin_data_clean <- bitcoin_data[, colSums(is.na(bitcoin_data)) != nrow(bitcoin_data)]
#Coverting the price column to numeric type
bitcoin_data_clean$Close.price.adjusted <- as.numeric(gsub(",", "", bitcoin_data_clean$Close.
price.adjusted))
# Converting 'Date' column to Date type
bitcoin_data_clean$Date <- as.Date(bitcoin_data_clean$Date, format = "%d/%m/%Y")
```

```
# Converting 'Price' column to numeric of sp500 data
sp500_data$Price <- as.numeric(gsub(",", "", sp500_data$Price))
# Converting 'Date' column to Date type
sp500_data$Date <- as.Date(sp500_data$Date, format = "%d/%m/%Y")
```

```
# Calculating mean, median,range, standard deviation for 'Price' column
sp500_mean <- mean(sp500_data$Price)
sp500_median <- median(sp500_data$Price)
sp500_min <- min(sp500_data$Price)
sp500_max <- max(sp500_data$Price)
sp500_range <- sp500_max - sp500_min
sp500_sd <- sd(sp500_data$Price)
```

```
#Descriptive Statistics for Bitcoin Data
# Calculating descriptive statistics for 'Close.price.adjusted' column
bitcoin_mean <- mean(bitcoin_data_clean$Close.price.adjusted)
bitcoin_median <- median(bitcoin_data_clean$Close.price.adjusted)
bitcoin_min <- min(bitcoin_data_clean$Close.price.adjusted)
bitcoin_max <- max(bitcoin_data_clean$Close.price.adjusted)
bitcoin_range <- bitcoin_max - bitcoin_min
bitcoin_sd <- sd(bitcoin_data_clean$Close.price.adjusted)
```

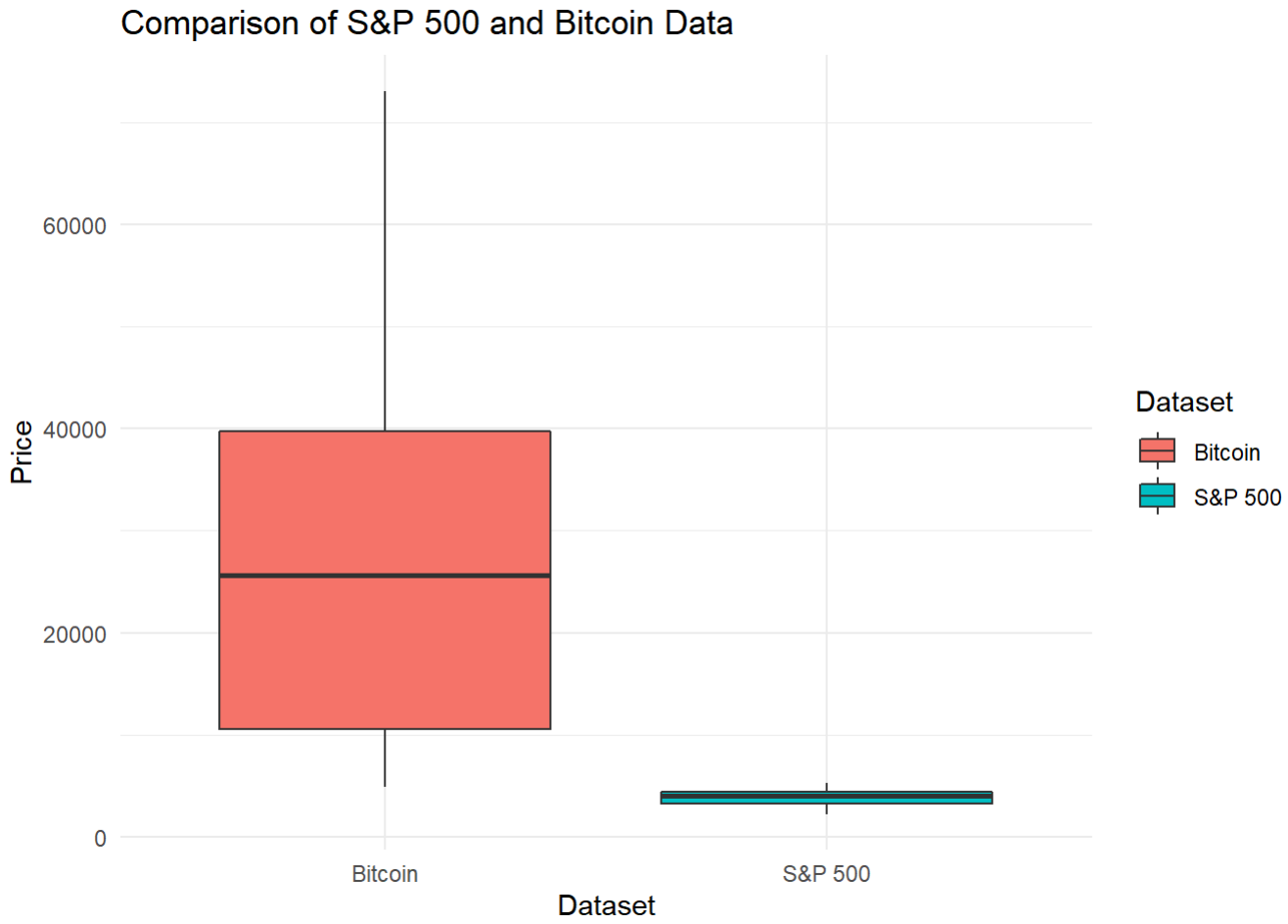
```
# Creating a summary table to compare statistics
comparison_table <- data.frame(
  Measure = c("Mean", "Median","Range", "Standard Deviation"),
  S_and_P_500 = c(sp500_mean, sp500_median,  sp500_range, sp500_sd),
  Bitcoin = c(bitcoin_mean, bitcoin_median, bitcoin_range, bitcoin_sd))
# Displaying comparison table
print(comparison_table)
```

