Applied Analytics

Statistical analysis of Bitcoin and S&P 500 index

21st April 2024

Loading neccessary packages

```
library(kableExtra)
library(dplyr)
library(tidyverse)
library(ggplot2)
```

Student name and Number

Table 1: Student Information

Name	Student ID
Snigdha Mathur	S4017572

Introduction

The S&P 500, standing for Standard and Poor's 500 is a very reputed stock market index that captures the performance of 500 prominent publicly listed corporations in the United States. The S&P 500 index helps in providing knowledge about the US Stock market and economy. In common terms, S&P 500 is a stock market index which tracks the US equity market. Bitcoin is basically a decentralized cryptocurrency asset. Much like physical cash, Bitcoin allows for direct transactions without the need of governmental authorities to obtain permission to create an account, setting it apart from most modern electronic financial transactions.

In the report, we conduct a detailed study understanding the relationship between the S&P 500 stock index data and Bitcoin market data. The primary goal of the analysis is to to identify the past trends that existed between the index values and the fluctuating market value of Bitcoin over the years. The research aims to establish correlation between the fluctuation of the stock market with the digital currency, Bitcoin.

Methods

The BTC dataset records the end-of-day closing prices for the cryptocurrency : Bitcoin,, while the S&P 500 dataset details the daily closing figures of a major stock market index, reflecting the market's end-of-day valuation.

For analysing the datset, we use the tidyverse package which includes many data transformation packages like ggplot2, readr, dpylr and many more.
The dplyr package in R enables data manipulation, enhancing data manipulation for subsequent analysis. It includes functions such as mutate(), select(), filter(), summarise(), and arrange(), which streamline various data transformation processes. Similiary, the ggplot2 package supports the creation of diverse visualizations, including histograms, line charts, bar charts, and scatter plots, among others. This package is responsible in managing the aesthetics of graphs to ensure clear and effective data presentation.

Results

Loading the datasets using read.csv functions and storing in new variables 1. btc_data
2. sp500_data
Displaying the first few rows of each dataset to see the values present in the dataset

```
btc_data <- read_csv("BTC-USD.csv")
head(btc_data)
```

```
## # A tibble: 6 x 12
##   Date    `Close price adjusted` ...3  ...4  ...5  ...6  ...7  ...8  ...9  ...10
##   <chr>                   <dbl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl>
## 1 2/04/2~                 4880. NA    NA    NA    NA    NA    NA    NA    NA
## 2 3/04/2~                 4973. NA    NA    NA    NA    NA    NA    NA    NA
## 3 4/04/2~                 4923. NA    NA    NA    NA    NA    NA    NA    NA
## 4 5/04/2~                 5037. NA    NA    NA    NA    NA    NA    NA    NA
## 5 6/04/2~                 5060. NA    NA    NA    NA    NA    NA    NA    NA
## 6 7/04/2~                 5199. NA    NA    NA    NA    NA    NA    NA    NA
## # i 2 more variables: ...11 <lgl>, ...12 <lgl>
```

```
sp500_data <- read_csv("S&P 500.csv")
head(sp500_data)
```

```
## # A tibble: 6 x 2
##   Date      Price
##   <chr>     <dbl>
## 1 2/04/2019 2867.
## 2 3/04/2019 2873.
```

```
## 3 4/04/2019 2879.
## 4 5/04/2019 2893.
## 5 8/04/2019 2896.
## 6 9/04/2019 2878.
```

