MATH1324 Assignment 1

Applied Analytics

Tanishka Sahni (s4175835)

Loading Packages

```
library(dplyr)
```

```
##
## 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
```

```
library(ggplot2)
```

Implementing Data

```
sp500 <- read.csv("~/Downloads/S&P 500 Historical Data1825.csv", stringsAsFactors
= FALSE)
bitcoin <- read.csv("~/Downloads/Bitcoin Historical Data1825.csv", stringsAsFactor
s = FALSE)

sp500$Price <- as.numeric(gsub(",","",sp500[[2]]))
sp500$Date <- as.Date(sp500[[1]], format = "%m/%d/%Y")

bitcoin$Price <- as.numeric(gsub(",","",bitcoin[[2]]))
bitcoin$Date <- as.Date(bitcoin[[1]], format = "%m/%d/%Y")



sp500 <- sp500 %>% arrange(Date)
bitcoin <- bitcoin %>% arrange(Date)
```

#Performing Task-1

```
#Mean of both the things
mean(sp500$Price, na.rm = TRUE)
```

```
## [1] 4112.208
```

```
mean(bitcoin$Price, na.rm = TRUE)
```

```
## [1] 35568.82
```

```
#Median of both
median(sp500$Price, na.rm = TRUE)
```

```
## [1] 4110.745
```

```
median(bitcoin$Price, na.rm = TRUE)
```

```
## [1] 28002
```

```
#Mode
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}
getmode(sp500$Price)
```

```
## [1] 2914
```

```
getmode(bitcoin$Price)
```

```
## [1] 7208
```

```
#Range of both
range(sp500$Price, na.rm = TRUE)
```

```
## [1] 2237.40 6389.77
```

```
range(bitcoin$Price , na.rm = TRUE)
```

```
## [1]   3228.7 119965.5
```

```
#standard deviation
sd(sp500$Price, na.rm = TRUE)
```

```
## [1] 1010.757
```

```
sd(bitcoin$Price, na.rm = TRUE)
```

```
## [1] 28828
```

```
#variance
var(sp500$Price , na.rm = TRUE)
```

```
## [1] 1021630
```

```
var(bitcoin$Price, na.rm = TRUE)
```

```
## [1] 831053564
```

```
#summary of both
summary(sp500$Price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2237    3191    4111    4112    4672    6390
```

