DSC520: Final Project

Joanna Sierra-Mendoza

2024-05-06

Analyzing and Comparing Three Sectors of the Stock Market: A Data-Driven Approach to Identifying the Best Investment Opportunity

In today's financial markets, making informed investment decisions is important for investors to navigate the complexities of the economy. This study focuses on analyzing and comparing three sectors of the stock market: technology, consumer discretionary, and energy. The sectors will be represented by their most popular respective ETF: Technology Select Sector SPDR Fund (QQQ), iShares U.S. Consumer Discretionary ETF (IYC), and Energy Select Sector SPDR Fund (XLE).The objective is to find statistical methods that can be used as tools to help investors gain insights into understanding the dynamics within each sector, thereby facilitating investment decisions based on stock performance and potential. The investment choice out of the three sectors will be found by using a combination of quantitative analysis, qualitative insights, and risk assessment. In addition to analyzing sector-specific metrics, this study will consider broader economic indicators such as inflation and interest rates. These predictors play an important role in shaping market sentiment, influencing investor behavior, and ultimately impacting sector performance.

The following questions aim to research various aspects of sector analysis, including historical performance, correlation with market indices, returns, volatility assessment, and the impact of macroeconomic factors. The data found can benefit investors in making well-informed decisions that align with their investment plans and risk tolerance. 1. How do past returns of technology, consumer discretionary, and energy sectors compare? 2. What is the level of correlation between the returns of each sector's ETF and the S&P 500, which is the leading stock index that reflects market performance across all stock market sectors? 3. Are there statistically significant differences in the mean returns of each sector ETF, and if so, what factors contribute to these differences? 4. How does the volatility of returns for each sector ETF vary, and what implications does this have for risk management and investment decision-making? 5.To what extent do macroeconomic indicators such as inflation and interest rates influence the performance of the three sectors, and can these indicators be used to predict future sector trends?

A structured approach will help organize the data collected for each sector, making it easier to analyze and compare each sector to see which one is the best performing sector out of the three. The research will be started by collecting historical data from each sector as well as the macroeconomic sectors selected for this research. The data will then be cleaned to ensure accuracy and consistency. This may involve handling missing values, selecting a specific timeline, and aligning datasets. Descriptive analysis will then be used to gain insights on the behavior of each sector by looking at historical performance, volatility, and correlation patterns. Statistical analysis will then be applied to determine if there is a significant difference in performance and to help find out the factors that may be contributing to these differences. If predictors are found, then regression analysis will be conducted to explore the relationship between sector returns and macroeconomic indicators. This might also be helpful in predicting future sector trends. The final step will then be to interpret the findings and identify the sector that offers the most promising investment opportunity.

The ETF datasets were collected from Yahoo Finance - Stock Market Live, Quotes, Business & Finance News (https://finance.yahoo.com/). Some of the variables were removed such as market open, highest, and lowest price since only the closing price is of interest for this study. The macroeconomic data was obtained from GDP and Components - IMF Data (https://data.imf.org/regular.aspx?key=61545852). For the GDP file, a new csv file was created due to a lot of insignificant data included in the original file. The only data of interest is gross domestic product price per year in the U.S. Inflation data was retrieved from Historical Inflation Rates: 1914-2024 (https://www.usinflationcalculator.com/inflation/historical-inflation-rates/) , where data on the site was copied onto a csv file. S&P 500 data was retrieved from S&P 500 Historical Rates (SPX) (https://www.investing.com/indices/us-spx-500-historical-data) and the same variables as the other stocks were omitted. Inflation data was obtained from Interest Rates Data CSV Archive | U.S. Department of the Treasury (https://home.treasury.gov/interest-rates-data-csv-archive) and the 10 year treasury rate was removed. A variety of packages will be used to analyze and clean the data such as the tidyverse package. Dplyr will help manipulate the data and ggplot2 will be used to create visualizations. Statistical testing that will be required will need packages such as stats, PerformanceAnalytics, lmtest, and Metrics. Depending on the specific analysis tasks, additional packages may be found to be useful.

```r
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
```

```r
#ETFs
data_q <- read.csv("C:/Users/joann/OneDrive/Documents/rStudio projects/QQQ.csv")
data_QQQ <- subset(data_q, select = -c(Open, High, Low, Adj.Close))
head(data_QQQ)
```

```
##        Date Close    Volume
## 1  1/4/2010 46.42  62822800
## 2  1/5/2010 46.42  62935600
## 3  1/6/2010 46.14  96033000
## 4  1/7/2010 46.17  77094100
## 5  1/8/2010 46.55  88886600
## 6 1/11/2010 46.36 104673400
```

```r
data_i <- read.csv("C:/Users/joann/OneDrive/Documents/rStudio projects/IYC.csv")
data_IYC <- subset(data_i, select = -c(Open, High, Low, Adj.Close))
head(data_IYC)
```

```
##        Date   Close Volume
## 1  1/4/2010 13.9800 196400
## 2  1/5/2010 14.0050 102400
## 3  1/6/2010 13.9825 146400
## 4  1/7/2010 14.0825 138000
## 5  1/8/2010 14.0775 104000
## 6 1/11/2010 14.0750 142400
```

```
data_x <- read.csv("C:/Users/joann/OneDrive/Documents/rStudio projects/XLE.csv")
data_XLE <- subset(data_x, select = -c(Open, High, Low, Adj.Close))
head(data_XLE)
```

```
##        Date Close   Volume
## 1  1/4/2010 58.81 16928400
## 2  1/5/2010 59.29 17368100
## 3  1/6/2010 60.00 24351900
## 4  1/7/2010 59.91 17449500
## 5  1/8/2010 60.30 13344300
## 6 1/11/2010 60.22 19459900
```

```
#INTEREST RATE
data_t <- read.csv("C:/Users/joann/OneDrive/Documents/rStudio projects/intr.csv")
data_IT <- subset(data_t, select = -c(LT.COMPOSITE...10.Yrs.))
head(data_IT)
```

```
##         Date TREASURY.20.Yr.CMT
## 1 12/29/2023               4.20
## 2 12/28/2023               4.14
## 3 12/27/2023               4.10
## 4 12/26/2023               4.20
## 5 12/22/2023               4.21
## 6 12/21/2023               4.19
```

```
#GDP
data_GDP <- read.csv("C:/Users/joann/OneDrive/Documents/rStudio projects/GDP.csv")
head(data_GDP)
```

```
##   Year    X2010    X2011    X2012    X2013    X2014    X2015    X2016    X2017
## 1  GDP 15048971 15599732 16253970 16880683 17608138 18295019 18804913 19612103
##      X2018    X2019    X2020    X2021    X2022    X2023
## 1 20656516 21521395 21322950 23594031 25744108 27356393
```

```
#INFLATION
data_IF <- read.csv("C:/Users/joann/OneDrive/Documents/rStudio projects/infl.csv")
head(data_IF)
```

```
##   Year  Jan Feb  Mar  Apr May Jun Jul Aug Sep Oct Nov Dec Ave
## 1 2010  2.6 2.1  2.3  2.2 2.0 1.1 1.2 1.1 1.1 1.2 1.1 1.5 1.6
## 2 2011  1.6 2.1  2.7  3.2 3.6 3.6 3.6 3.8 3.9 3.5 3.4 3.0 3.2
## 3 2012  2.9 2.9  2.7  2.3 1.7 1.7 1.4 1.7 2.0 2.2 1.8 1.7 2.1
## 4 2013  1.6 2.0  1.5  1.1 1.4 1.8 2.0 1.5 1.2 1.0 1.2 1.5 1.5
## 5 2014  1.6 1.1  1.5  2.0 2.1 2.1 2.0 1.7 1.7 1.7 1.3 0.8 1.6
## 6 2015 -0.1 0.0 -0.1 -0.2 0.0 0.1 0.2 0.2 0.0 0.2 0.5 0.7 0.1
```

```
#S&P500
data_SP <- read.csv("C:/Users/joann/OneDrive/Documents/rStudio projects/SP.csv")
head(data_SP)
```

```
##         Date    Price
## 1 4/30/2024 5,035.69
## 2 4/29/2024 5,116.17
## 3 4/26/2024 5,099.96
## 4 4/25/2024 5,048.42
## 5 4/24/2024 5,071.63
## 6 4/23/2024 5,070.55
```

```
#RETURN
data_QQQ$Date <- as.Date(data_QQQ$Date, format = "%m/%d/%Y")
data_IYC$Date <- as.Date(data_IYC$Date, format = "%m/%d/%Y")
data_XLE$Date <- as.Date(data_XLE$Date, format = "%m/%d/%Y")


initial_price <- data_QQQ$Close[1]
final_price <- data_QQQ$Close[nrow(data_QQQ)]
percent_return <- ((final_price - initial_price) / initial_price) * 100
cat("Percent return from 01/04/2010 to 04/30/2024 is:", percent_return, "%")
```

```
## Percent return from 01/04/2010 to 04/30/2024 is: 832.2491 %
```

```
initial_price1 <- data_IYC$Close[1]
final_price1 <- data_IYC$Close[nrow(data_IYC)]
percent_return1 <- ((final_price1 - initial_price1) / initial_price1) * 100
cat("Percent return from 01/04/2010 to 04/30/2024 is:", percent_return1, "%")
```

```
## Percent return from 01/04/2010 to 04/30/2024 is: 467.5966 %
```