Checking the structure of the datasets for understanding the value attributes

```
str(btc_data)
```

```
## spc_tbl_ [1,827 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date                : chr [1:1827] "2/04/2019" "3/04/2019" "4/04/2019" "5/04/2019" ...
##  $ Close price adjusted: num [1:1827] 4880 4973 4923 5037 5060 ...
##  $ ...3                : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...4                : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...5                : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...6                : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...7                : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...8                : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...9                : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...10               : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...11               : logi [1:1827] NA NA NA NA NA NA ...
##  $ ...12               : logi [1:1827] NA NA NA NA NA NA ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_character(),
##   ..   `Close price adjusted` = col_double(),
##   ..   ...3 = col_logical(),
##   ..   ...4 = col_logical(),
##   ..   ...5 = col_logical(),
##   ..   ...6 = col_logical(),
##   ..   ...7 = col_logical(),
##   ..   ...8 = col_logical(),
##   ..   ...9 = col_logical(),
##   ..   ...10 = col_logical(),
##   ..   ...11 = col_logical(),
##   ..   ...12 = col_logical()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(sp500_data)
```

```
## spc_tbl_ [1,258 x 2] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date : chr [1:1258] "2/04/2019" "3/04/2019" "4/04/2019" "5/04/2019" ...
##  $ Price: num [1:1258] 2867 2873 2879 2893 2896 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_character(),
##   ..   Price = col_number()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

From the output, we get to know that :
1. BTC dataset only has values present in 2 columns. The rest 10 columns are NA, hence we shall clean the unwanted columns for thr ease of analysis.
2. The date columns in BTC and S&P500 dataset are of character type. Converting the date columns to date format (yyyy/mm/dd)

3. The price column in S&P500 dataset contains prices in a string format possibly due to commas. Converting the price column to a numeric type after removing the commas.

```
btc_data <- btc_data %>% select(Date, Close_price_adjusted = `Close price adjusted`)

# Convert Date columns to Date type
btc_data$Date <- as.Date(btc_data$Date, format = "%d/%m/%Y")
sp500_data$Date <- as.Date(sp500_data$Date, format = "%d/%m/%Y")

# Converting the Price column in S&P 500 dataset to numeric
sp500_data$Price <- as.numeric(gsub(",", "", sp500_data$Price))
```

Re-checking the structure of the datasets

```
str(btc_data)
```

```
## tibble [1,827 x 2] (S3: tbl_df/tbl/data.frame)
##  $ Date                : Date[1:1827], format: "2019-04-02" "2019-04-03" ...
##  $ Close_price_adjusted: num [1:1827] 4880 4973 4923 5037 5060 ...
```

```
str(sp500_data)
```

```
## spc_tbl_ [1,258 x 2] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Date : Date[1:1258], format: "2019-04-02" "2019-04-03" ...
##  $ Price: num [1:1258] 2867 2873 2879 2893 2896 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Date = col_character(),
##   ..   Price = col_number()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

The str() function confirms that the date columns are now in correct format and the unwanted columns in BTC data have been deleted.

TASK 1.

Descriptive statistic helps in summarizing the characteristics of the database It consists of three types of measures: Measures of central tendency, Measures of variability, and Frequency distribution.
1. Measures of central tendency describe the middle of the data set and includes mean, median and mode.
2. Measures of variability describe the how the datapoints are spread and includes variance and standard deviation.
3. Measures of frequency distribution describes the how often the data occurs and hence includes the count of dataset

```
# Defining a function to calculate the mode
calculate_mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

btc_stats <- btc_data %>%
  summarize(
    Mean = mean(Close_price_adjusted, na.rm = TRUE),
    Median = median(Close_price_adjusted, na.rm = TRUE),
    Mode = calculate_mode(btc_data$Close_price_adjusted),
    SD = sd(Close_price_adjusted, na.rm = TRUE),
    Var = var(Close_price_adjusted, na.rm = TRUE),
```

```
    Count = sum(!is.na(Close_price_adjusted)),
    Max = max(Close_price_adjusted, na.rm = TRUE),
    Min = min(Close_price_adjusted, na.rm = TRUE)
  )

sp500_stats <- sp500_data %>%
  summarize(
    Mean = mean(Price, na.rm = TRUE),
    Median = median(Price, na.rm = TRUE),
    Mode = calculate_mode(sp500_data$Price),
    SD = sd(Price, na.rm = TRUE),
    Var = var(Price, na.rm = TRUE),
   Count = sum(!is.na(Price)),
   Max = max(Price, na.rm = TRUE),
   Min = min(Price, na.rm = TRUE)

  )

comparison <- bind_rows(BTC = btc_stats, SP500 = sp500_stats)
rownames(comparison) <- c("BTC-USD", "S&P 500")
comparison
```

```
## # A tibble: 2 x 8
##     Mean Median  Mode     SD        Var Count    Max   Min
## *  <dbl>  <dbl> <dbl>  <dbl>      <dbl> <int>  <dbl> <dbl>
## 1 27098. 25576. 4880. 16712. 279281159.  1827 73084. 4880.
## 2  3868.  3971. 2926.   643.    413699.  1258  5254. 2237.
```

From the above output, we can state the following observations :

Central Tendency
BTC USD - The mean price of Bitcoin is higher than the median, indicating a right-skewed distribution of data. This means that there are times with high BTC prices that increase the mean greater than the median
S&P 500 - The median is higher than the mean, indicating a slightly left-skewed distribution.

