Recent S&P 500 ETF(SPY) Price Analysis

Reid Peterson (reidpeterson@ucsb.edu)

2025-02-18

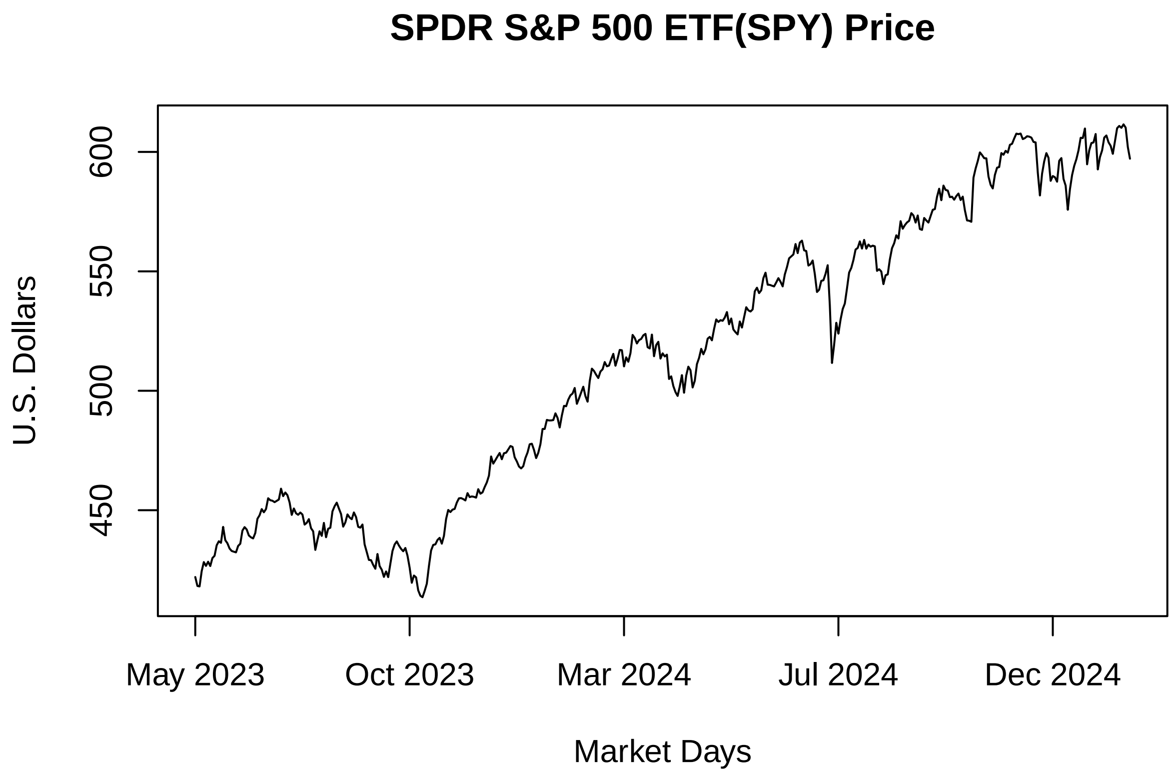
Abstract

Data collected over time can produce very useful projections and inferences using time series analysis techniques. This project serves to conduct time series analysis on the opening price of the SPY fund over the last two years. The analysis begins with basic exploration and informative testing, including tests such as the Augmented Dickey-Fuller(ADF), Auto-Correlation Function(ACF), and Partial ACF(PACF) tests. Throughout the process, many different models are fit, including variations of the ARIMA, SARIMA and GARCH. These models are then compared on the basis of simplicity, and AIC/BIC values. To conclude, the best model is used to produce a forecast for future data points.

1. Introduction

The U.S. stock market is an opportunity for investors and businesses alike to grow their capital and help contribute to positive change in the world. It's also very indicative of the condition of the economy, which affects everyone living here. Fortunately for the population, the data from the stock market is both readily available and very extensive, covering thousands of businesses dating all the way back to the 19th century. Throughout this project I will be taking a close look at the opening prices of the SPDR S&P 500 ETF(SPY) over the last two years. I am conducting this analysis to search for trends in the data, and to see if any inferences can be drawn. I decided to choose this fund in particular as it covers all the main industries, and is thus representative of the U.S. stock market in general.