```
##              Measure S_and_P_500  Bitcoin
## 1               Mean   3867.8815 27097.76
## 2             Median   3971.1800 25576.39
## 3              Range   3016.9500 68203.62
## 4 Standard Deviation    643.1942 16711.71
```

```
# Combining data into a single dataframe for box plot
data_combined <- data.frame(
  Dataset = c(rep("S&P 500", length(sp500_data$Price)), rep("Bitcoin", length(bitcoin_data_cl
ean$Close.price.adjusted))),
  Price = c(sp500_data$Price, bitcoin_data_clean$Close.price.adjusted)
)

# Creating box plot
library(ggplot2)

ggplot(data_combined, aes(x = Dataset, y = Price, fill = Dataset)) +
  geom_boxplot() +
  labs(title = "Comparison of S&P 500 and Bitcoin Data",
       x = "Dataset", y = "Price") +
  theme_minimal()
```



Comparison and Analysis:

1. **Central Tendency**:
    - **Mean and Median**: The mean and median prices of Bitcoin ($27097.76 and $25576.39, respectively) are significantly higher than those of the S&P 500 index ($3867.881 and $3971.18, respectively). This suggests that Bitcoin prices have a higher average and middle value compared to the S&P 500 index prices.
2. **Variability**:
    - **Range and Standard Deviation**: Bitcoin exhibits much higher variability in prices compared to the S&P 500 index. The range of Bitcoin prices ($68203.62) is substantially larger than that of the S&P 500 index ($3016.95). Similarly, the standard deviation of Bitcoin prices ($16711.71) indicates greater dispersion or volatility compared to the S&P 500 index ($643.1942).

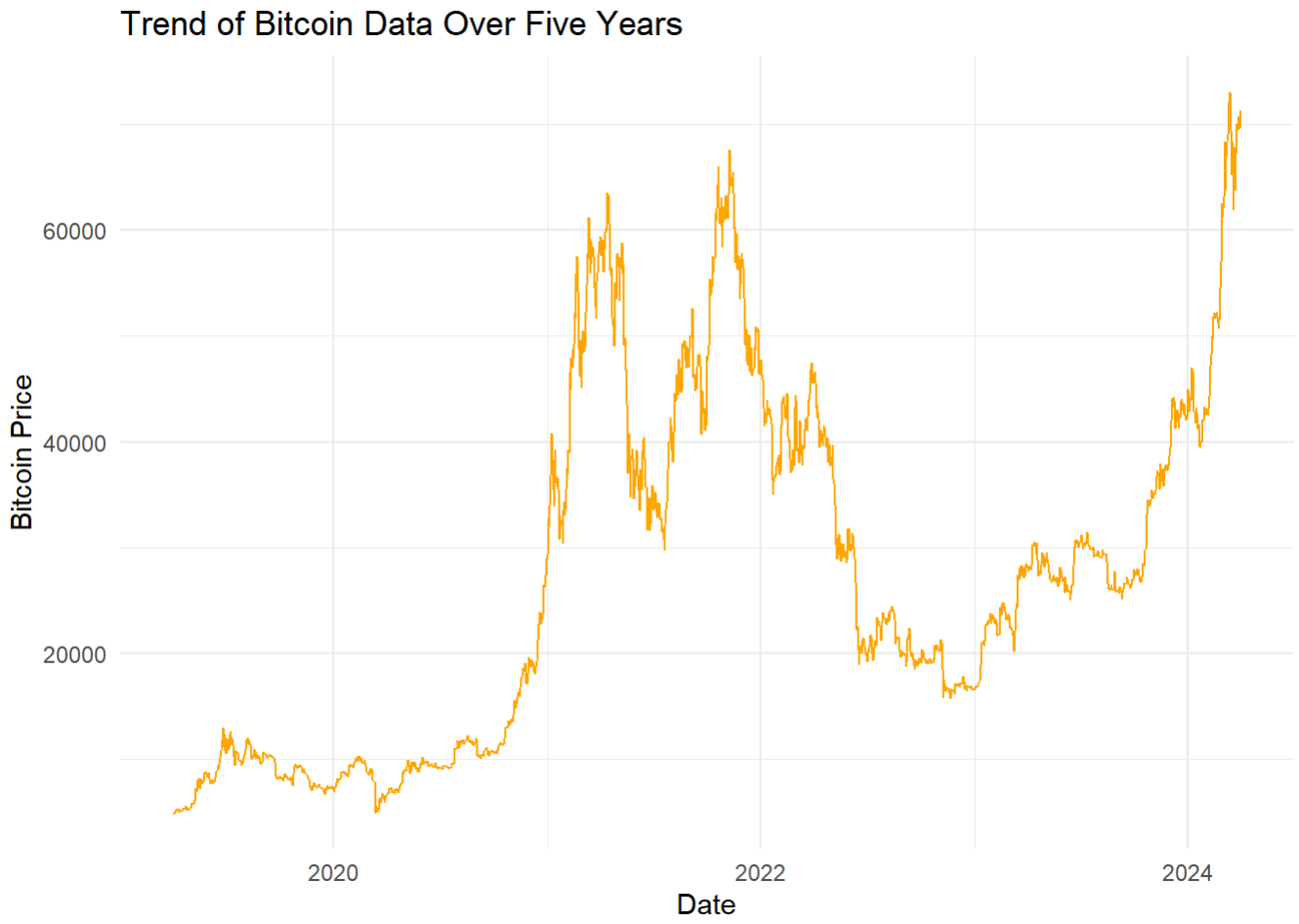
Conclusion:

- The descriptive statistics highlight significant differences between the S&P 500 index and Bitcoin in terms of average prices, price distributions, and volatility. Bitcoin prices demonstrate higher variability and a wider range of values compared to the more stable and narrower range observed in the S&P 500 index prices.

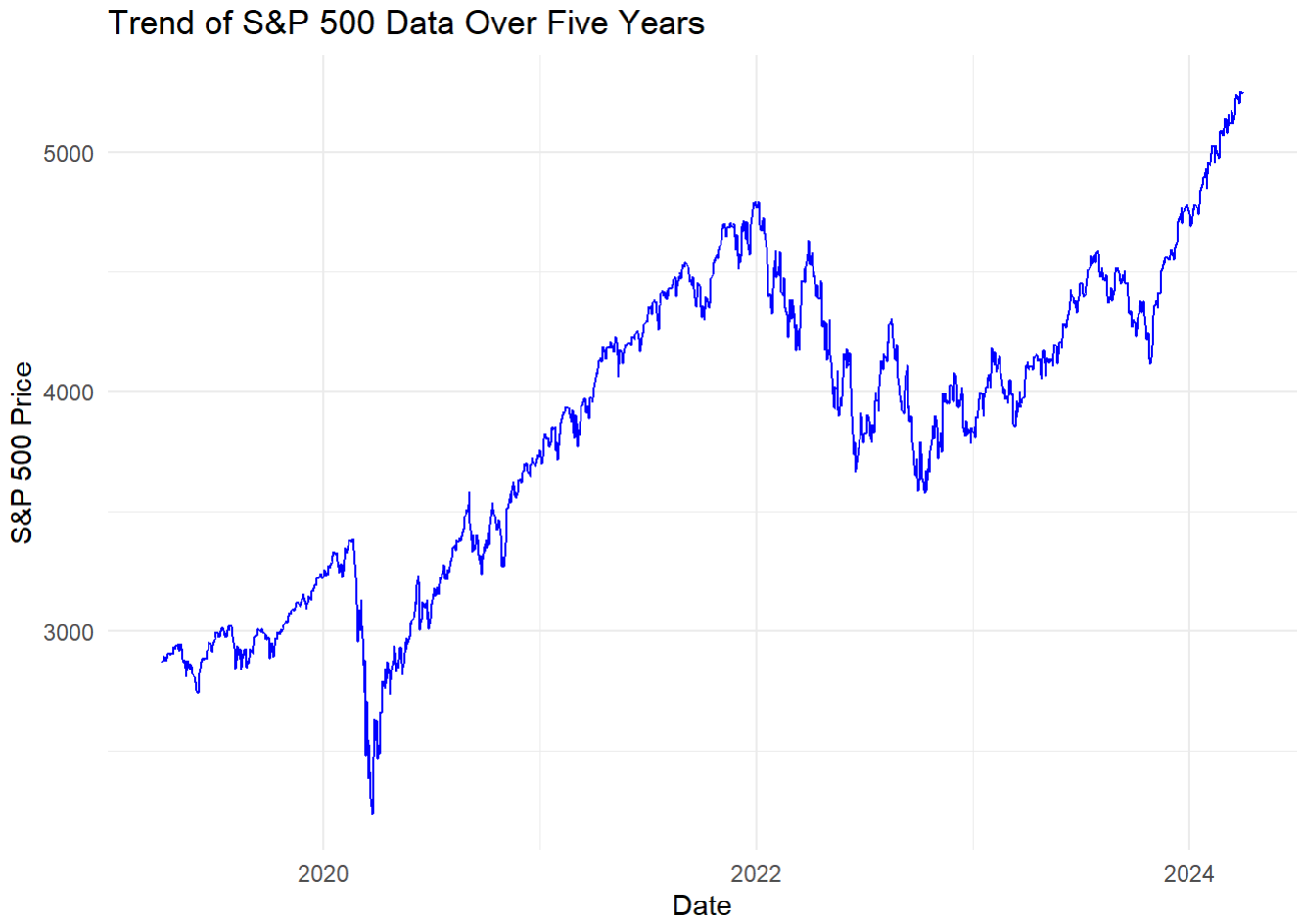
Task 2

Finding Trend or Pattern

```
library(ggplot2)
ggplot(bitcoin_data_clean, aes(x = Date, y = Close.price.adjusted)) +
  geom_line(color = "orange") +
  labs(title = "Trend of Bitcoin Data Over Five Years",
       x = "Date", y = "Bitcoin Price") +
  theme_minimal()
```