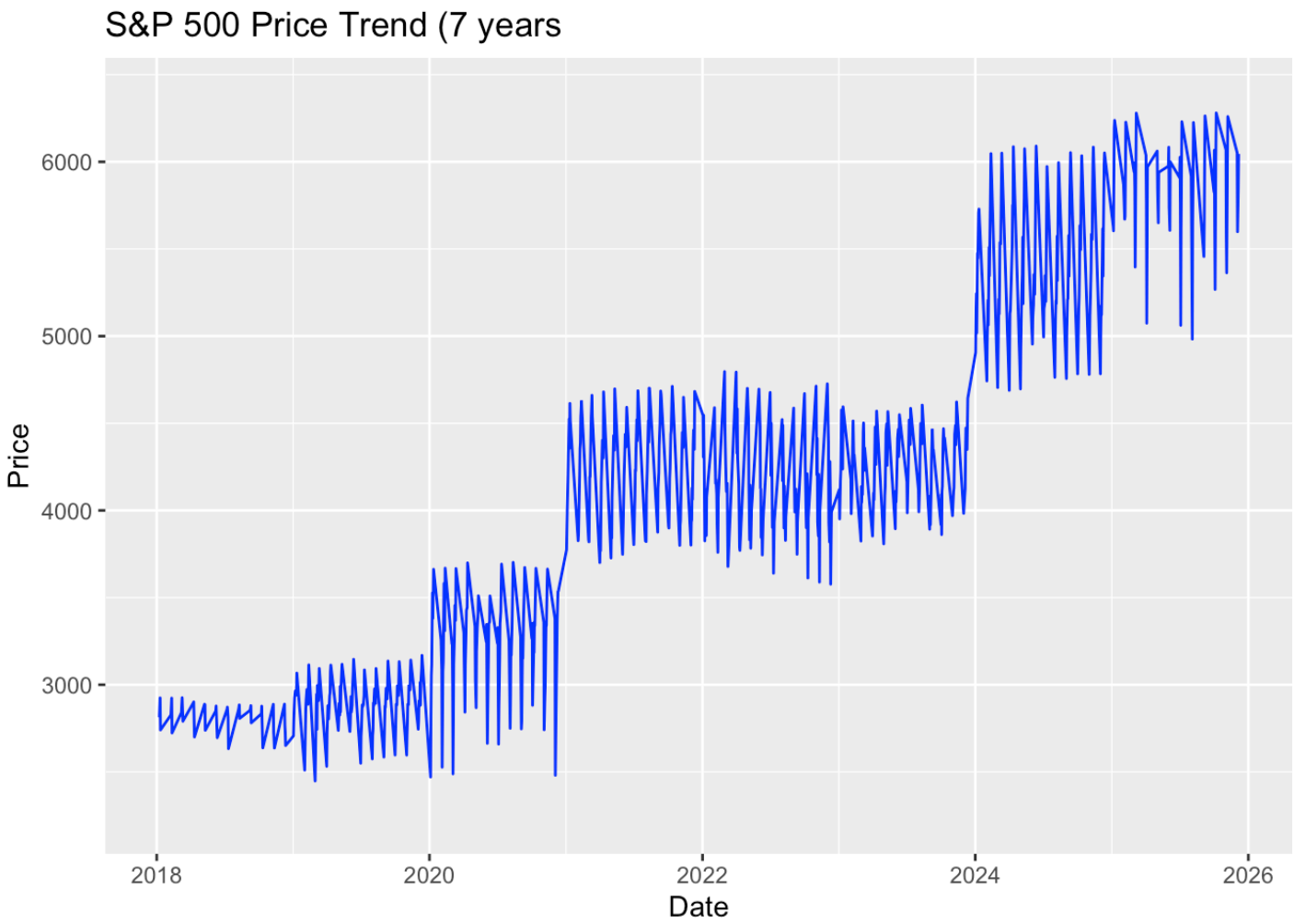
```
summary(bitcoin$Price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    3229    9983   28002   35569   53773  119966
```