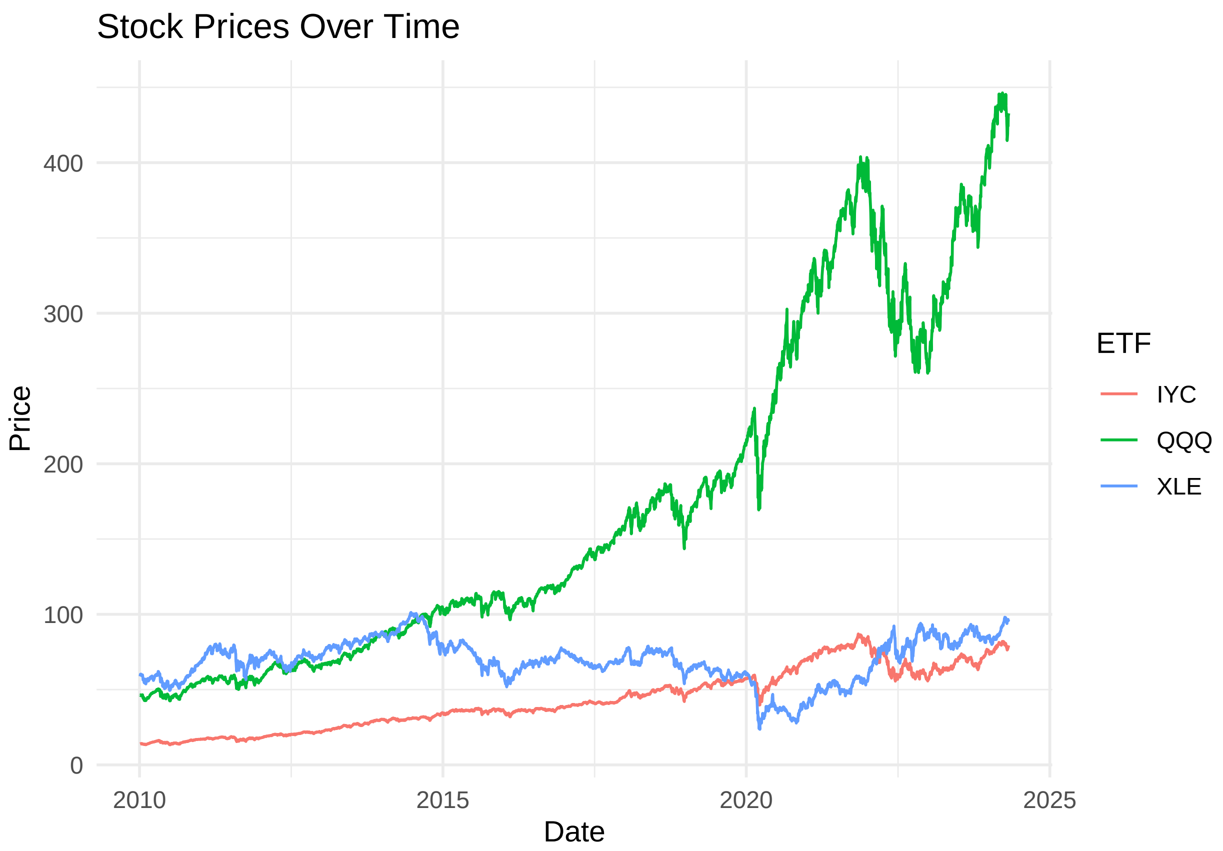
```
initial_price2 <- data_XLE$Close[1]
final_price2 <- data_XLE$Close[nrow(data_XLE)]
percent_return2 <- ((final_price2 - initial_price2) / initial_price2) * 100
cat("Percent return from 01/04/2010 to 04/30/2024 is:", percent_return2, "%")
```

```
## Percent return from 01/04/2010 to 04/30/2024 is: 63.90069 %
```

```
data_SP$Price <- as.numeric(gsub(",", "", data_SP$Price))
initial_price3 <- data_SP$Price[1]
final_price3 <- data_SP$Price[length(data_SP$Price)]
percent_return3 <- ((initial_price3 - final_price3) / final_price3) * 100
cat("Percent return from 01/04/2010 to 04/30/2024 is:", percent_return3, "%")
```

```
## Percent return from 01/04/2010 to 04/30/2024 is: 344.4563 %
```

```
#PRICE VISUALIZATION
library(ggplot2)
data_QQQ$ETF <- "QQQ"
data_IYC$ETF <- "IYC"
data_XLE$ETF <- "XLE"

compare_etfs <- rbind(data_QQQ, data_IYC, data_XLE)
ggplot(compare_etfs, aes(x = Date, y = Close, color = ETF)) +
geom_line() + labs(title = "Stock Prices Over Time",
x = "Date", y = "Price", color = "ETF") + theme_minimal()
```



```r
#Interest Rate + QQQ
data_IT$Date <- as.Date(data_IT$Date, format = "%m/%d/%Y")
merged_data <- merge(data_QQQ, data_IT, by = "Date")
head(merged_data)
```

```
##         Date Close    Volume ETF TREASURY.20.Yr.CMT
## 1 2010-01-04 46.42  62822800 QQQ               4.60
## 2 2010-01-05 46.42  62935600 QQQ               4.54
## 3 2010-01-06 46.14  96033000 QQQ               4.63
## 4 2010-01-07 46.17  77094100 QQQ               4.62
## 5 2010-01-08 46.55  88886600 QQQ               4.61
## 6 2010-01-11 46.36 104673400 QQQ               4.64
```