Previous studies have found that SPY exhibits seasonal patterns, with certain months like March and October showing wider ranges of returns. Similar to this study we too examine potential seasonal cycles, but with additional focus on heteroskedasticity. In other words, after our initial testing and model fitting, we explore how dependent the volatility of the data is on time. We do this using the Generalized Autoregressive Conditional Heteroskedasticity(GARCH) model. Ultimately, we discover that this time series is ideally represented with an ARIMA(0,1,0), or just a first-differenced ARMA(0,0). Also, we find that while past shocks impact current volatility, past volatility does not persist significantly.



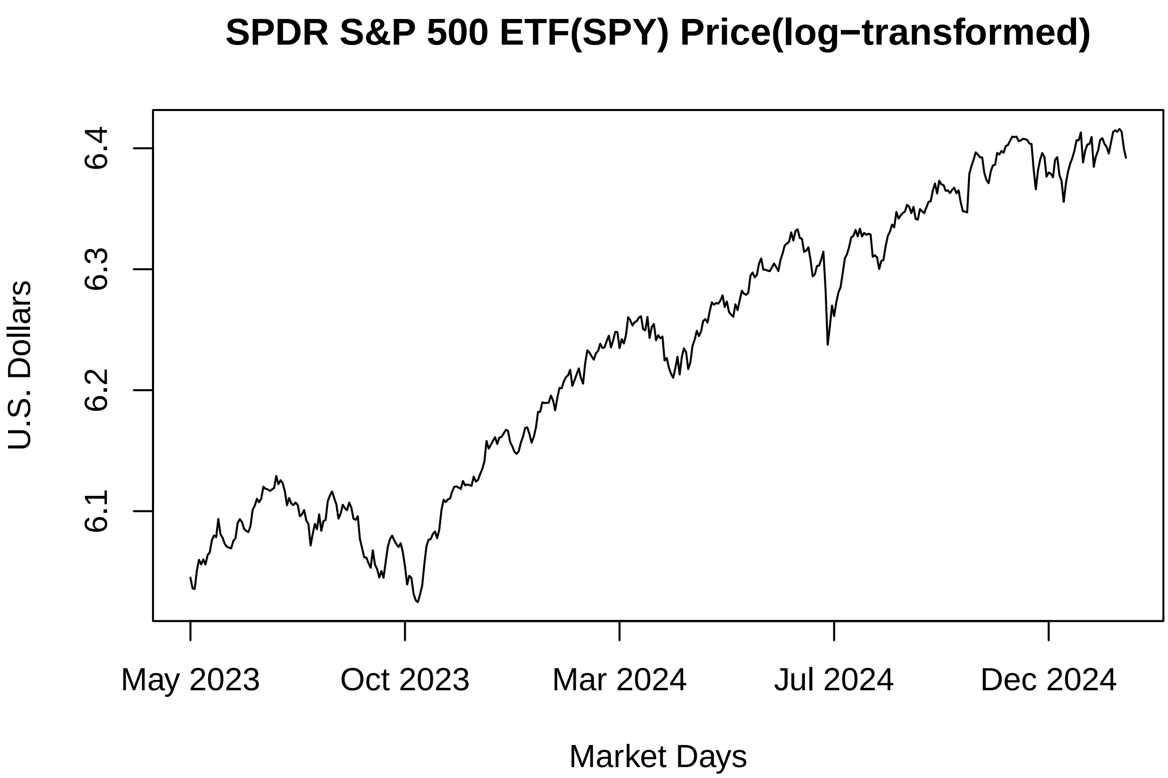
2. Data

The data set contains SPY opening prices from May 2023 to February 2025 (431 market days). The frequency of the data set is daily with the exception of weekends. The units are in US dollars, with the values all being non-negative and ranging from $413.56 to $611.54. I chose to analyze the daily price of the SPY fund as it strongly reflects the bulk of the US economy, and thought it would be interesting to see how its evolved recently. Although there are arguably easier alternatives to obtaining such data, this set in particular was extracted from an online API: polygon.io, https://polygon.io/docs/stocks/get_v2_aggs_ticker___stocksticker___range___multiplier____timespan____from____to. I believe the collectors of this data include mainly stock exchanges, and stock research firms, and I believe the data was also likely collected through stock exchanges. The dataset is important as it tracks a very popular and insightful ETF that is also very interlaced with the U.S. stock market as a whole. The purpose of studying this fund is to see if there are any noticeable trends or patterns which might help in developing future predictions for both this fund and the overall market.