#Performing Task-2

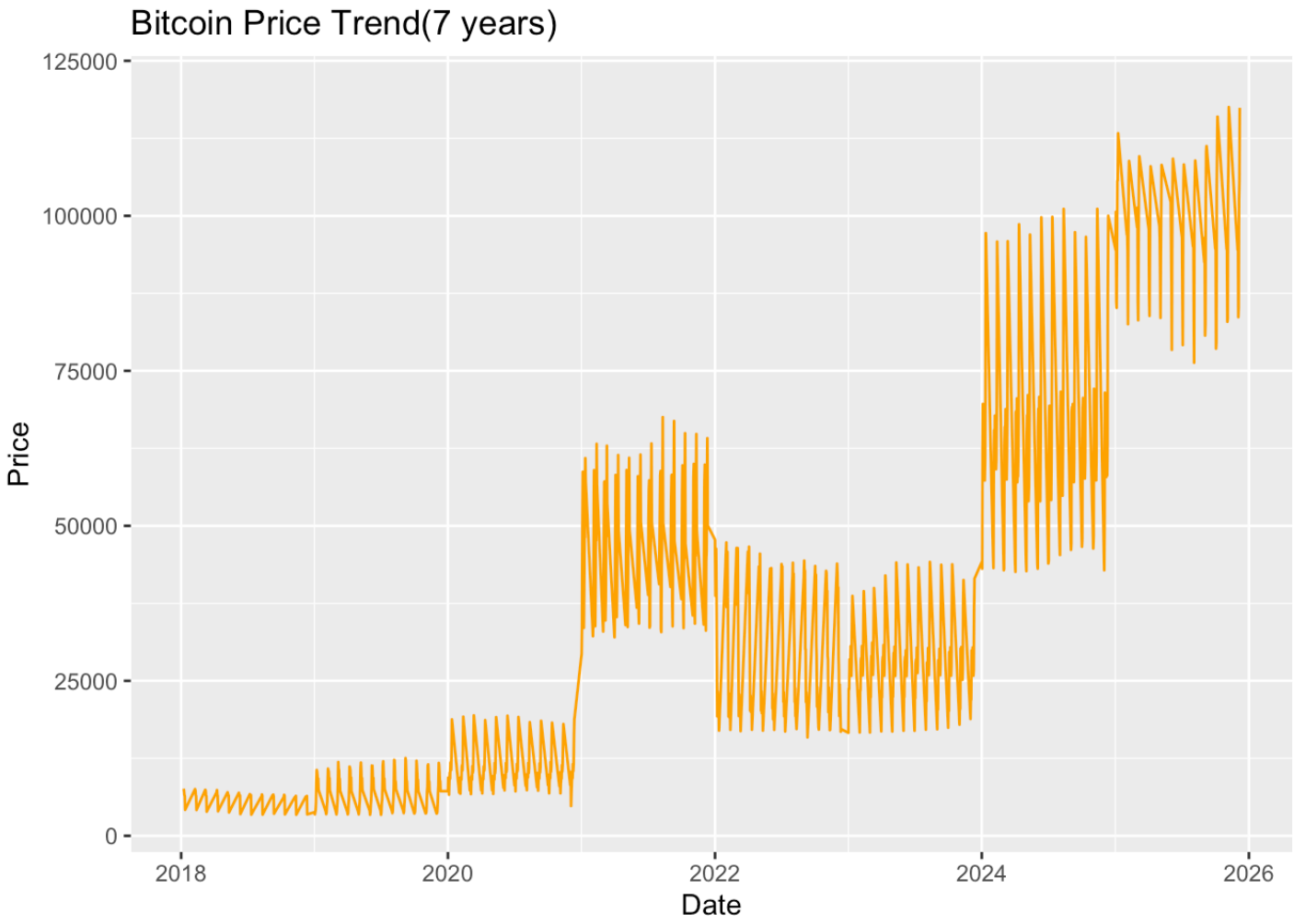
```
ggplot(sp500 , aes(x=Date, y=Price))+
  geom_line(color="blue")+
  labs(title = "S&P 500 Price Trend (7 years", x="Date", y="Price")
```

```
## Warning: Removed 1065 rows containing missing values or values outside the scal
e range
## (`geom_line()`).
```




```
ggplot(bitcoin, aes(x=Date, y=Price))+
  geom_line(color="orange")+
  labs(title = "Bitcoin Price Trend(7 years)", x="Date", y="Price")
```