```
# Plotting S&P 500 data
ggplot(sp500_data, aes(x = Date, y = Price)) +
  geom_line(color = "blue") +
  labs(title = "Trend of S&P 500 Data Over Five Years",
       x = "Date", y = "S&P 500 Price") +
  theme_minimal()
```



INFERENCE: From the trend pattern of both the datasets , we can say that it is a increasing trend . The Bitcoin datasets shows an unpredictable variance pattern and a systematic increase in the end . The S&P dataset shows more of a seasonal pattern with a increasing pattern.

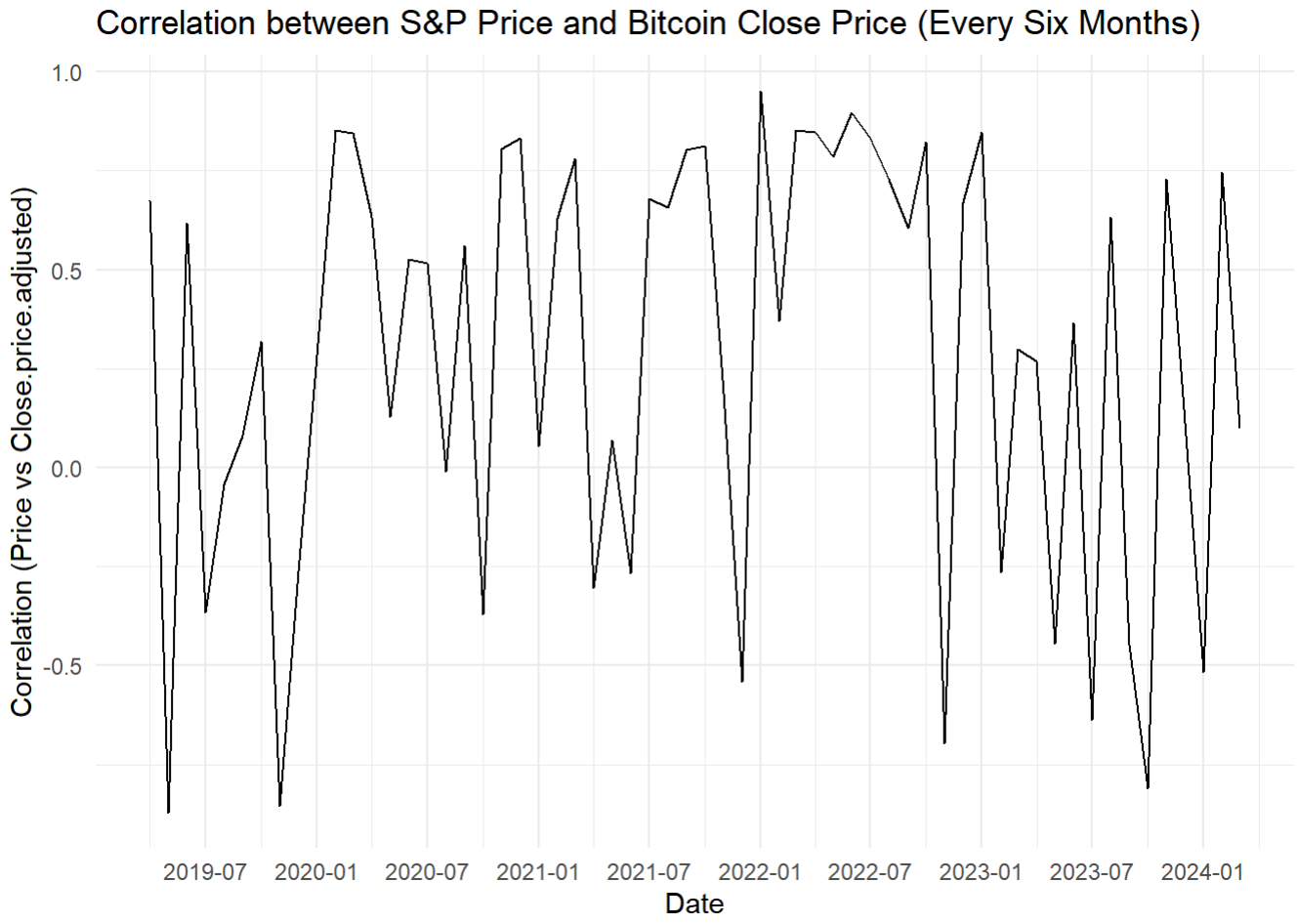
Correlation

```
# Convert 'Date' column to Date type
merged_data <- merge(sp500_data, bitcoin_data_clean, by = "Date", all = FALSE)
merged_data$Date <- as.Date(merged_data$Date, format = "%Y-%m-%d")
merged_data <- merged_data %>%
  mutate(YearMonth = format(Date, "%Y-%m")) %>%
  group_by(YearMonth) %>%
  summarise(correlation = cor(Price, Close.price.adjusted))
merged_data <- na.omit(merged_data)
```