Variability
BTC USD - The standard deviation of the dataset is very high at $16,712, which means that Bitcoin is extremly volatile in nature.
S&P 500 - Compared to Bitcoin, the S&P 500 has a lower standard deviation, indicating less day-to-day price volatility.

Frequency Distribution
BTC-USD has higher number of observations compared to the S&P 500. This difference might be due to diiference in time frame for data collection.

Overall, we can say that Bitcoin value is more volatile and has a higher average closing price compared to the S&P 500 during the time period.

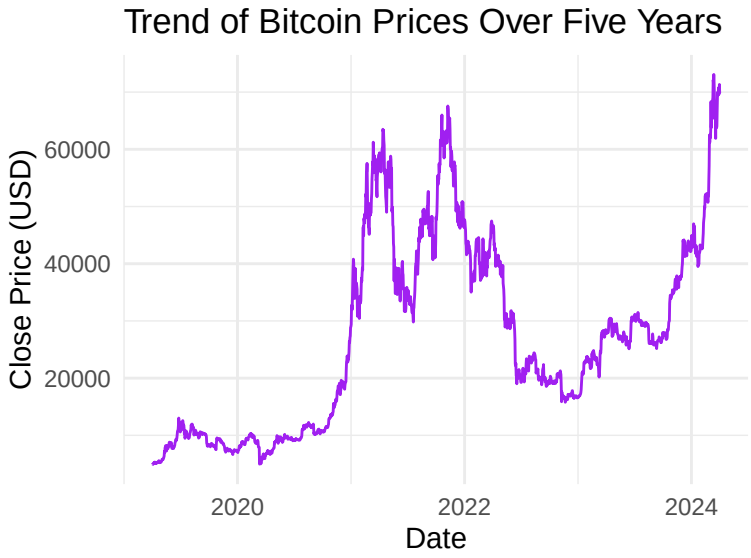
Insights
Trend Analysis: For Bitcoin, examining periods of high spikes and sharp declines could provide insights into market value trends.
Investment : Since Bitcoin is more volatile in nature, investing in it might lead to a higher risk and potential

return. The S&P 500, being more stable, is likely more suitable for steady investors.

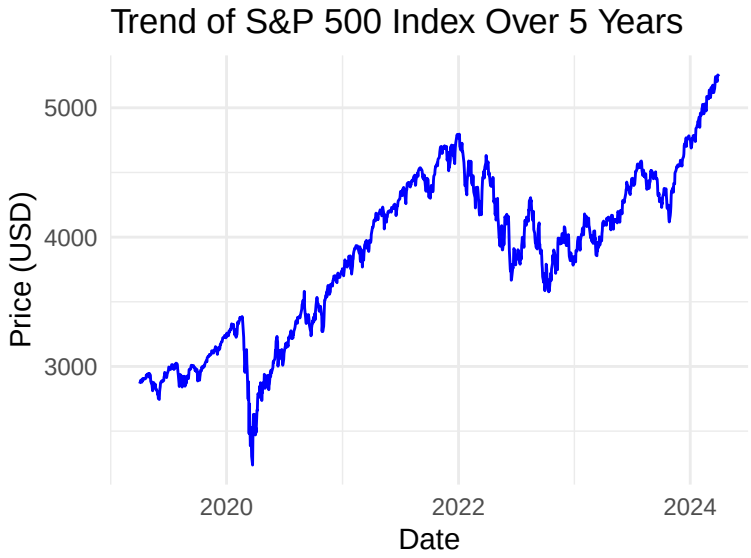
TASK 2.