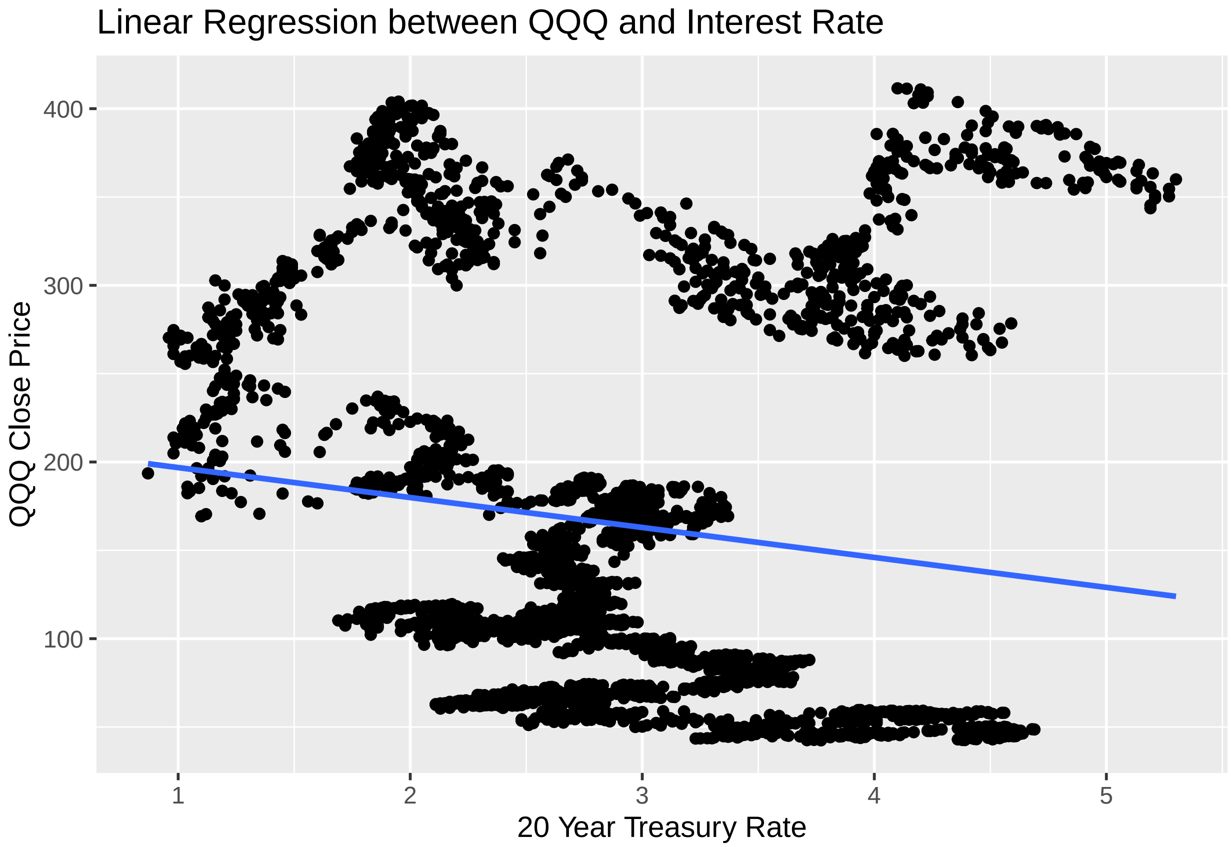
```r
lm_QQQ <- lm(Close ~ TREASURY.20.Yr.CMT, data = merged_data)
summary(lm_QQQ)
```

```
##
## Call:
## lm(formula = Close ~ TREASURY.20.Yr.CMT, data = merged_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -120.11  -82.11  -45.29   56.44  268.29
##
## Coefficients:
```

```
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           213.797      6.247  34.222  < 2e-16 ***
## TREASURY.20.Yr.CMT    -16.953      2.091  -8.106 7.17e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105.1 on 3494 degrees of freedom
## Multiple R-squared:  0.01846,    Adjusted R-squared:  0.01818
## F-statistic: 65.71 on 1 and 3494 DF,  p-value: 7.171e-16
```

```
ggplot(merged_data, aes(x = TREASURY.20.Yr.CMT, y = Close)) +
geom_point() +  geom_smooth(method = "lm", se = FALSE) +
labs(title = "Linear Regression between QQQ and Interest Rate",
x = "20 Year Treasury Rate", y = "QQQ Close Price")
```