```
install.packages("ggplot2")
```

```
## Warning: package 'ggplot2' is in use and will not be installed
```

```
library(ggplot2)
# Creating plot
# Ploting the correlation over time
ggplot(merged_data, aes(x = as.Date(paste0(YearMonth, "-01")), y = correlation)) +
  geom_line() +
  labs(
    x = "Date",
    y = "Correlation (Price vs Close.price.adjusted)",
    title = "Correlation between S&P Price and Bitcoin Close Price (Every Six Months)"
  ) +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "6 months") +
  theme_minimal()
```



INFERENCE:From the above plot we can observe that there is somewhat like a cyclic pattern repeating over the period of time in the span of 5 years .

Task 3

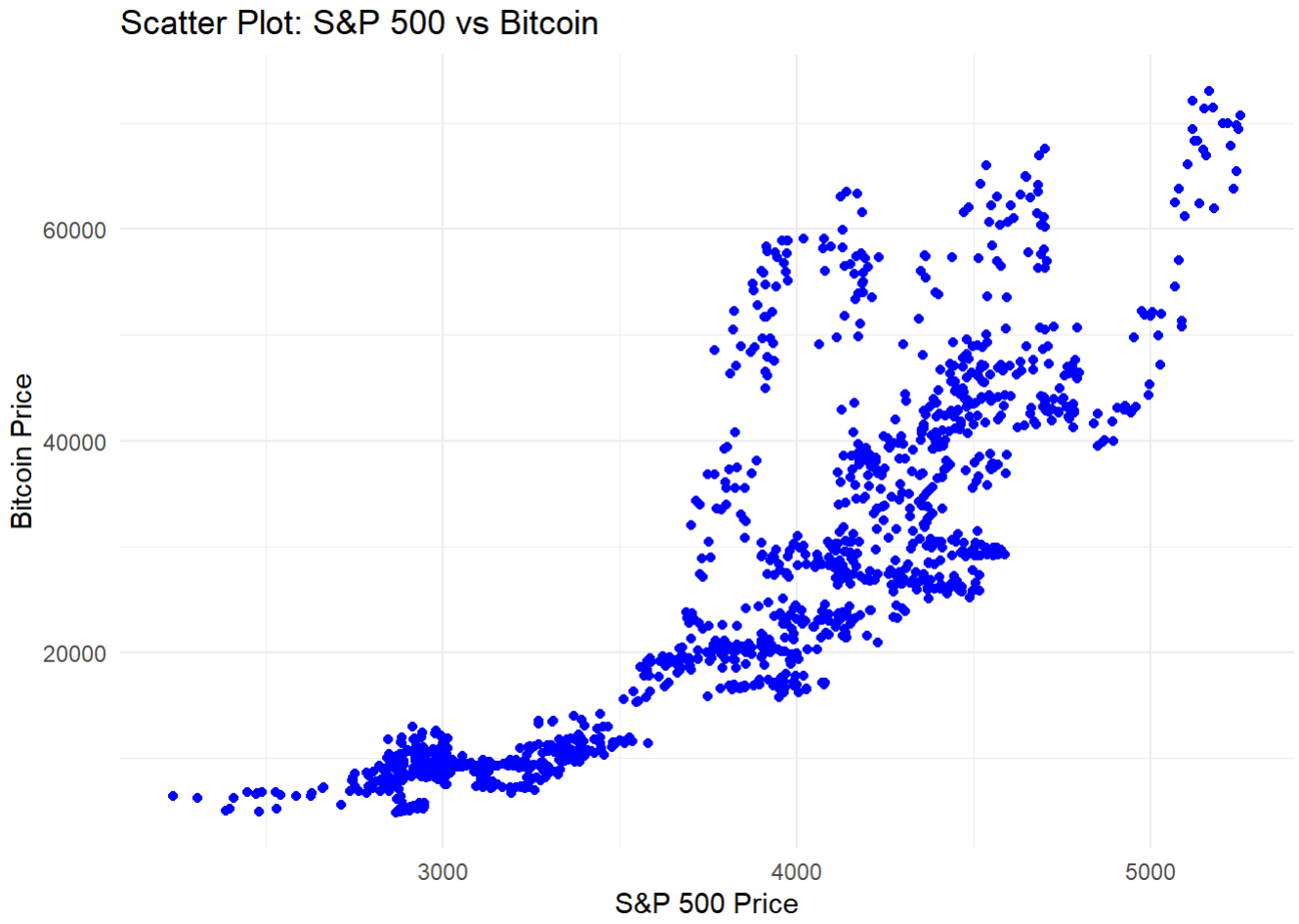
Correlation Coefficient

```
merged_data2 <- merge(sp500_data, bitcoin_data_clean, by = "Date", all = FALSE)
# Calculate correlation coefficient
correlation_coefficient <- cor(as.numeric(gsub(",", "", merged_data2$Price)), merged_data2$Cl
ose.price.adjusted, use = "pairwise.complete.obs")
# Print correlation coefficient
print(paste("Correlation Coefficient:", round(correlation_coefficient, 3)))
```

```
## [1] "Correlation Coefficient: 0.845"
```

NOTE: A correlation coefficient of 0.845 which is close to +1 between S&P data and Bitcoin data indicates a strong positive linear relationship between these two datasets. The value of r is 0.845 is relatively high, suggesting a substantial degree of association between S&P data and Bitcoin data. This indicates that changes in S&P prices explain a large proportion of the variability in Bitcoin prices.