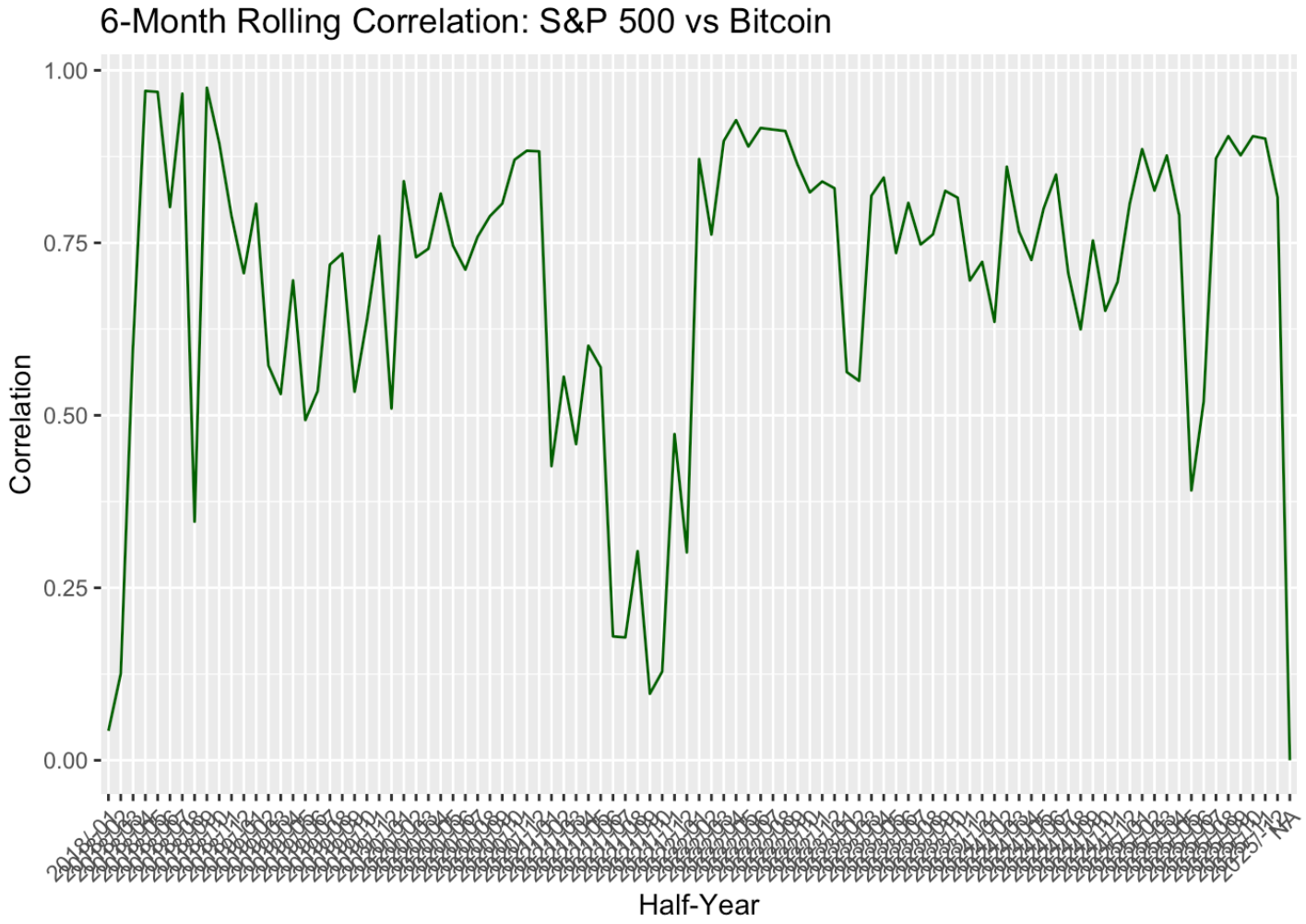
```
## Warning: Removed 1549 rows containing missing values or values outside the scal
e range
## (`geom_line()`).
```




```
merged <- merge(sp500, bitcoin, by="Date", suffixes = c("_sp500","_btc"))

merged$halfyear <- format(merged$Date, "%Y/-%m")
cor_halfyear <- merged %>%
  group_by(halfyear) %>%
  summarise(corr = cor(Price_sp500, Price_btc, use = "complete.obs"))

ggplot(cor_halfyear, aes(x=halfyear, y=corr, group = 1))+
  geom_line(color="darkgreen")+
  theme(axis.text.x = element_text(angle=45, hjust = 1))+
  labs(title = "6-Month Rolling Correlation: S&P 500 vs Bitcoin",
       x="Half-Year", y="Correlation")
```