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



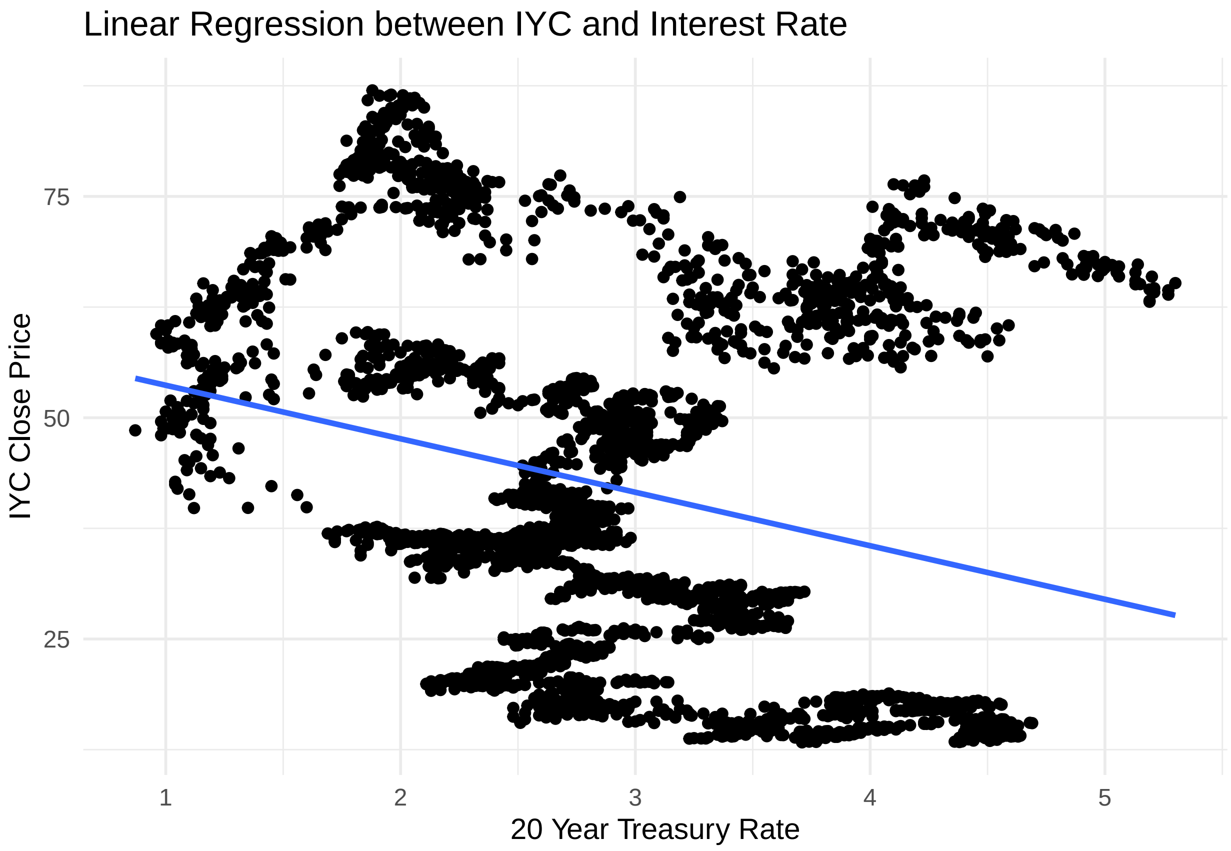
```
#Interest Rate + IYC
data_IT$Date <- as.Date(data_IT$Date, format = "%m/%d/%Y")
merged_data1 <- merge(data_IYC, data_IT, by = "Date")
lm_IYC <- lm(Close ~ TREASURY.20.Yr.CMT, data = merged_data1)
summary(lm_IYC)
```

```
##
```

```
## Call:
## lm(formula = Close ~ TREASURY.20.Yr.CMT, data = merged_data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.010 -14.584  -5.181  10.746  42.654
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)          59.7089     1.1162   53.49   <2e-16 ***
## TREASURY.20.Yr.CMT  -6.0433     0.3737  -16.17   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 18.78 on 3494 degrees of freedom
## Multiple R-squared:  0.06964,    Adjusted R-squared:  0.06938
## F-statistic: 261.6 on 1 and 3494 DF,  p-value: < 2.2e-16
```

```r
ggplot(merged_data1, aes(x = TREASURY.20.Yr.CMT, y = Close)) +
geom_point() +  geom_smooth(method = "lm", se = FALSE) +
labs(title = "Linear Regression between IYC and Interest Rate",
x = "20 Year Treasury Rate", y = "IYC Close Price") + theme_minimal()
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```

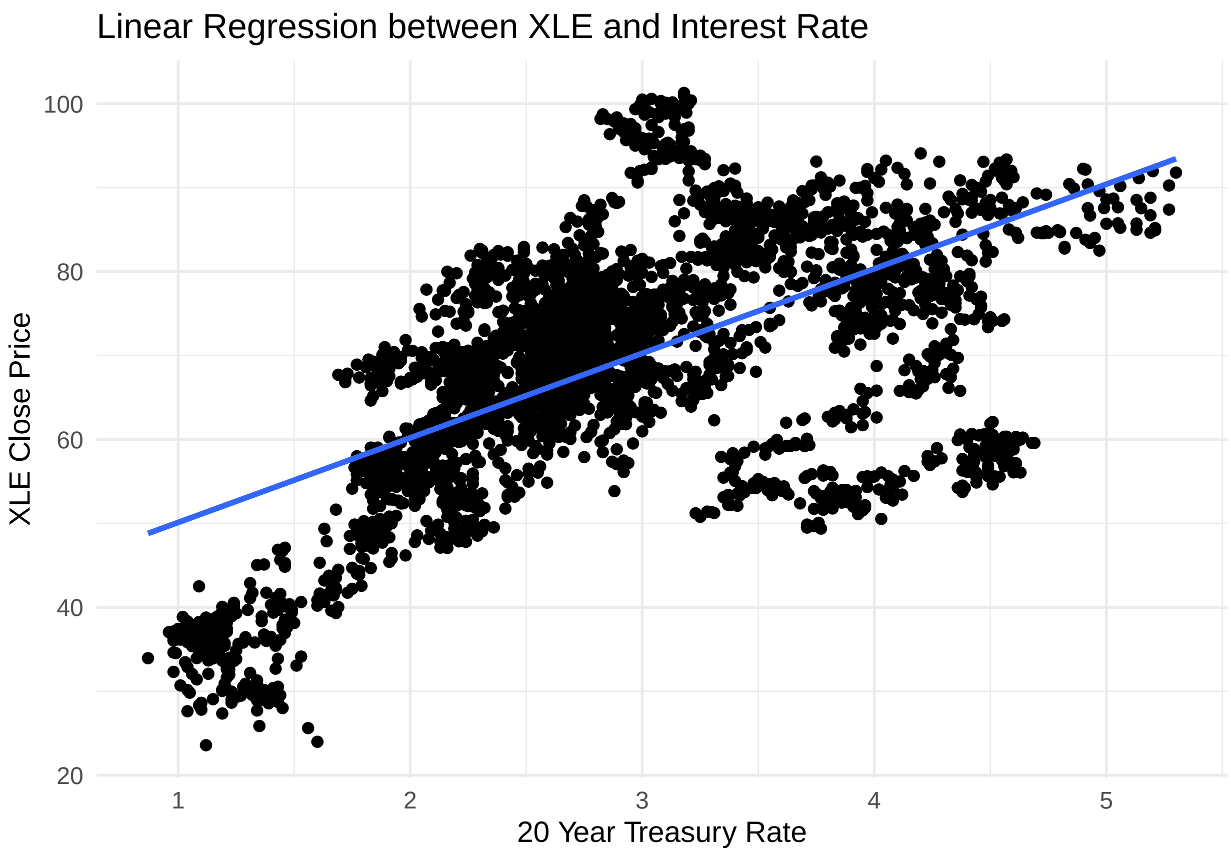


```r
#Interest Rate + XLE
data_IT$Date <- as.Date(data_IT$Date, format = "%m/%d/%Y")
merged_data2 <- merge(data_XLE, data_IT, by = "Date")
lm_XLE <- lm(Close ~ TREASURY.20.Yr.CMT, data = merged_data2)
summary(lm_XLE)
```

```
##
## Call:
## lm(formula = Close ~ TREASURY.20.Yr.CMT, data = merged_data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.182  -5.572   0.908   7.319  30.223
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         40.0419     0.6747   59.35   <2e-16 ***
## TREASURY.20.Yr.CMT  10.0750     0.2259   44.61   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 11.35 on 3494 degrees of freedom
## Multiple R-squared:  0.3628, Adjusted R-squared:  0.3627
## F-statistic:  1990 on 1 and 3494 DF,  p-value: < 2.2e-16
```

```r
ggplot(merged_data2, aes(x = TREASURY.20.Yr.CMT, y = Close)) +
geom_point() + geom_smooth(method = "lm", se = FALSE) +
labs(title = "Linear Regression between XLE and Interest Rate",
x = "20 Year Treasury Rate",
y = "XLE Close Price") +
theme_minimal()
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’
```