```
#Visualizing the correlation using scatter plot
ggplot(merged_data2, aes(x = as.numeric(gsub(",", "", Price)), y = Close.price.adjusted)) +
  geom_point(color = "blue") +
  labs(title = "Scatter Plot: S&P 500 vs Bitcoin",
       x = "S&P 500 Price", y = "Bitcoin Price") +
  theme_minimal()
```



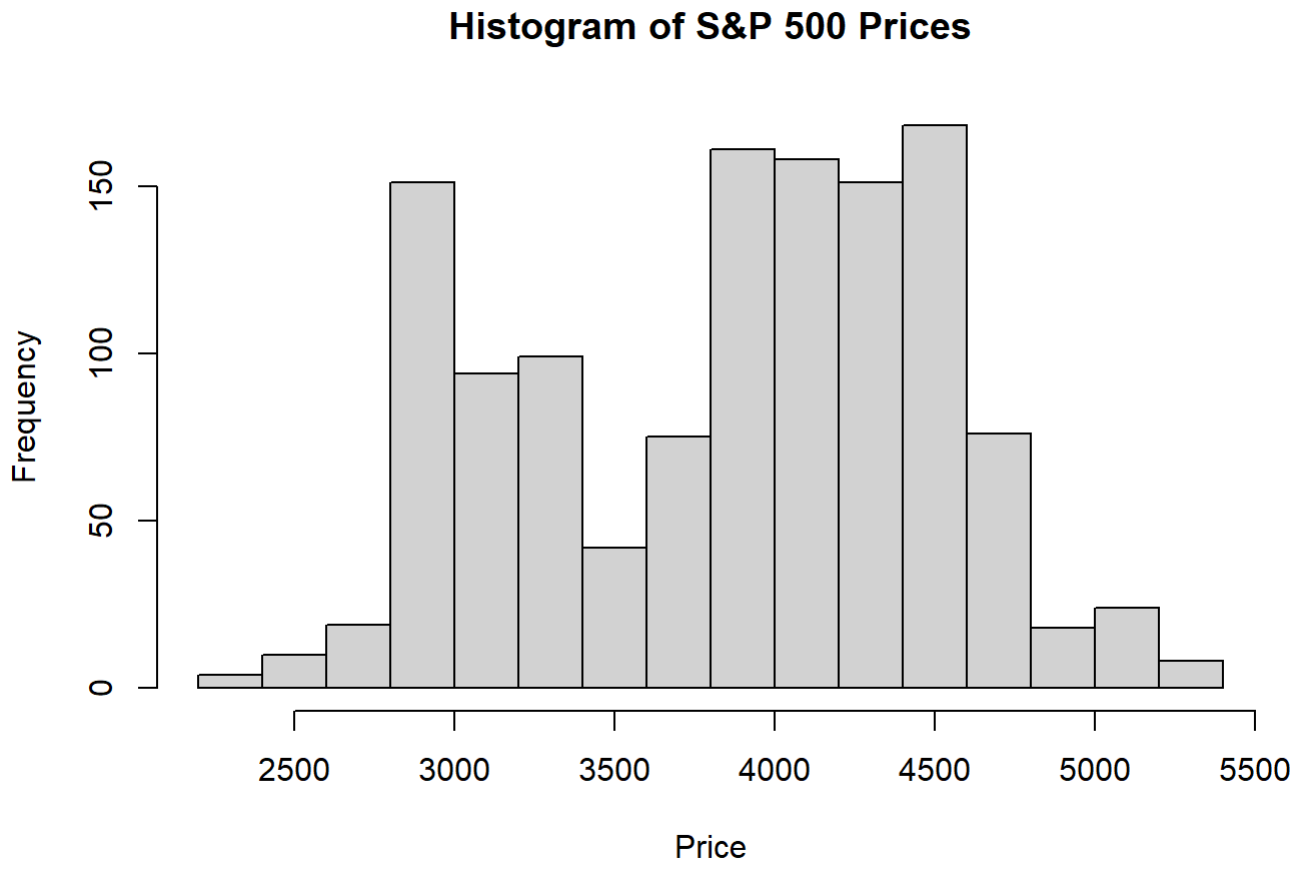
Inference

- The scatter plot shows an overall increasing trend, indicating a positive relationship between S&P 500 price and Bitcoin price.As the S&P 500 price increases, there is a tendency for the Bitcoin price to also increase.
- The plot reveals that the majority of data points are concentrated towards the lower end of the S&P 500 price range and lower Bitcoin prices.This suggests that most observations occur when the S&P 500 price is relatively lower, with a wider range of Bitcoin prices.
- At higher S&P 500 prices, the distribution of Bitcoin prices becomes more scattered and less dense.This implies that while there is a positive trend overall, the relationship between S&P 500 and Bitcoin prices may be less predictable or consistent at higher price levels.

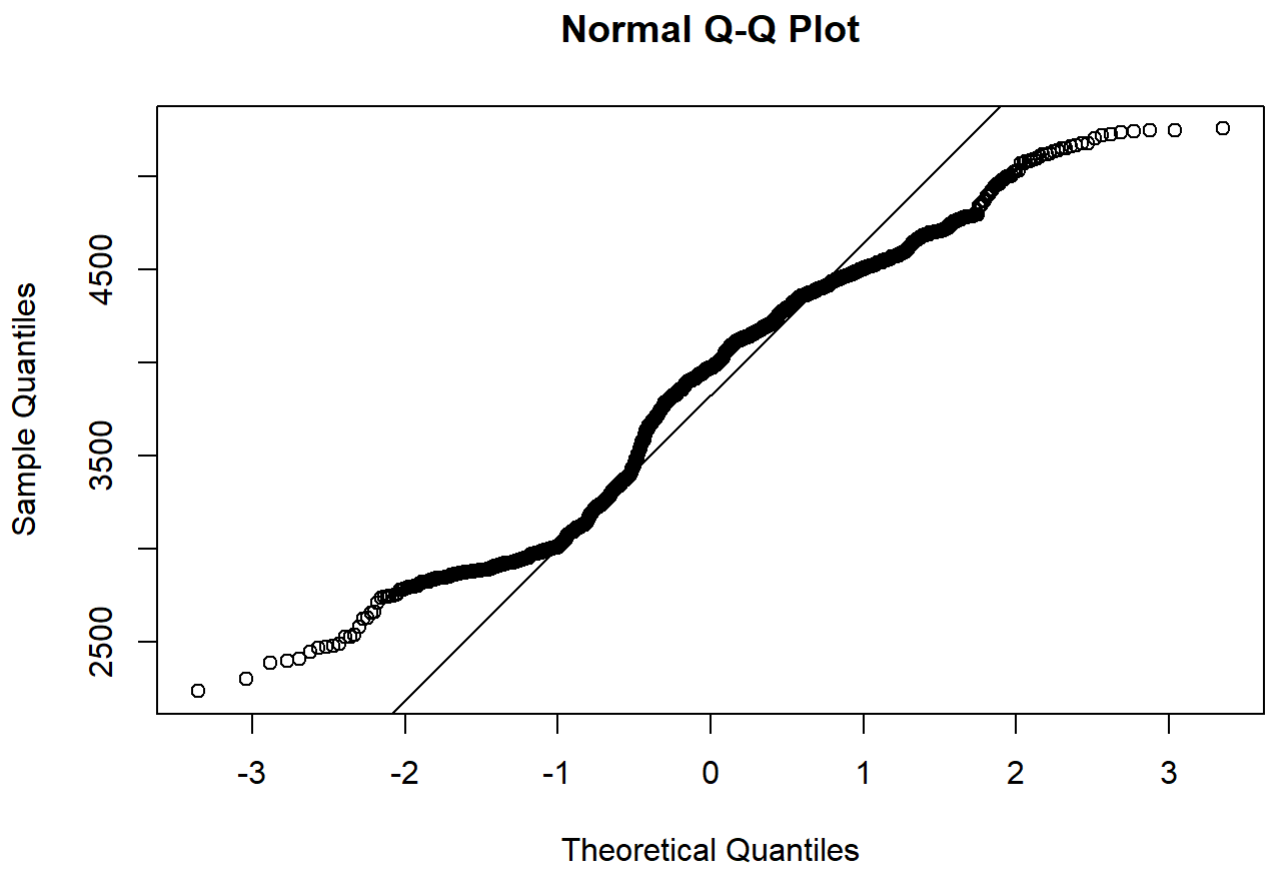
Task 4

Normal Distribution