3. Methodology

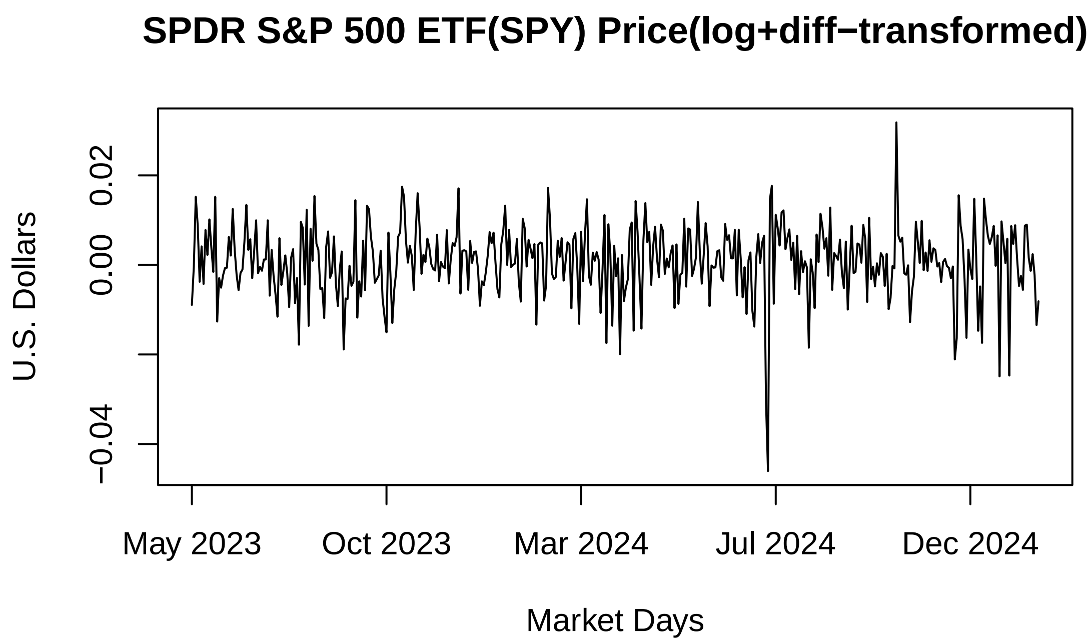
The analysis began with some basic exploratory data analysis including plotting the time series, and checking for certain characteristics like stationarity, varying volatility, and seasonality. After concluding that the data appeared non-stationary, I ran the data through an ADF test to confirm. Following this, I took the first-difference of the log of the data, which allowed the data to be further examined through ACF and PACF tests. These test surprisingly showed no indication of either a significant autoregressive term or a moving average, suggesting that an ARMA(0,0) might be the best fit for the first-differenced and logged data. I then verified this with an auto.arima() calculation, and it was confirmed to be the optimal model. Next, I fit this ARMA(0,0) into a sarima() to receive diagnostics on the residuals. From this point on the analysis focused on fitting a GARCH model. The parameters were chosen through examination of the ACF and PACF of the squared residuals from our sarima(). Finally, the GARCH model was used to forecast 2 weeks(10 market days) into the future, which was followed by a reversal of the initial modifications for the forecasted points so they could fit back into the original data set and be displayed.