Plotting graphs for both datasets for examing the trends of data over the past 5 years.

```
# Plot for BTC
btc_plot <- ggplot(btc_data, aes(x = Date, y = Close_price_adjusted)) +
  geom_line(color = "purple") +
  labs(title = "Trend of Bitcoin Prices Over Five Years",
       x = "Date",
       y = "Close Price (USD)") +
  theme_minimal()
btc_plot
```




```
# Plot for S&P 500
sp500_plot <- ggplot(sp500_data, aes(x = Date, y = Price)) +
  geom_line(color = "blue") +
  labs(title = "Trend of S&P 500 Index Over 5 Years",
       x = "Date",
       y = "Price (USD)") +
  theme_minimal()
sp500_plot
```



Looking at the plots for Bitcoin and S&P 500 index over the past 5 years, the following observations can be made:

1. BTC USD plot

   - The plots shows sharp increases with steep decreases.
   - There is a visible upward trend in the value of Bitcoin over the 5 year period. This suggests an overall increase in Bitcoin's price.
   - There are 2- 3 significant spikes, with the last spike pointing to the highest value on the plot. This indicates the presence of some external factors or scenarios which lead to increase in Bitcoin market value.
   - A consistent dip in the Bitcoin value is observed in the year of 2020 due to the COVID 19 pandemic.

2. S&P 500 plot

- The S&P 500 Index also shows an overall upward trend, indicating growth in the value of the index over the five-year period. Compared to Bitcoin, the growth is more steady and gradual over the period of time.
- The significant dip observed around early 2020 might be a result of global economic impact of the COVID-19 pandemic
  -Post the dip, the index showed gradual increase with time indicating the growth of US economic market value.

Now creating the plot showing the correlation calculated every 6 months between S&P 500 and Bitcoin data over the past 5 years, we shall need to perform the the steps
Step 1: Merge the dataset based on the date
Step 2: Creating segments of the combined dataset into intervals of 6 months
Step 3: Plotting Bitcoin closing prices against S&P 500 closing prices for each time period on the scatter plot.

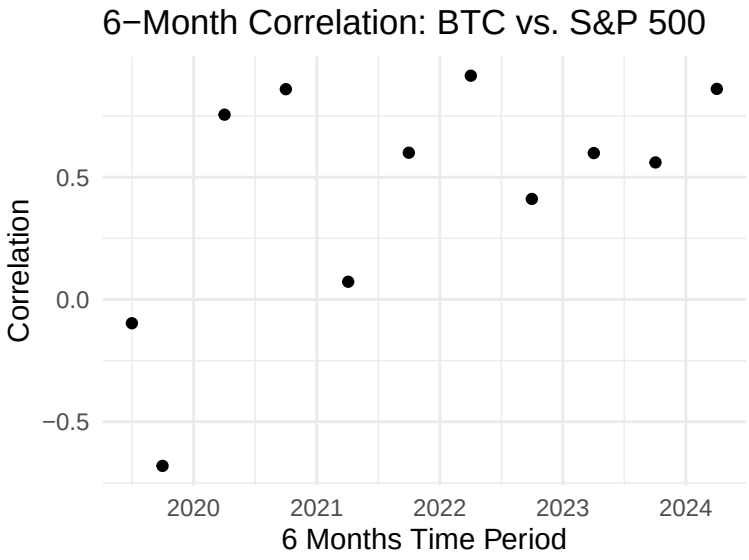
```
# Merging the datasets on the date column
combined_data <- inner_join(btc_data, sp500_data, by = "Date")

# Creating a six-month period identifier for each date
combined_data <- combined_data %>%
```

```r
  mutate(Period = floor_date(Date, "6 months"))

# Function to calculate correlation for each period
correlation_data <- combined_data %>%
  group_by(Period) %>%
  summarize(
    Correlation = cor(Close_price_adjusted, Price, use = "complete.obs"),
    Mid_Period = first(Date) + months(3)
  )


# Create a scatter plot with the mid-period on the x-axis and the correlation on the y-axis
ggplot(correlation_data, aes(x = Mid_Period, y = Correlation)) +
  geom_point() +  # Add points
  labs(title = "6-Month Correlation: BTC vs. S&P 500",
       x = "6 Months Time Period",
       y = "Correlation") +
  theme_minimal()
```



1. *Variability*

The scatter plot showing the correlation between Bitcoin and the S&P 500 indicates fluctuations over time in 6 months interval as teh correlation points do not form any straight line hence indicating that the relationship between these two markets is not constant

2. *Range of Correlation*

The correlation coefficients range from slightly negative to moderately positive, with values observed below 0, up to approximately 0.5. This implies that in some periods, the prices of Bitcoin and S&P 500 move in opposite directions, while in other periods they move together, but the strength of their relationship varies significantly.