```
# Plot histogram
hist(sp500_data$Price, breaks = 20, main = "Histogram of S&P 500 Prices",
     xlab = "Price", ylab = "Frequency")
```

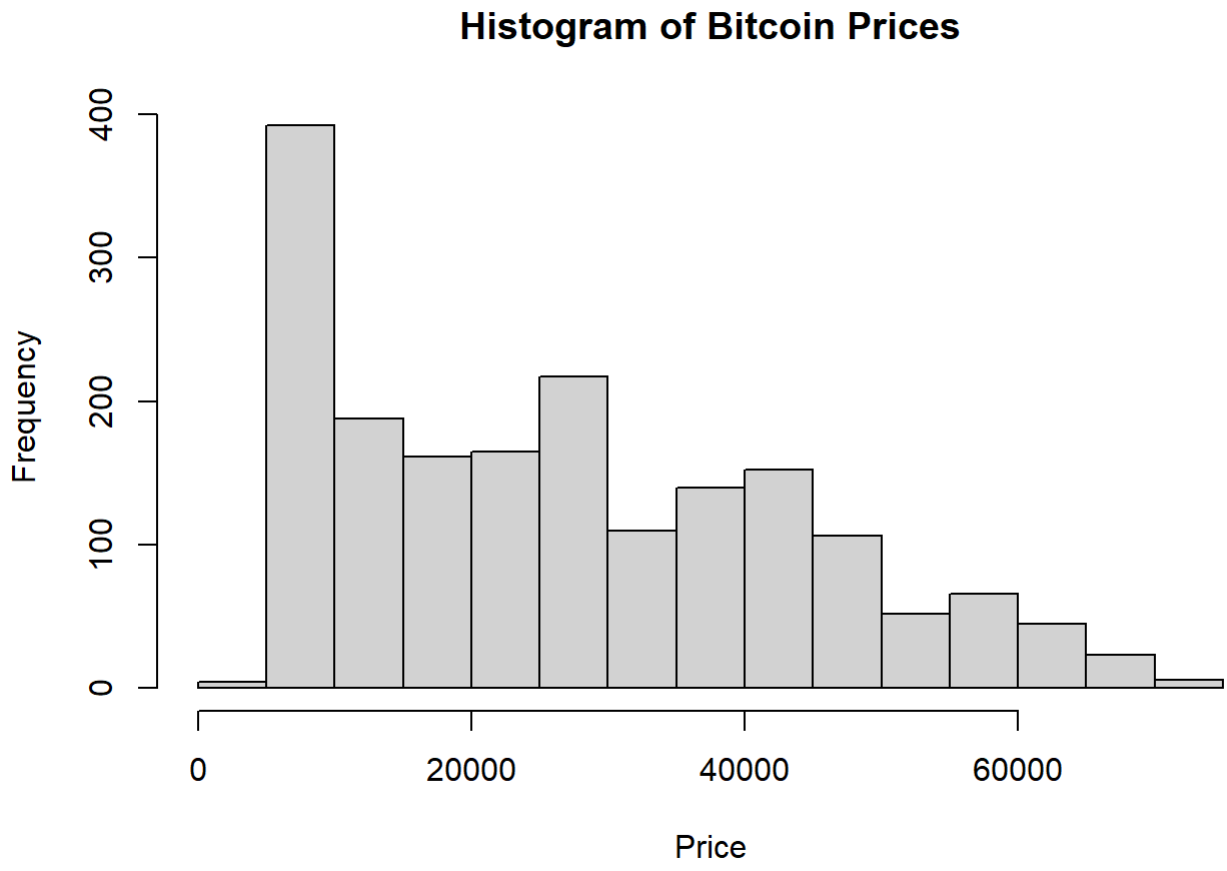