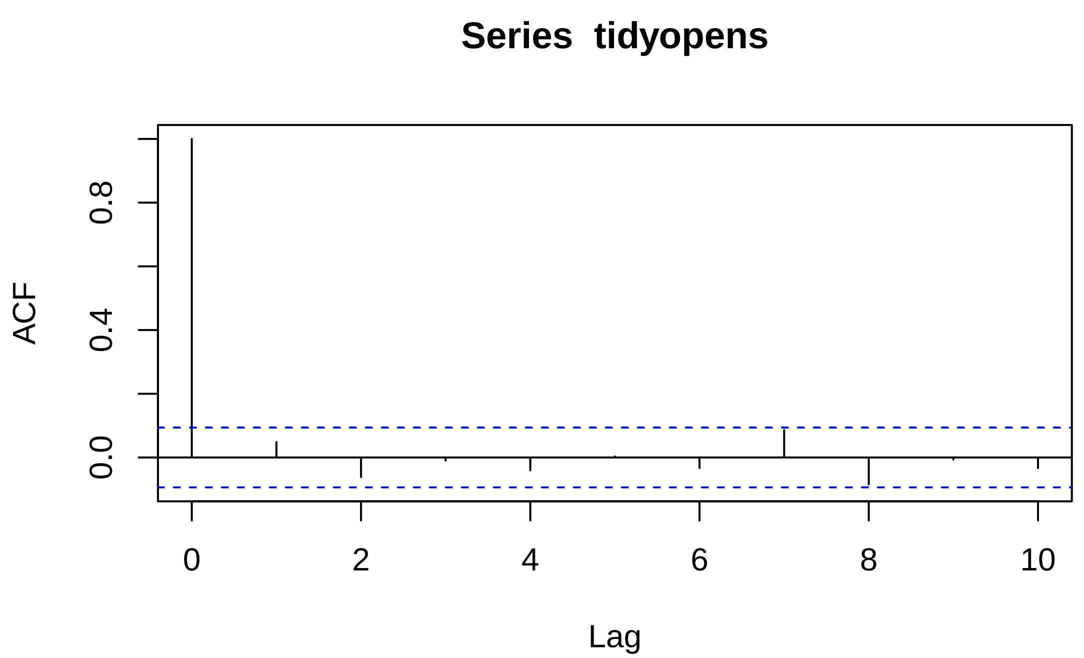
4. Results

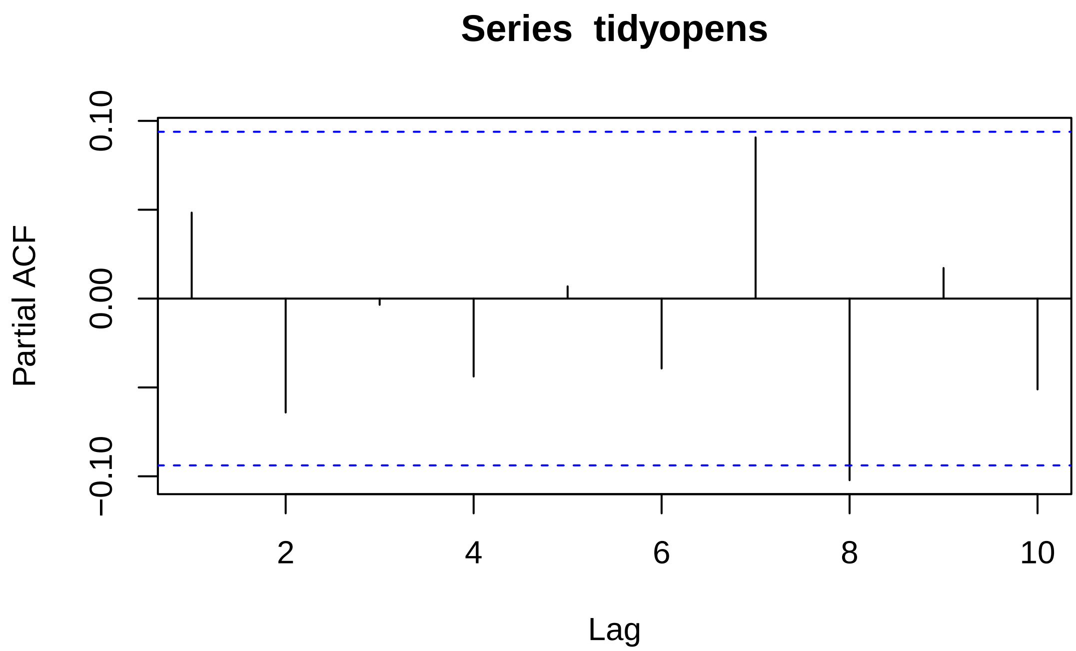



```
##
##  Augmented Dickey-Fuller Test
##
## data:  log(opens)
## Dickey-Fuller = -2.8839, Lag order = 7, p-value = 0.204
## alternative hypothesis: stationary
```

We first log-transform the data for simplification purposes. Since the original series appeared to not be stationary, and the Augmented Dickey-Fuller test failed to conclude that this series is stationary(p-value $>$ 0.05), we take the first difference.