TASK 3.

Correlation coefficient is the measure of closeness of the association to points represented in a scatter plot over a given period of time. For finding correlation coefficient of Btc and S&P dataset, we use the cor() function in R.

```r
# Calculating the correlation coefficient
correlation_coefficient <- cor(combined_data$Close_price_adjusted, combined_data$Price, use = "complete
```

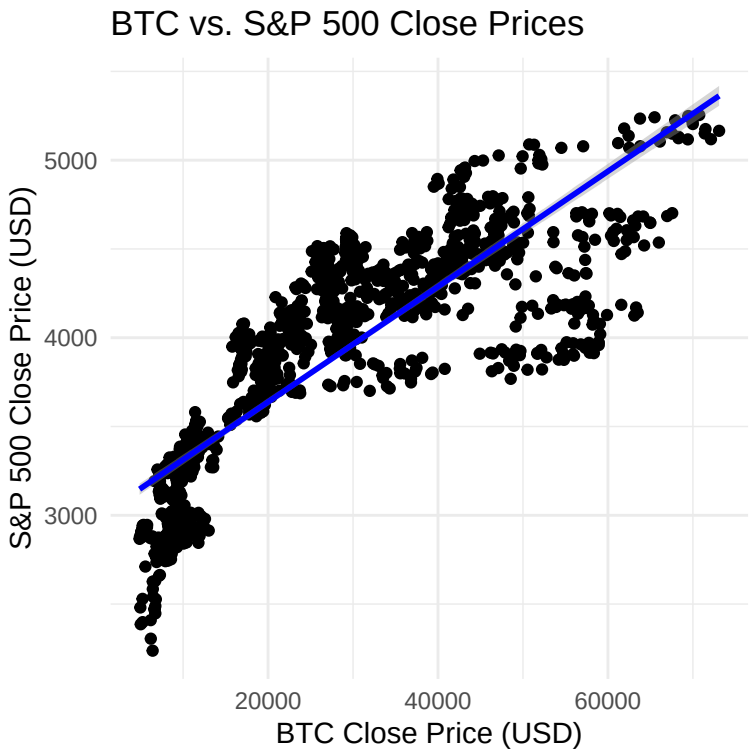
```r
# Print the correlation coefficient
print(paste("Correlation coefficient:", correlation_coefficient))
```

```
## [1] "Correlation coefficient: 0.844521873919683"
```

```r
# Create a scatter plot

ggplot(combined_data, aes(x = Close_price_adjusted, y = Price)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "BTC vs. S&P 500 Close Prices",
       x = "BTC Close Price (USD)",
       y = "S&P 500 Close Price (USD)") +
  theme_minimal()
```

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



1. The scatter plot reveals a positive relationship between BTC and S&P 500 closing prices because as the BTC value is increasing, the closing value of S&P also increases over the period of time. This proves an upward trend between the dataset
2. The relationship does not appear to be perfectly linear. While there's a general upward trend, the data points don't form a tight line, indicating that other factors might be affecting prices and their relation.
3. There's a noticeable concentration of data points in the lower left corner of the plot, which suggests that for a considerable period, both BTC and the S&P 500 traded at lower prices. As we move to the right (higher BTC prices), the S&P 500 prices also rise, but the spread of the points becomes wider, suggesting increased variability in the S&P 500 prices as BTC prices get higher.
4. The plot shows that there are some extreme values, particularly for BTC, which reach up to 60,000 USD. The S&P 500 also shows some high closing prices but less variability in the extremes compared to BTC.

TASK 4

A normal distribution is a continuous distribution with same values of mean, median and mode of the data set. The distribution displays a bell curve always symmetrical about the mean because of its flared shape.