#Performing Task-3

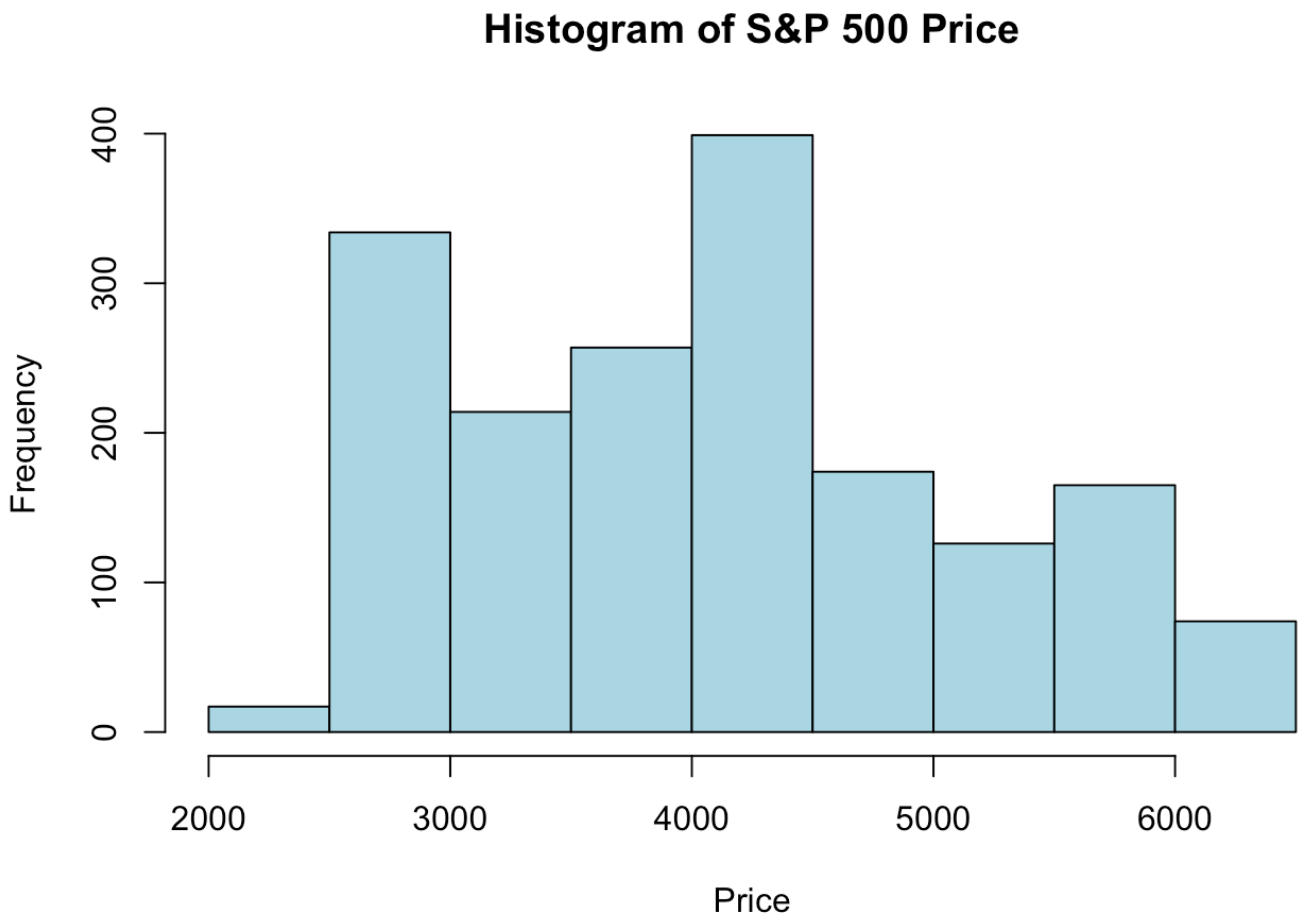
```
cor(merged$Price_sp500, merged$Price_btc, use = "complete.obs")
```

```
## [1] 0.0003677899
```