```
#Inflation + QQQ
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

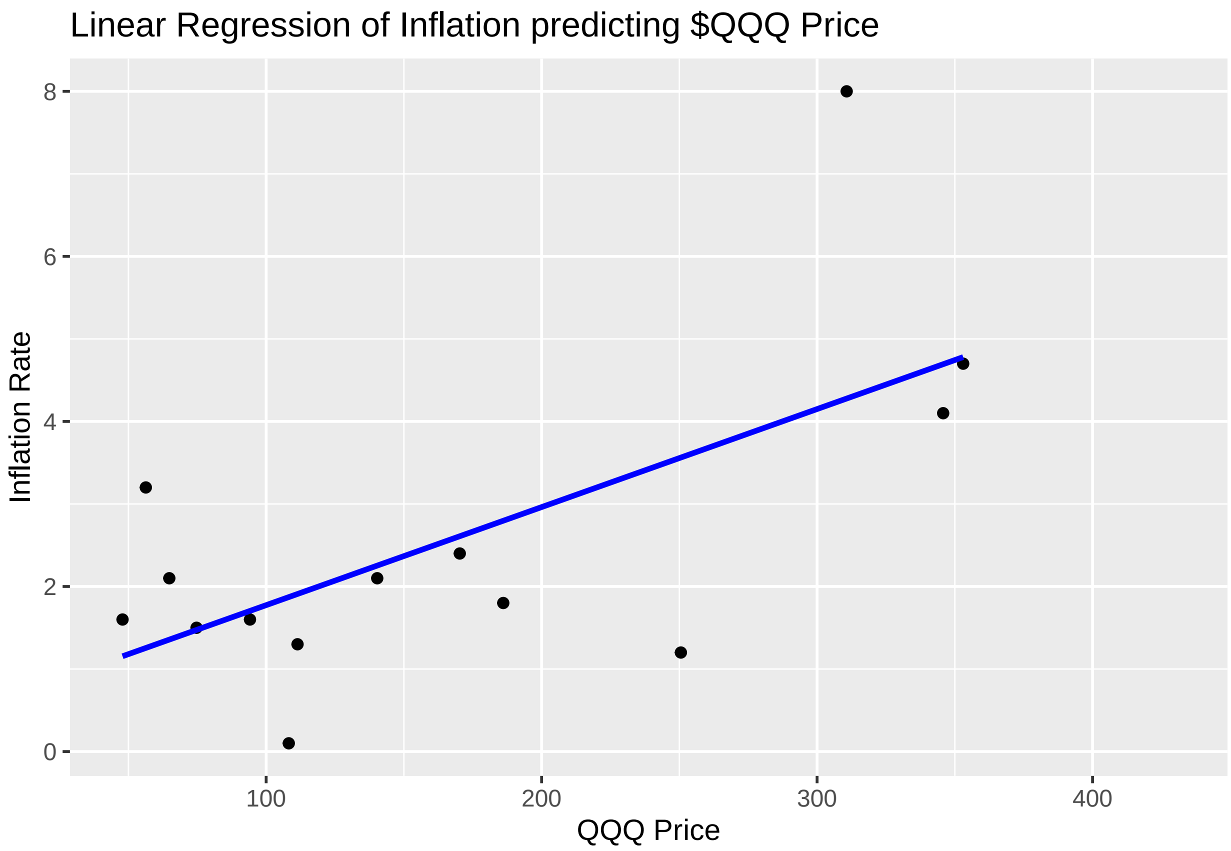
```
data_QQQ$Year <- year(data_QQQ$Date)
annual_data_QQQ <- data_QQQ %>% group_by(Year) %>%
summarise(Ave_Close = mean(Close))
merged_data4 <- merge(data_IF, annual_data_QQQ, by = "Year")
model_qqq <- lm(Ave ~ Ave_Close, data = merged_data4)
summary(model_qqq)
```

```
##
## Call:
## lm(formula = Ave ~ Ave_Close, data = merged_data4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3627 -0.6056 -0.1289  0.3403  3.7224
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.586270   0.777061   0.754   0.4651
## Ave_Close   0.011879   0.003975   2.989   0.0113 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.552 on 12 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4267, Adjusted R-squared:  0.3789
## F-statistic: 8.932 on 1 and 12 DF,  p-value: 0.0113
```

```r
ggplot(merged_data4, aes(x = Ave_Close, y = Ave)) +
geom_point() + geom_smooth(method = "lm", se = FALSE, color = "blue") +
labs(title = "Linear Regression of Inflation predicting $QQQ Price",
x = "QQQ Price", y = "Inflation Rate")
```

```
## ‘geom_smooth()‘ using formula = ’y ~ x’

## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_smooth()‘).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (‘geom_point()‘).
```



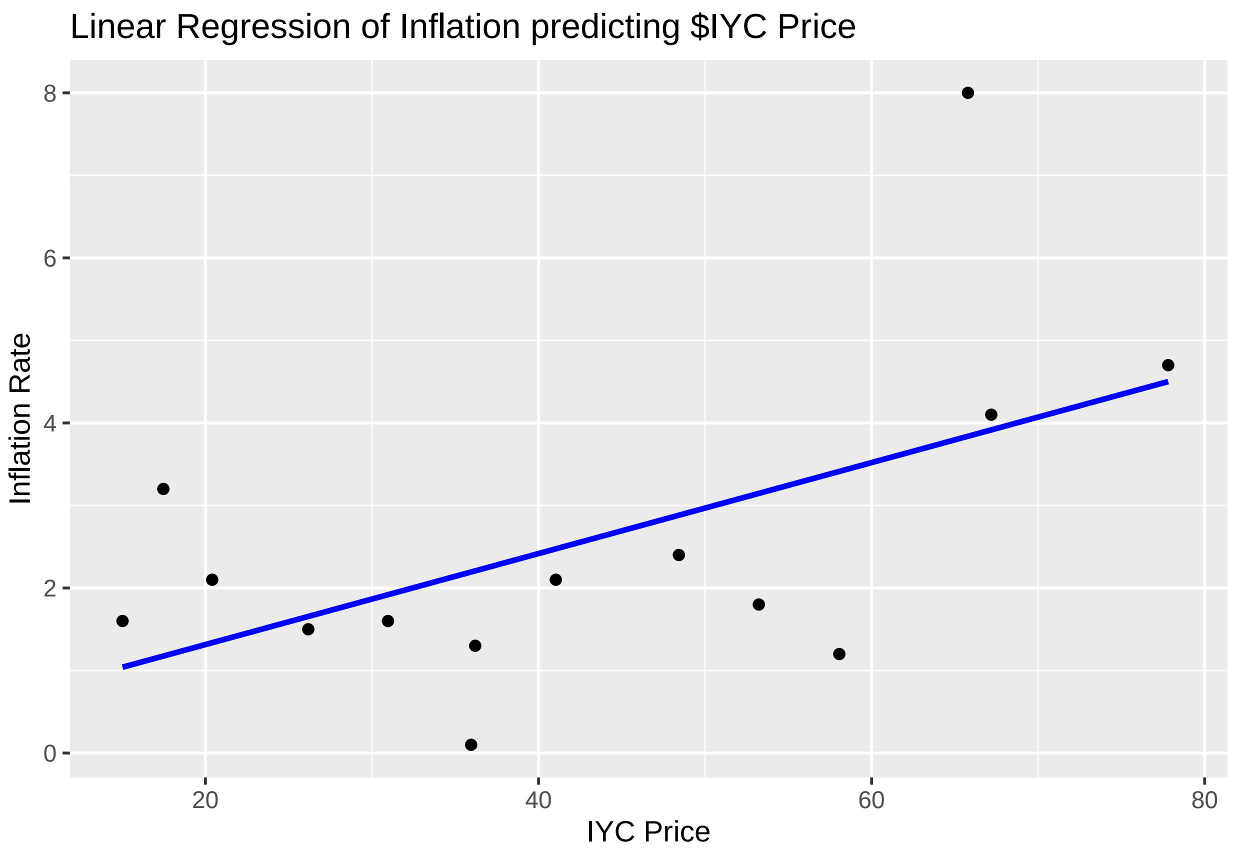
```r
#Inflation + IYC
data_IYC$Year <- year(data_IYC$Date)
annual_data_IYC <- data_IYC %>% group_by(Year) %>%
summarise(Ave_Close1 = mean(Close))
merged_data5 <- merge(data_IF, annual_data_IYC, by = "Year")
model_iyc <- lm(Ave ~ Ave_Close1, data = merged_data5)
summary(model_iyc)
```

```
##
## Call:
## lm(formula = Ave ~ Ave_Close1, data = merged_data5)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2129 -0.8011 -0.2370  0.4694  4.1614
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.21222    1.10501   0.192   0.8509
## Ave_Close1   0.05512    0.02374   2.322   0.0386 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.703 on 12 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:   0.31,  Adjusted R-squared:  0.2525
## F-statistic:  5.39 on 1 and 12 DF,  p-value: 0.03865
```

```r
ggplot(merged_data5, aes(x = Ave_Close1, y = Ave)) +
geom_point() + geom_smooth(method = "lm", se = FALSE, color = "blue") +
labs(title = "Linear Regression of Inflation predicting $IYC Price",
x = "IYC Price", y = "Inflation Rate")
```