SPDR S&P 500 ETF(SPY) Price(log+diff−transformed)
U.S. Dollars
0.02
0.00
−0.04
May 2023
Oct 2023
Mar 2024
Jul 2024
Dec 2024
Market Days


Series tidyopens
ACF
0.8
0.4
0.0
0
2
4
6
8
10
Lag

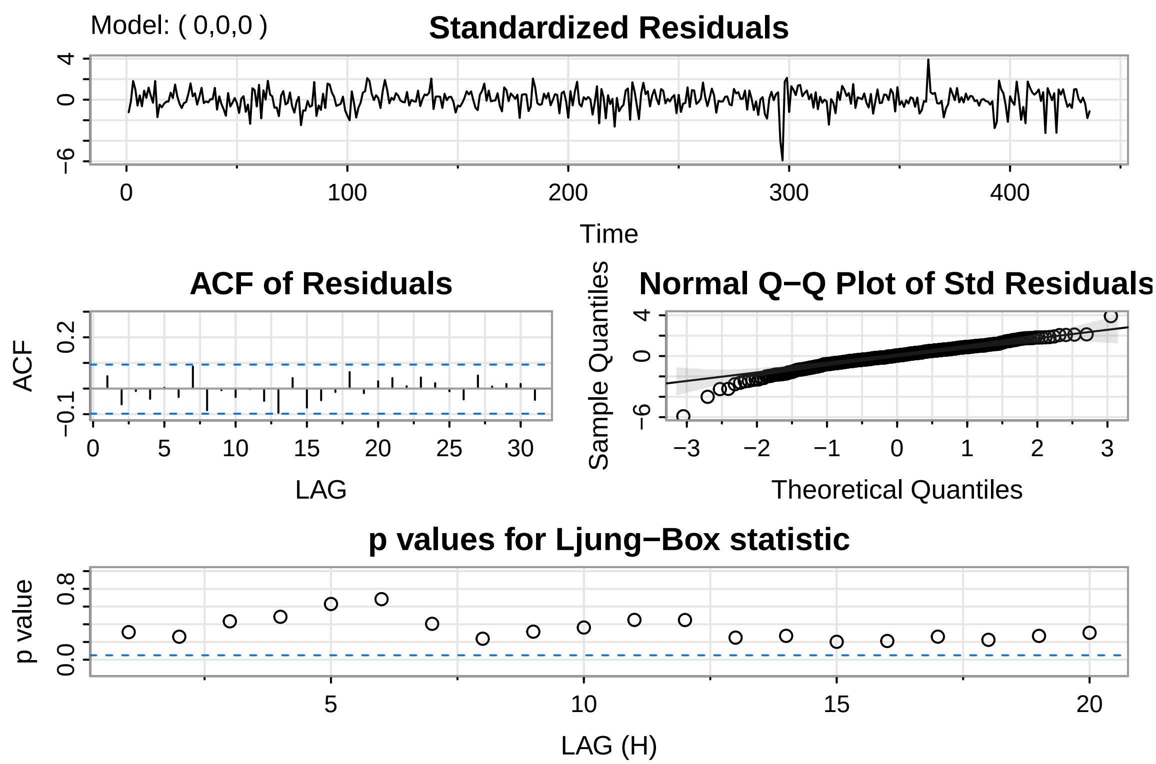


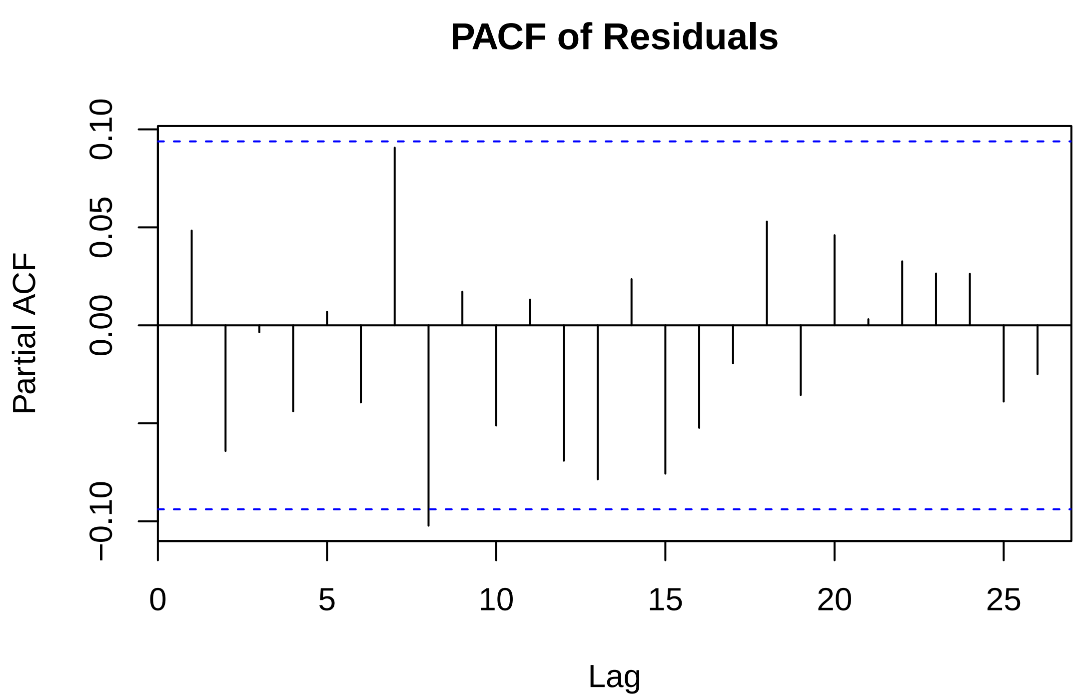
The ACF shows an immediate drop off after lag=0 suggesting there is no significant moving average(MA) component. The PACF shows small spikes at lag = 7,8, however, they don't seem significant enough to include in our model, so we will continue with no auto-regressive(AR) component. While the spike are very close to statistical significance, including an auto-regressive(AR) term would sacrifice interpretability for very minimal improvements. So, we continue with an ARMA(0,0).

```
## Series: tidyopens
## ARIMA(0,0,0) with non-zero mean
##
## Coefficients:
##         mean
##        8e-04
## s.e.  4e-04
##
## sigma^2 = 6.285e-05:  log likelihood = 1490.96
## AIC=-2977.91   AICc=-2977.89   BIC=-2969.76
```

```
## initial  value -4.838565
## iter   1 value -4.838565
## final  value -4.838565
## converged
## initial  value -4.838565
## iter   1 value -4.838565
## final  value -4.838565
## converged
## <><><><><><><><><><><><><><><>
##
## Coefficients:
##       Estimate    SE t.value p.value
## xmean    8e-04 4e-04  2.0827  0.0379
```

```
##
## sigma^2 estimated as 6.270117e-05 on 435 degrees of freedom
##
## AIC = -6.830079  AICc = -6.830058  BIC = -6.811374
##
```