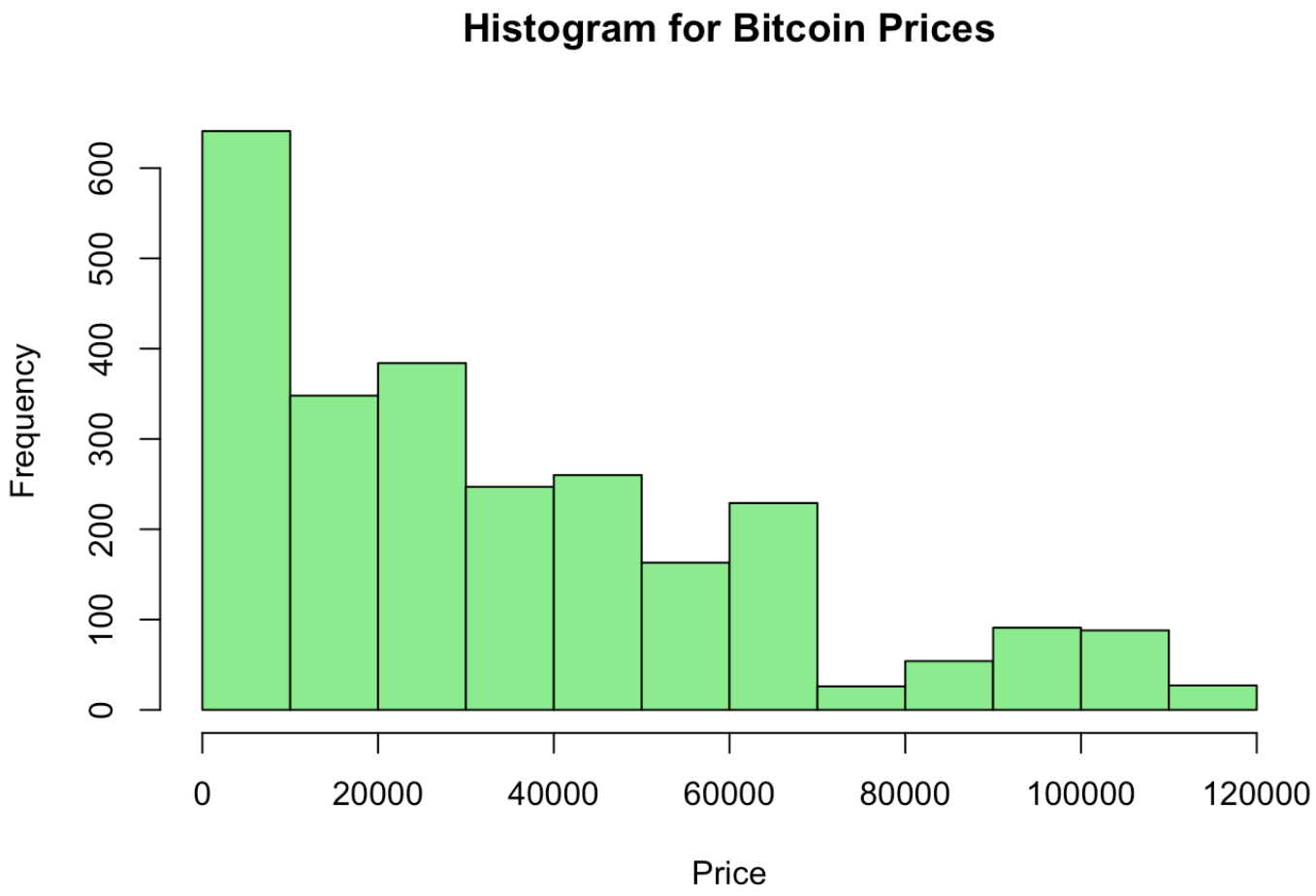
```
ggplot(merged, aes(x=Price_sp500, y=Price_btc))+
  geom_point(alpha=0.5, color="purple")+
  labs(title = "Scatter Plot: S&P 500 vs Bitcoin Prices",
       x="S&P 500 Price", y="Bitcoin Price")
```