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

```
## Warning: Removed 1 row containing non-finite outside the scale range (`stat_smooth()`).
## Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```

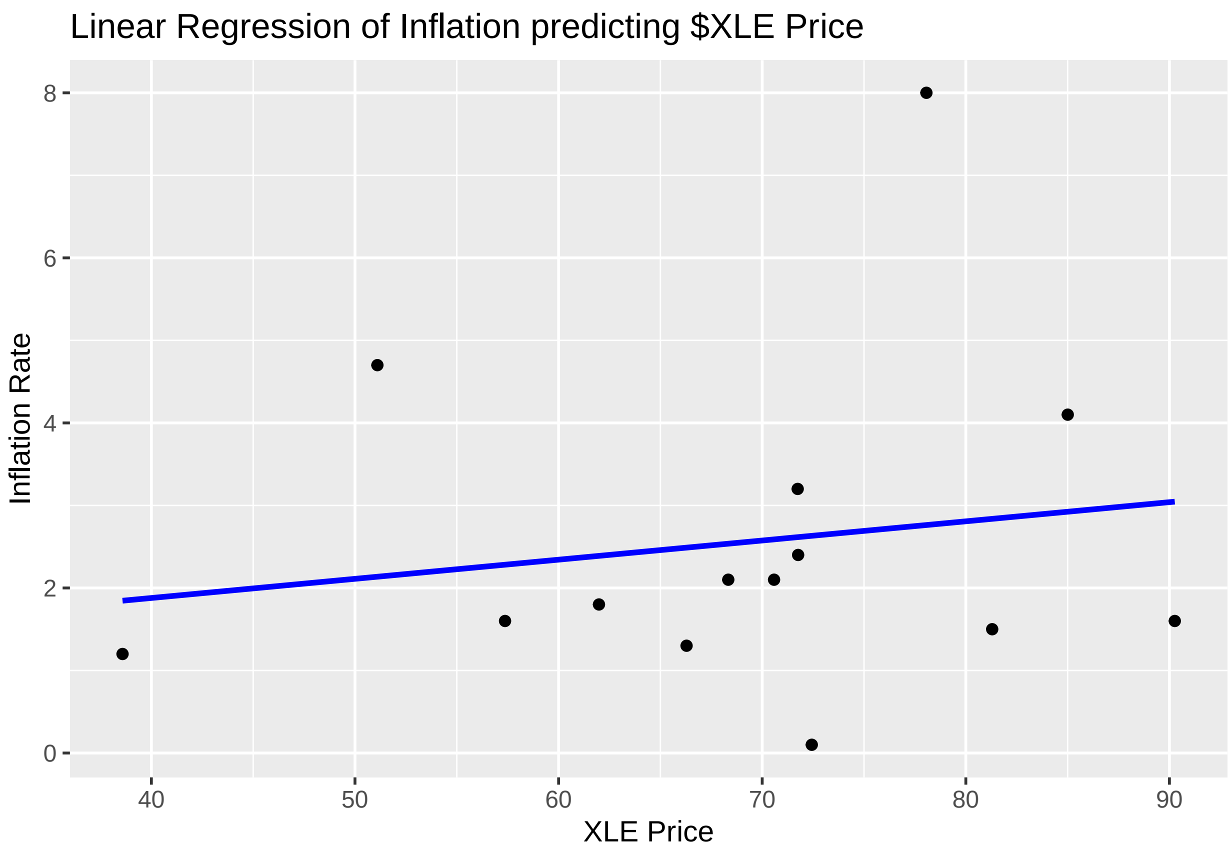


```
#Inflation + XLE
data_XLE$Year <- year(data_XLE$Date)
annual_data_XLE <- data_XLE %>% group_by(Year) %>%
summarise(Ave_Close2 = mean(Close))
merged_data6 <- merge(data_IF, annual_data_XLE, by = "Year")
model_xle <- lm(Ave ~ Ave_Close2, data = merged_data6)
summary(model_xle)
```

```
##
## Call:
## lm(formula = Ave ~ Ave_Close2, data = merged_data6)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.5317 -1.0622 -0.5389  0.3842  5.2376
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.95003    2.89230   0.328    0.748
## Ave_Close2   0.02322    0.04123   0.563    0.584
##
## Residual standard error: 2.024 on 12 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.02574,    Adjusted R-squared:  -0.05544
## F-statistic: 0.3171 on 1 and 12 DF,  p-value: 0.5837
```

```
ggplot(merged_data6, aes(x = Ave_Close2, y = Ave)) +
geom_point() + geom_smooth(method = "lm", se = FALSE, color = "blue") +
labs(title = "Linear Regression of Inflation predicting $XLE Price",
x = "XLE Price", y = "Inflation Rate")
```

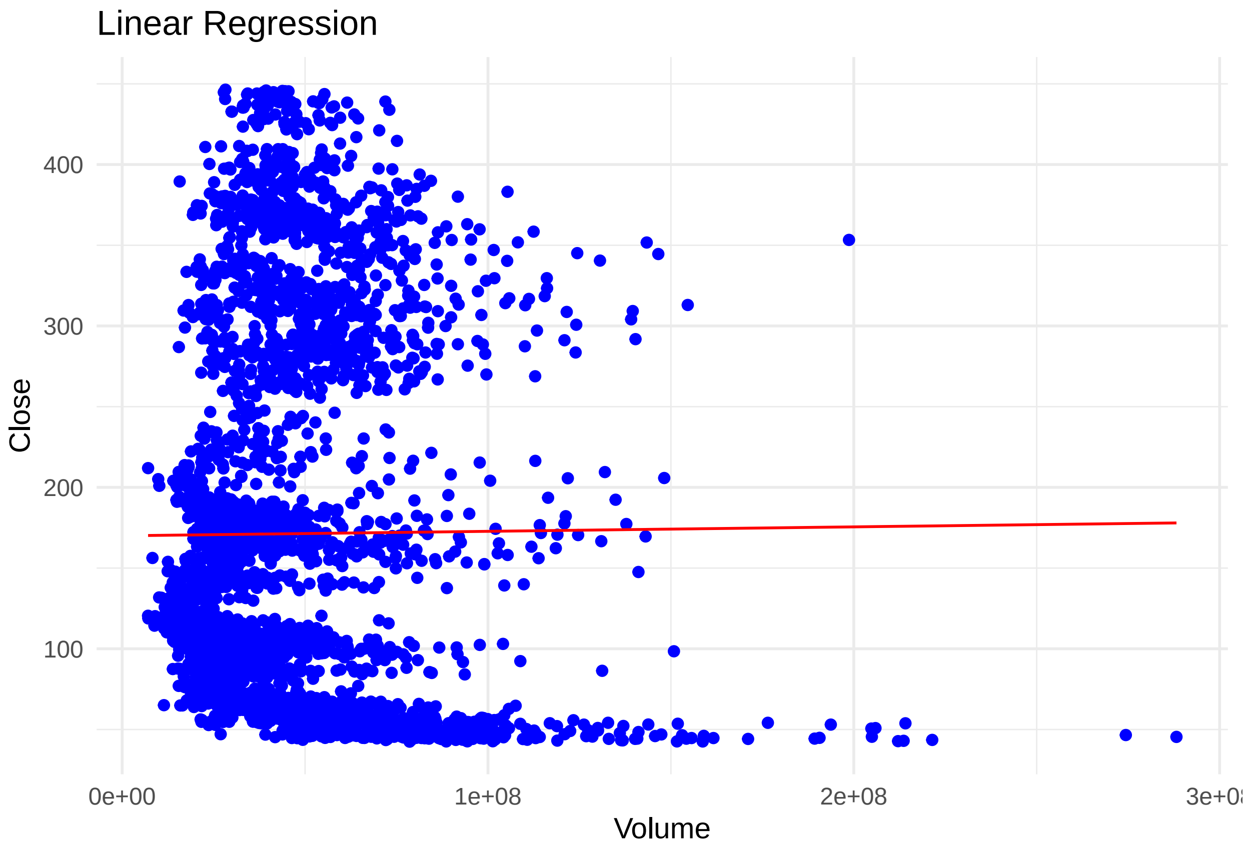
```
## ‘geom_smooth()‘ using formula = ’y ~ x’