```
# QQ plot
qqnorm(sp500_data$Price)
qqline(sp500_data$Price)
```

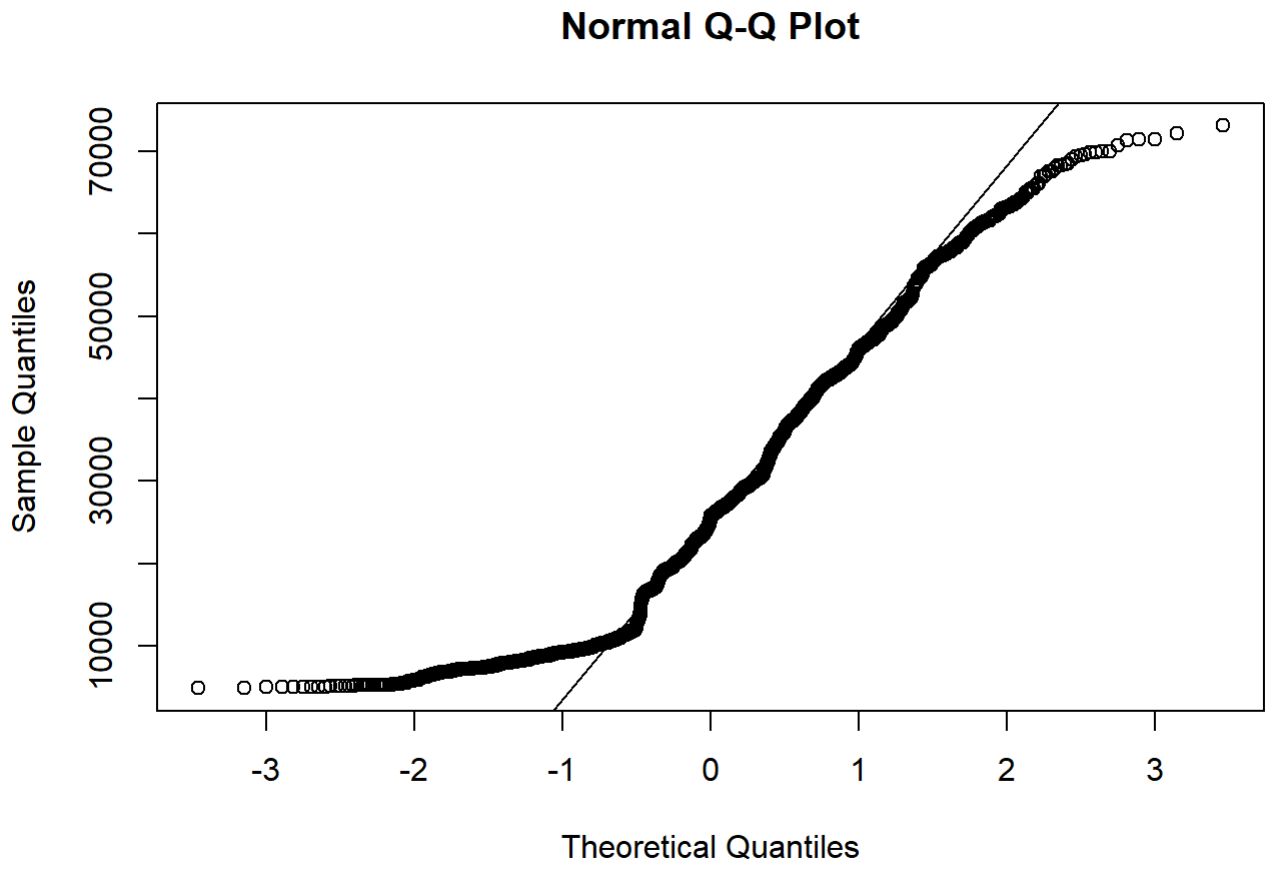


To assess the normal distribution , I have used the histogram and QQ line plots.
From the above graph the histogram plots has a bell curve pattern . The QQ plot is close to the straight line with little deviations . These indicate the the S&P dataset follows a normal distribution.

```
# Histogram of Bitcoin prices
hist(bitcoin_data_clean$Close.price.adjusted, breaks = 20, main = "Histogram of Bitcoin Price
s",
     xlab = "Price", ylab = "Frequency")
```




```
# Q-Q plot of Bitcoin prices against theoretical normal distribution
qqnorm(bitcoin_data_clean$Close.price.adjusted)
qqline(bitcoin_data_clean$Close.price.adjusted)
```



The histogram of this dataset has a symmetrical pattern . The QQ line plot is also very close to the straight which also indicates that this dataset also follows the normal distribution.

Conclusion : Both the dataset has a normal distribution data.

Reference

https://www.geeksforgeeks.org/data-visualization-in-r/ (https://www.geeksforgeeks.org/data-visualization-in-r/)

https://rkabacoff.github.io/datavis/Customizing.html (https://rkabacoff.github.io/datavis/Customizing.html)