#Performing Task-4

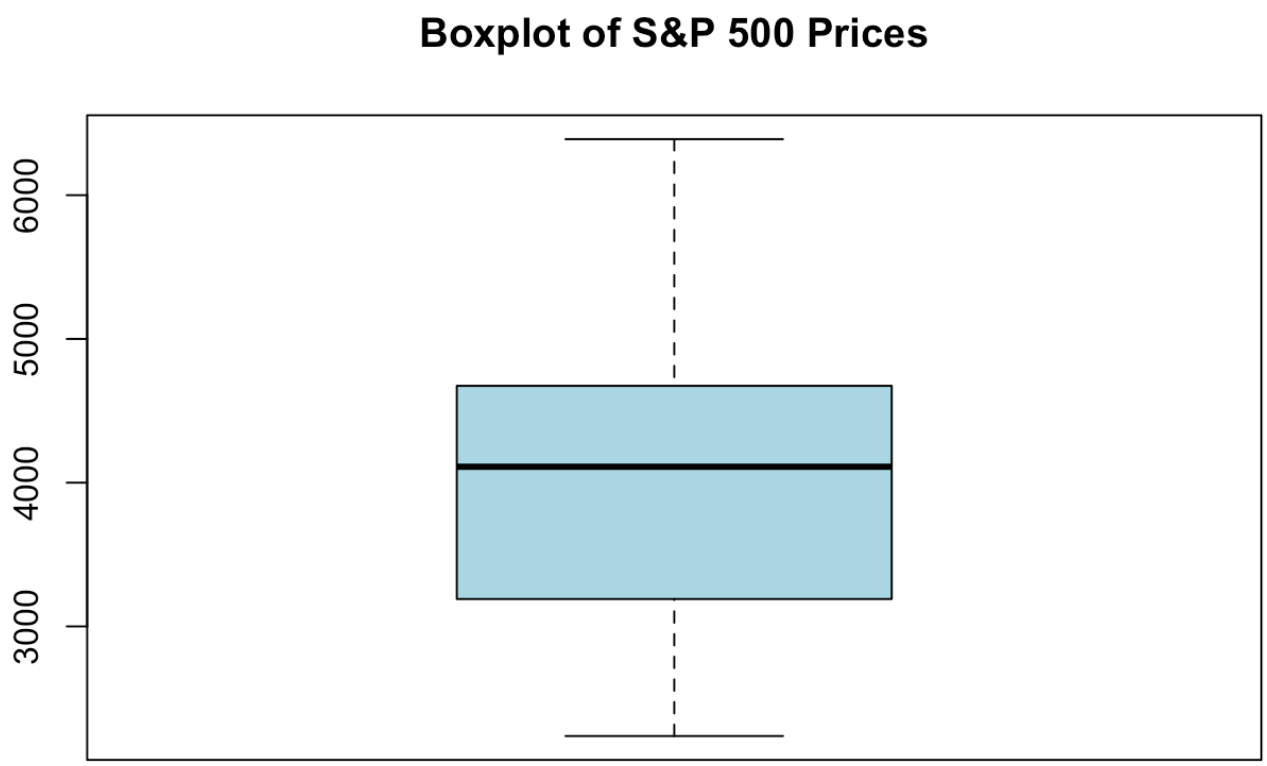
```
#histogram for both
hist(sp500$Price, main = "Histogram of S&P 500 Price", xlab = "Price", col = "ligh
tblue")
```