## Warning: Removed 1 row containing non-finite outside the scale range (‘stat_smooth()‘).
## Removed 1 row containing missing values or values outside the scale range
## (‘geom_point()‘).
```



```
#USING VOLUME AS A PREDICTOR
#QQQ
X <- data_QQQ$Volume
y <- data_QQQ$Close
model_qvol <- lm(y ~ X)
summary(model_qvol)
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -133.07  -95.19  -40.09   99.16  275.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.700e+02  3.729e+00  45.588   <2e-16 ***
## X           2.762e-08  7.047e-08   0.392    0.695
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 112.1 on 3602 degrees of freedom
## Multiple R-squared:  4.263e-05,  Adjusted R-squared:  -0.000235
## F-statistic: 0.1536 on 1 and 3602 DF,  p-value: 0.6952
```

```
plot_qvol <- data.frame(Volume = X, Close = y, Predicted_q= predict(model_qvol))
ggplot(plot_qvol, aes(x = Volume, y = Close)) +
geom_point(color = "blue") + geom_line(aes(y = Predicted_q), color = "red") +
labs(title = "Linear Regression", x = "Volume", y = "Close") + theme_minimal()
```

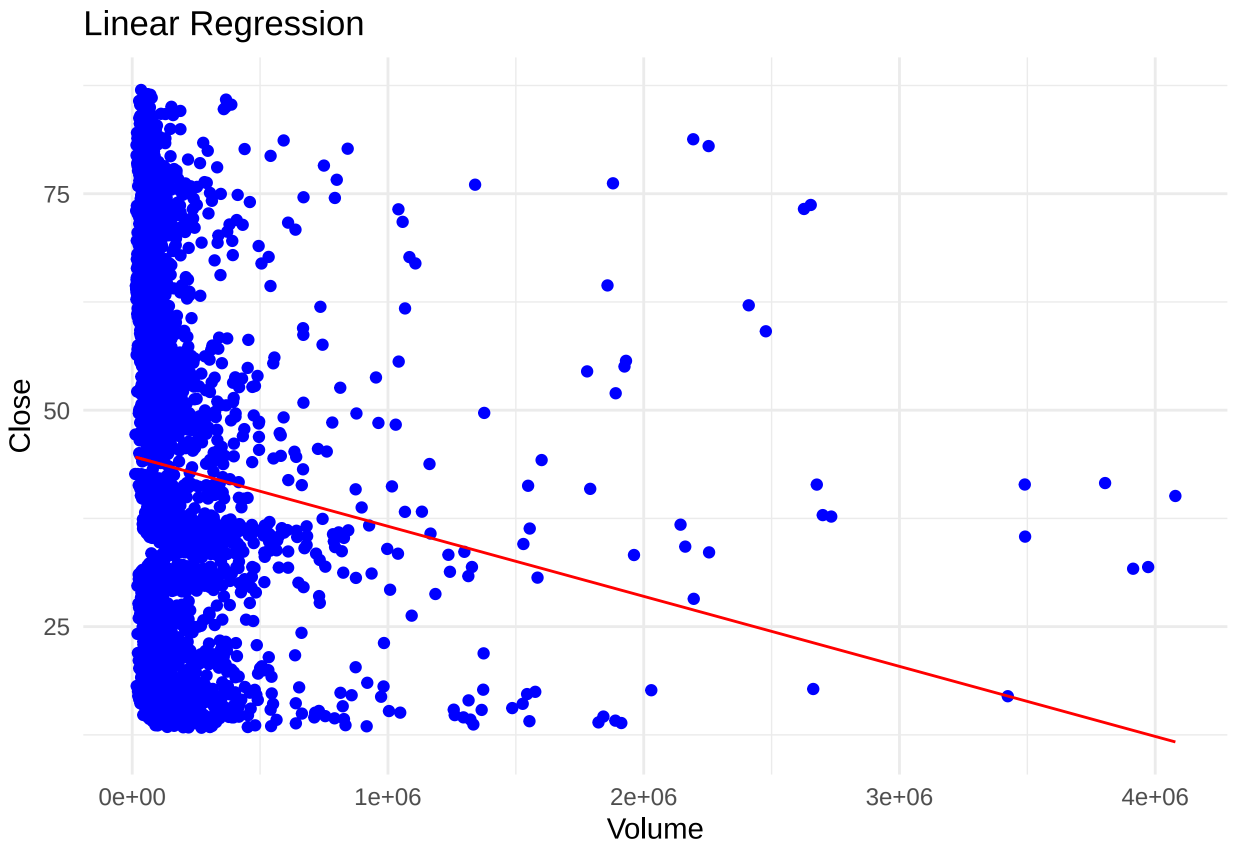


```
#IYC
X <- data_IYC$Volume
y <- data_IYC$Close
model_ivol <- lm(y ~ X)
summary(model_qvol)
```

```
##
```

```
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -133.07  -95.19  -40.09   99.16  275.60
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.700e+02  3.729e+00  45.588   <2e-16 ***
## X           2.762e-08  7.047e-08   0.392    0.695
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 112.1 on 3602 degrees of freedom
## Multiple R-squared:  4.263e-05,  Adjusted R-squared:  -0.000235
## F-statistic: 0.1536 on 1 and 3602 DF,  p-value: 0.6952
```

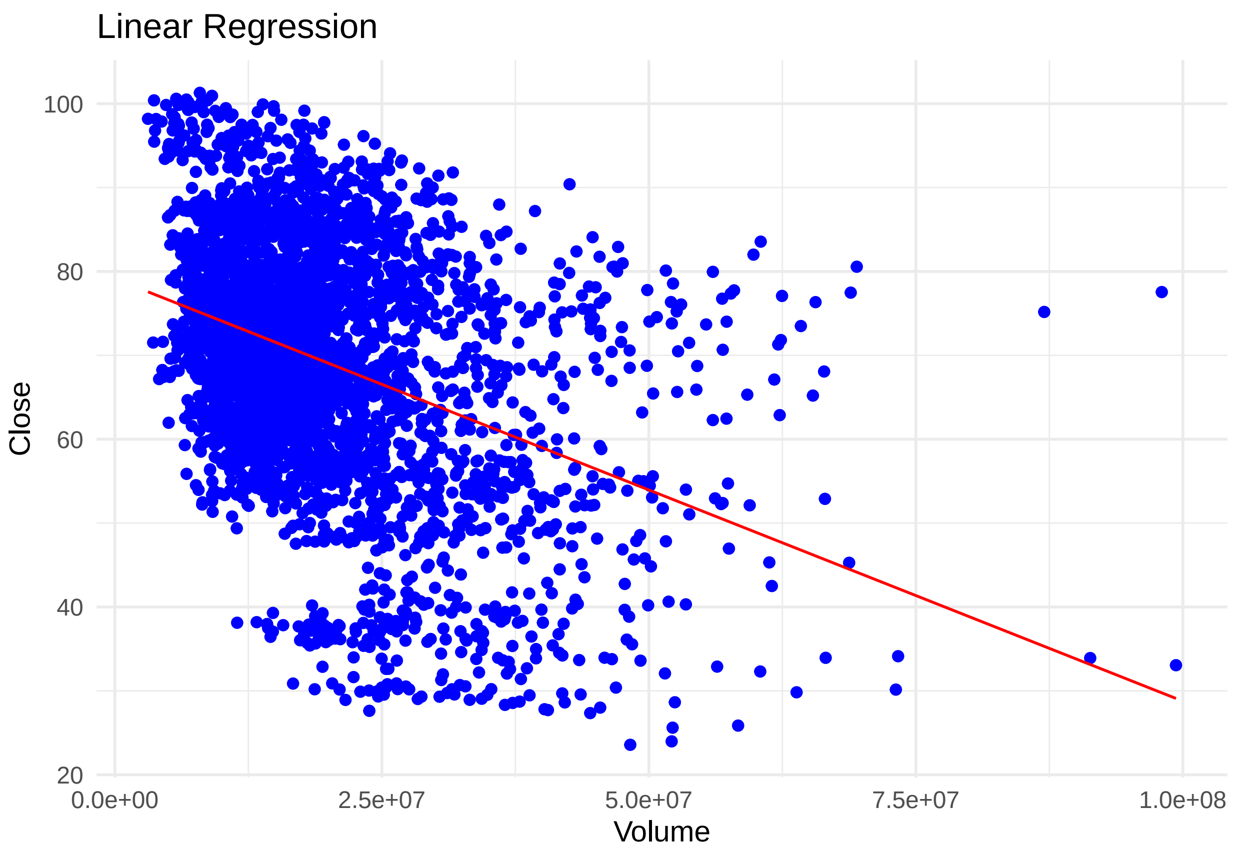
```
plot_ivol <- data.frame(Volume = X, Close = y, Predicted_i= predict(model_ivol))
ggplot(plot_ivol, aes(x = Volume, y = Close)) +
geom_point(color = "blue") + geom_line(aes(y = Predicted_i), color = "red") +
labs(title = "Linear Regression", x = "Volume", y = "Close") +
theme_minimal()
```