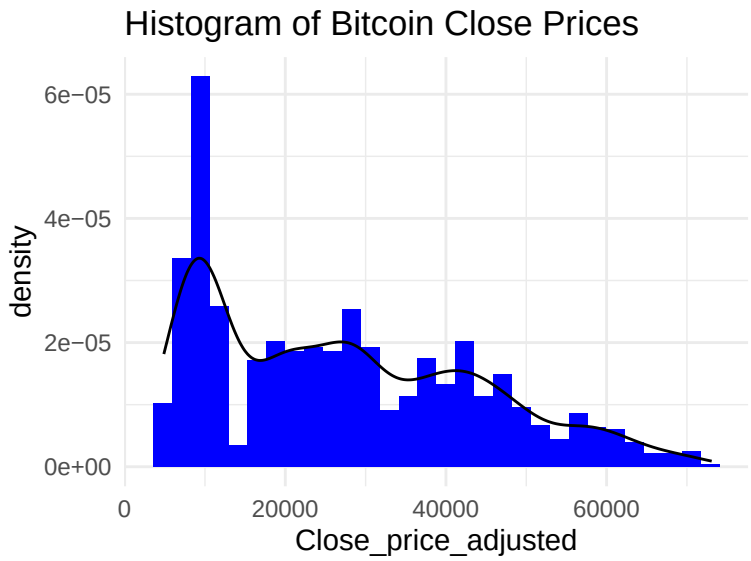
For checking if the BTC data and the S&P 500 index follow normal distribution, we can use the below mentioned methods :

1. isual representation by plotting histograms of the datapoints. A bell like shape of the histogram would mean presence of normal distribution.

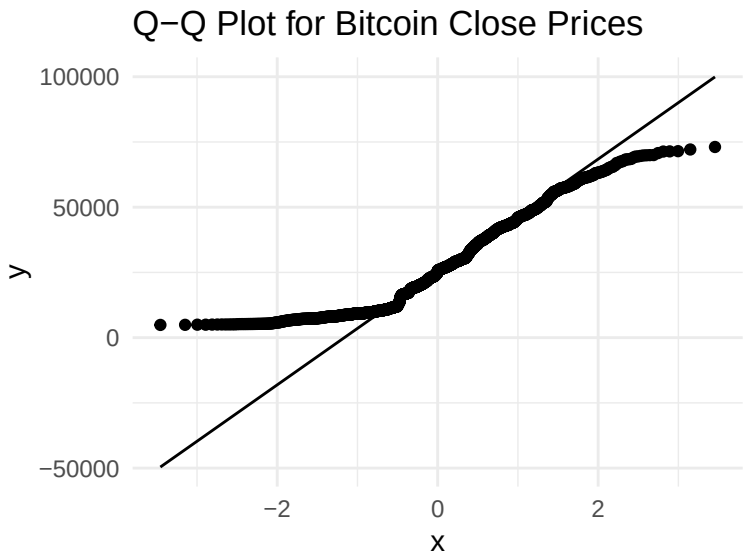
2. QQ Plots (Quartile - Quartile graphs) also help in checking normal distribution in a dataset. If the points in a QQ plot lie in a straight diagonal line , then the data can be assumed to be normally distributed.

3. Statistical methods like Shapiro Wilk test also test the presence of normal distribution. This test check for the p-value of dataset. If the value of p is equal to or less than 0.05, then the Shapiro test rejects normality in the data. On failing, the test can state that the data will not fit the distribution normally.

```
# Histogram and QQ Plot for Bitcoin
btc_hist <- ggplot(btc_data, aes(x = Close_price_adjusted )) +
  geom_histogram(aes(y = ..density..) , bins = 30, fill = "blue") +
  geom_density(color = "black") +
  labs(title = "Histogram of Bitcoin Close Prices") +
  theme_minimal()
btc_hist
```



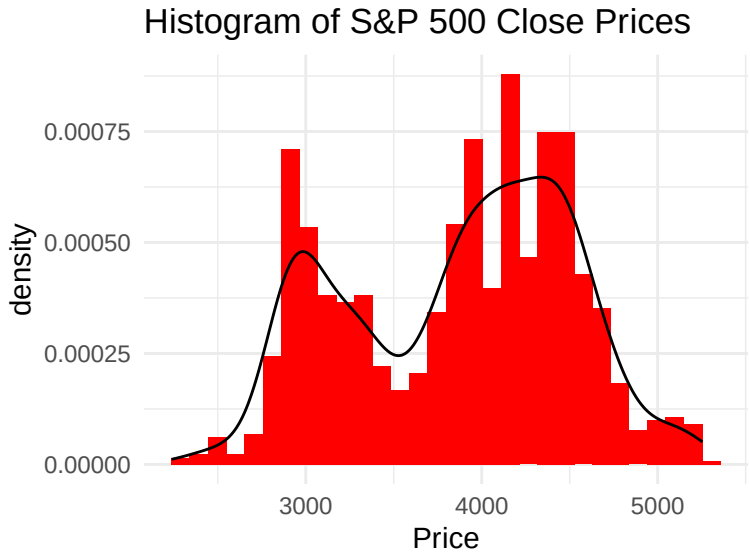

```
btc_qq <- ggplot(btc_data, aes(sample = Close_price_adjusted)) +
  stat_qq() +
  stat_qq_line() +
  labs(title = "Q-Q Plot for Bitcoin Close Prices")+
  theme_minimal()
btc_qq
```




```
#Shapiro-Wilk test for Bitcoin
btc_shapiro_test <- shapiro.test(btc_data$Close_price_adjusted)
btc_shapiro_test
```