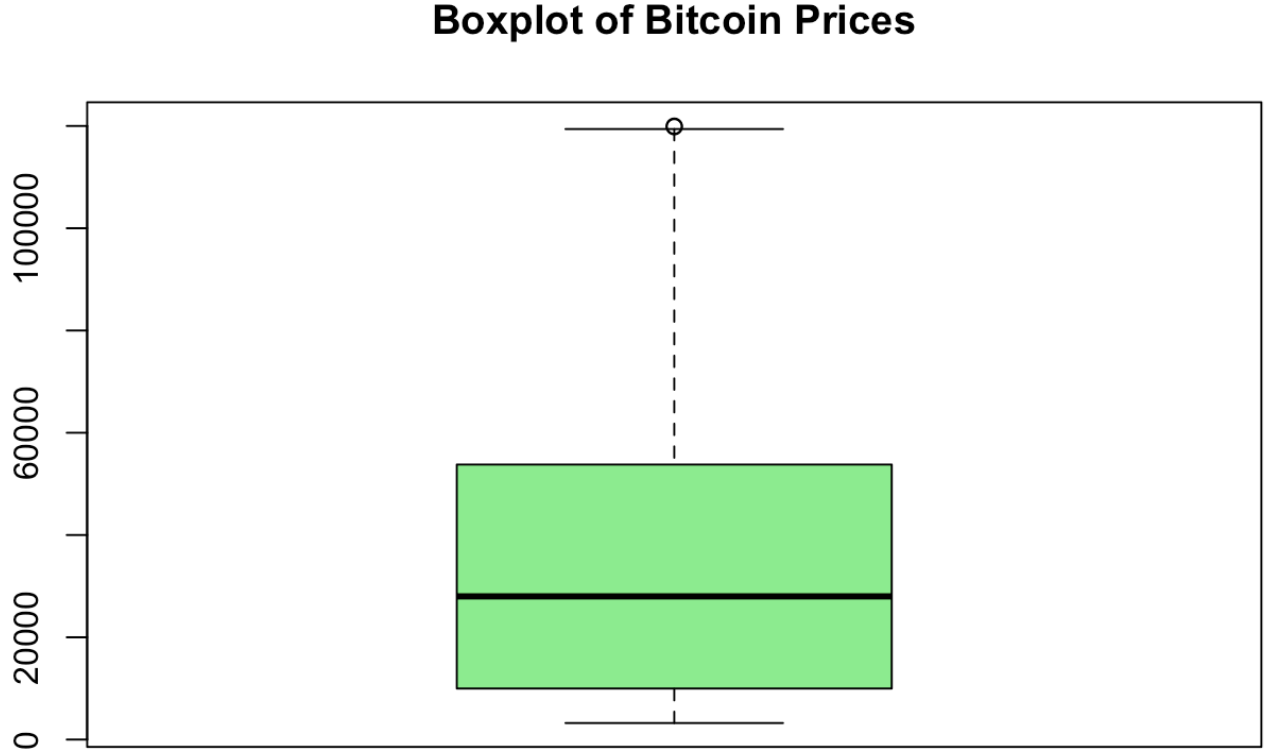

```
hist(bitcoin$Price, main= "Histogram for Bitcoin Prices", xlab = "Price", col = "l
ightgreen")
```




```
#boxplot for both
boxplot(sp500$Price, main="Boxplot of S&P 500 Prices", col = "lightblue")
```




```
boxplot(bitcoin$Price, main="Boxplot of Bitcoin Prices", col = "lightgreen")
```




```
#Shapiro-Wilk test
shapiro.test(sample(sp500$Price, 500))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(sp500$Price, 500)
## W = 0.95613, p-value = 4.888e-11
```

```
shapiro.test(sample(bitcoin$Price, 500))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample(bitcoin$Price, 500)
## W = 0.89034, p-value < 2.2e-16
```

#References:

Baglin, J. (2016) Statistics course materials: Modules 1–5. RMIT University.