```r
#XLE
X <- data_XLE$Volume
y <- data_XLE$Close
model_xvol <- lm(y ~ X)
summary(model_xvol)
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.859  -8.109  -0.740   9.613  47.775
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.913e+01  4.581e-01  172.74   <2e-16 ***
## X           -5.036e-07  2.061e-08  -24.44   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 13.32 on 3602 degrees of freedom
## Multiple R-squared:  0.1422, Adjusted R-squared:  0.142
## F-statistic: 597.1 on 1 and 3602 DF,  p-value: < 2.2e-16
```

```r
plot_xvol <- data.frame(Volume = X, Close = y, Predicted_x= predict(model_xvol))
ggplot(plot_xvol, aes(x = Volume, y = Close)) +
geom_point(color = "blue") + geom_line(aes(y = Predicted_x), color = "red") +
labs(title = "Linear Regression", x = "Volume", y = "Close") +
theme_minimal()
```



To illustrate the findings of the research questions, a combination of plots and tables can effectively convey the results. For descriptive and regression analysis, line charts can show the historical performance of each sector ETF overtime. Scatter plots with regression lines can be used to visualize the relationship between sector returns and macroeconomic indicators. Residual plots would also be used to check the assumptions of regression models. Statistical testing with scatter plots with trend lines would help visualize the relationships between variables. ANOVA can be used to assess differences in mean returns between the sectors.

Percent return was calculated for each ETF and of the S&P 500 from 01/04/2010 to 04/30/2024. The S&P 500 is the index used see the performance of the overall stock market. The S&P had a return of 344% and out of the three ETFs, QQQ and IYC outperformed the overall stock market with a percent return of 832 and 468, respectively. The energy sector did not outperform the U.S. market and XLE only had a percent return of 64. Looking at the chart for price overtime does show that QQQ is volatile and might carry greater risk since stocks in this sector have a high PE ratio and may be overvalued. The P/E ratio is a metric used to compare a company's current share price to its earnings per share. If the sector encounters trouble, this can cause a correction in the price. Stocks with high P/E ratios carry greater risk since they can be more vulnerable to market sentiment, a deviation can lead to a sharp decline. The average P/E ratio for the overall stock market has been from 15-20, in other words, investors have been willing to pay 15-20 times the company's earnings per share for its stock. A stock included in the QQQ ETF that has been trending is NVIDIA (NVDA) and it has a price-to-earnings ratio of 64 with a share price at around $1,100. A year ago, this stock was trading at around $300, making it a highly volatile stock in the technology sector.

Stock price for each ETF was plotted to help visualize the price change throughout the selected time period. Based on the plotted ETFs, QQQ had the most growth out of all of the three with an approximate return of 832%. A linear regression model was created for each ETF to see if interest rate and inflation could be helpful in predicting stock price. When comparing both models, inflation rate had a higher percentage as a predictor in stock price.Volume did not seem to be a good predictor of stock price. The typical p-value level is 0.05 and the p-value for this model was far from this value at 0.695, failing to reject the null hypothesis that there is a significant relationship between volume and stock price.

The relationship between the 20 year interest rate and closing price was observed for each ETF. A linear regression analysis was performed for all ETFs and it was found that only 1.8% of the variability in closing price of QQQ was explained by interest rate. For IYC and XLE, 6.9% and 36.3%, respectively, of the variability in closing price was explained by interest rate. An explanation for why the interest rate affects the energy sector more could be due to the fact that companies in the energy sector typically carry a lot of debt and a higher interest rate means higher debt servicing costs. This can reduce cash flow and net income of the company. The same observation was done using the inflation rate for each sector. Results showed that 42.7% of the variability in closing price for QQQ can be explained by inflation rate. Closing prices for IYC and XLE can be explained by inflation for 31% and 31.7% of the variability, respectively. These results showed that macroeconomic indicators, especially inflation, do have an affect on stock performance. Trading volume and its effect on stock price was observed and it was found that there was no significant correlation between the two since the p-value was high for all sectors.

There are advanced regression tests that I could perform if needed. Some of the models that I created could have weak statistical power due to problems such as the sample size not being large enough to detect any significance. I have not been challenged on trying to fix a model if it is biased. So far, the analysis will be conducted on data collected from the last 14 years. This might not be sufficient data to answer the question of which sector would be the best option to invest in since there could be variables other than macroeconomic ones that haven't been looked into and could be highly significant in predicting a sector's performance. Looking further into the residuals and retrieving the rmse of each model can also help in checking how accurate our model is in predicting. Other factors, such as social media, have had strong effects on stock price movement recently. An example of this that happened this year, Keith Gill who has built a large online community, made a tweet for the first time in three years. This alone caused the stock price of GameStop (GME) to run up almost 100% in price immediately after his tweet.

To continue this project, machine learning models would be useful in making more accurate predictions on stock price movement. It can leverage high amounts complex data and identify patterns more efficiently. Through the use of algorithms, machine learning models can make analyses based on past stock market performance, market sentiment, economic indicators, and other relevant factors that are common today such as social media. These models can also uncover different types of relationships, such as linear relationships where stock prices change at a constant rate, non-linear relationships where changes are not proportional and complex, and even time-dependent relationships that focus on how past prices influence future prices over time. Common approaches involve supervised learning techniques, where models such as linear regression, neural networks, and decision trees are trained on labeled data sets to predict stock price. More advanced methods like ensemble learning, which combines multiple models to improve accuracy, and time series analysis, which focuses on capturing trends and recurring fluctuations, could help enhance accuracy. More accurate predictions would aid investors in making informed decisions that can help enhance their portfolio returns.