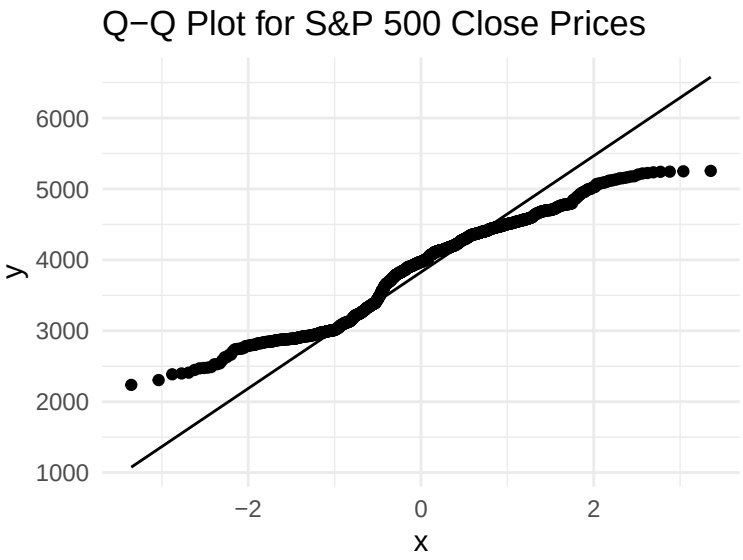
```
##
##  Shapiro-Wilk normality test
##
## data:  btc_data$Close_price_adjusted
## W = 0.93079, p-value < 2.2e-16
```

```
# Histogram and QQ plot for S&P 500
sp500_hist <- ggplot(sp500_data, aes(x = Price)) +
  geom_histogram(aes(y = ..density..), bins = 30, fill = "red") +
  geom_density(color = "black") +
  labs(title = "Histogram of S&P 500 Close Prices")+
  theme_minimal()
sp500_hist
```




```
sp500_qq <- ggplot(sp500_data, aes(sample = Price)) +
  stat_qq() +
```

```
  stat_qq_line() +
  labs(title = "Q-Q Plot for S&P 500 Close Prices") +
  theme_minimal()
sp500_qq
```




```
#Shapiro-Wilk test for S&P 500
sp500_shapiro_test <- shapiro.test(sp500_data$Price)
sp500_shapiro_test
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sp500_data$Price
## W = 0.96053, p-value < 2.2e-16
```

1. BTC dataset : From the visualization methods, the histogram plotted for data distribution also shows that the distribution is not normally distributed as the curve is not bell shaped. Additionally with help of the QQ plot, we further verify that the data is not normally distributed as the datapoints in the QQ plot do not lie along the straight diagonal line.
   The Shapiro test also concludes the BTC dataset does not follow normal distribution as the value of P is significantly lower than 0.05

2. S&P 500 dataset :The histogram plot for the S&P500 index values states that the values are not normally distributed since the curve of the plot does not resonate with a bell curve - which is observed in normal distribution.
   Additionally, the QQ plot for the dataset also cleary shows that normal distribution is not being followed as datapoints do not align along the straight diagonal line.
   Similartly for the S&P index, the value of p is also lower than 0.05, rejecting the presence of normal distribution in the dataset

Hence, we can state that neither BTC dataset nor S&P 500 index dataset follow a normal distribution.

Conclusion

The detailed analysis of the Bitcoin (BTC-USD) and S&P 500 datasets over a 5 year time period has provided meaningful insights in understanding the relationship and behavious of the 2 financial values.

The correlation analysis revealed that Bitcoin and S&P 500 closing prices have a strong positive relationship, identified by the high correlation coefficient . The scatter plot visualization further confirmed the positive linear relationship, suggesting that the prices of Bitcoin and the S&P 500 have increased during the time period. However, while the correlation is notably high, the datasets displayed differing levels of price variability. Bitcoin displayed greater volatility and wider price fluctuations than the S&P 500, whereas S&P 500 dispkayed a more stable and gradual growth trajectory.

The distributional analysis, assessing normality, indicated that neither Bitcoin nor S&P 500 closing prices followed a normal distribution within the observed period. The fact that the price movements for both Bitcoin and the S&P 500 don't follow the expected bell curve pattern is a key point for financial strategies and calculations that usually rely on this pattern to predict risks and prices.

In conclusion, the findings suggest that while Bitcoin and the S&P 500 share a positive correlation during the analyzed time frame, they also exhibit distinct risk profiles and distributional characteristics. Investors considering these assets should consider the implications of their volatility, correlation, and non-normal distributions in their future decision-making processes in the market.

References

1. The Investopedia team (2023) *What Does the S&P 500 Index Measure and How Is It Calculated?* , Investopedia, accessed on 12 April 2024. https://www.investopedia.com/ask/answers/040215/what-does-sp-500-index-measure-and-how-it-calculated

2. The Investopedia team (2023) *How Bitcoin works* , Investopedia, accessed on 12 April 2024. https://www.investopedia.com/news/how-bitcoin-works

3. Bitcoin.com (2022) *What is Bitcoin* Bitcoin.com, accessed on 12 April 2024 https://www.bitcoin.com/get-started/what-is-bitcoin/

4. Datacamp (n.d.) *ggplot2 Cheat Sheet* Datacamp.com, accessed on 13 April 2024 https://www.datacamp.com/cheat-sheet/ggplot2-cheat-sheet

5. R handbook (n.d) *30 ggplot basics* The Epidemiologist R Handbook, accessed on 13 April 2024, https://epirhandbook.com/en/ggplot-basics.html#ggplot_basics_facet

6. Bob Rudis, Noam Ross and Simon Garnier (2024), *Introduction to the viridis color maps*, cran - r-project,accessed on 14 April 2024 https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html

7. Ramzi W. Nahhas (2023), *4.6 Merge (join)* , An Introduction to R for Research, accessed on 14 April 2024, https://bookdown.org/rwnahhas/IntroToR/merge-join.html

8. Zach Bobbitt (2021), *How to Test for Normality in R (4 Methods)* , Statology, accessed on 15 April 2024, https://www.statology.org/test-for-normality-in-r/

9. Tutorials Point (n.d.), *R - Mean, Median and Mode*, Tutorials Point, accesed on 13 April 2024 https://www.tutorialspoint.com/r/r_mean_median_mode.htm

10. Hadley Wickham, Romain François, Lionel Henry, Kirill Müller, Davis Vaughan (n.d.), *dpylr*, dpylr-tidyverse, accessed on 17 April 2024, https://dplyr.tidyverse.org/

11. Hadley Wickham, Winston Chang, Lionel Henry, Thomas Lin Pedersen, (n.d.), *ggplot2*, ggplot- tidyverse, accessed on 17 April 2024 https://ggplot2.tidyverse.org

12. James Baglin (2016), *Module 2 - Descriptive Statistics through Visualisation* , Applied Analytics, accessed on 10th April 2024, https://astral-theory-157510.appspot.com/secured/MATH1324_Module_02.html

13. James Baglin (2016), *Module 4 - Probability Distributions: Random, but Predictable* , Applied Analytics, accessed on 10th April 2024, https://astral-theory-157510.appspot.com/secured/MATH1324_Module_04.html

14. Prof. Laleh (2024) 'Week 05 Demo Slides' [PDF, MATH1324], RMIT University, Melbourne.

15. Rohit T (2021), *gsub() in R*, Scaler Topics, accessed on 12 April 2024, https://www.scaler.com/topics/gsub-r/

16. Datacamp (n.d.) *cor: Correlation, Variance and Covariance (Matrices)* R Documentation, accessed on 14 April 2024, https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/cor

17. GeeksForGeeks (n.d.) *Shapiro–Wilk Test in R Programming* GeeksForGeeks. accessed on 19 April 2024, https://www.geeksforgeeks.org/shapiro-wilk-test-in-r-programming/