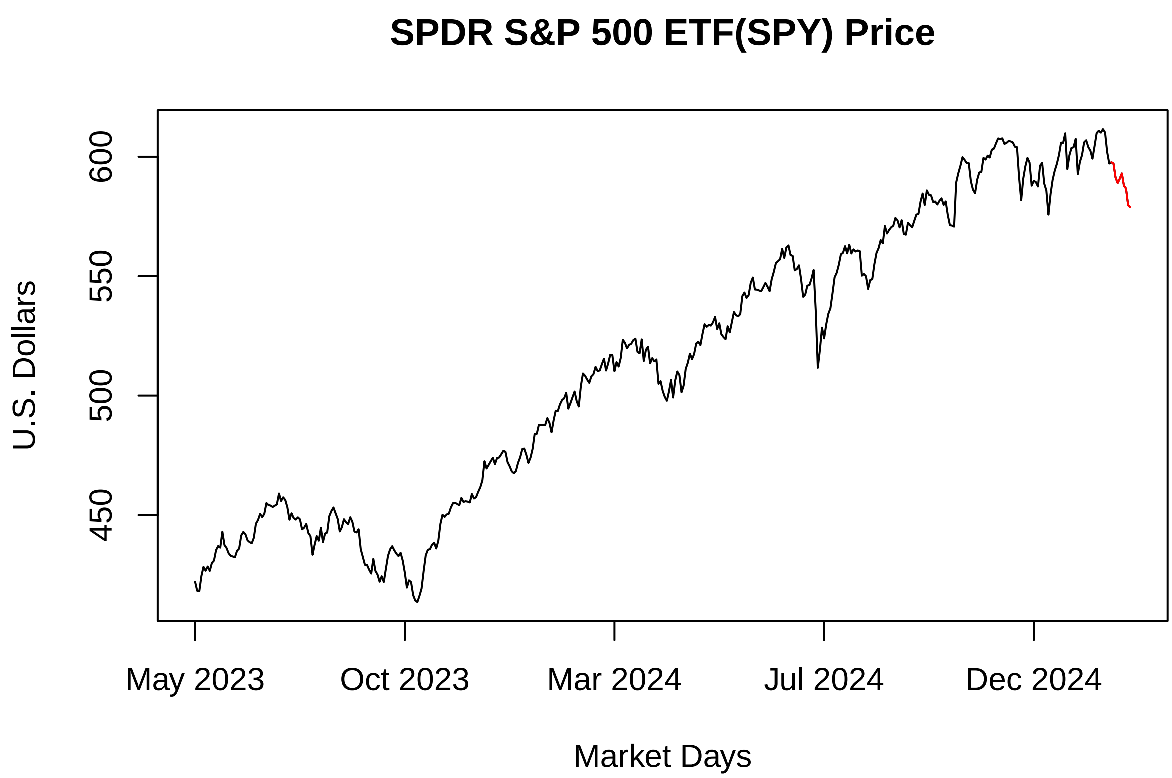
In fitting the auto.arima() we have confirmation for the ARMA(0,0) model, and we observe a mean of 8e-04, showing a slight positive trend in price. Our sarima() shows no irregular trends regarding the residuals. Since we are going to construct a GARCH model, we need to observe both the ACF and PACF of the residuals to determine its parameters. The ACF(shown in sarima()), and PACF show no significant spikes, and thus we will proceed with the default GARCH(1,1).

```
## Warning: package 'fGarch' was built under R version 4.4.3
```

```
## NOTE: Packages 'fBasics', 'timeDate', and 'timeSeries' are no longer
## attached to the search() path when 'fGarch' is attached.
##
## If needed attach them yourself in your R script by e.g.,
##         require("timeSeries")
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 0) + garch(1, 1), data = tidyopens,
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(0, 0) + garch(1, 1)
## <environment: 0x00000265b0e91540>
##  [data = tidyopens]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 7.3246e-04  5.2124e-05  1.4358e-01  1.0000e-08
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     7.325e-04   3.662e-04    2.000  0.04549 *
## omega  5.212e-05   9.220e-06    5.654 1.57e-08 ***
## alpha1 1.436e-01   5.153e-02    2.786  0.00533 **
## beta1  1.000e-08   1.443e-01    0.000  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  1502.579    normalized:  3.446283
##
## Description:
##  Sat Mar 15 15:32:31 2025 by user: reidp
##
##
```

```
## Standardised Residuals Tests:
##                                 Statistic      p-Value
##  Jarque-Bera Test   R    Chi^2  74.6514813 1.110223e-16
##  Shapiro-Wilk Test  R    W       0.9728868 3.072764e-07
##  Ljung-Box Test     R    Q(10)   9.4025285 4.943750e-01
##  Ljung-Box Test     R    Q(15)  18.2942737 2.475109e-01
##  Ljung-Box Test     R    Q(20)  22.9844177 2.895603e-01
##  Ljung-Box Test     R^2  Q(10)   4.3148977 9.320253e-01
##  Ljung-Box Test     R^2  Q(15)   6.1887971 9.763989e-01
##  Ljung-Box Test     R^2  Q(20)   9.7348030 9.727297e-01
##  LM Arch Test       R    TR^2    5.2242082 9.500677e-01
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -6.874218 -6.836808 -6.874384 -6.859454
```



There we have it! Using the GARCH model to simulate points, and converting them to fit back into the original dataset shows a bit of a decrease in the 2 weeks following February 25 2025.

5. Conclusion and Future Study

In conclusion, this study found interesting results. We observed that the ideal ARIMA model for the price of SPY over the last two years is an ARMA(0,1,0). Also, from our GARCH modelling we found that past shocks impact current volatility, but past volatility does not persist significantly. We concluded our study by using this information to forecast future data in which we saw a general decrease. For future study on SPY stock prices, there is likely more to explore concerning seasonality, with potential upsides coming from applications of spectral analysis.

##Appendix:

Python- import pandas as pd import requests url = "https://api.polygon.io/v2/aggs/ticker/SPY/range/1/day/2023-05-28/2025-02-25?adjusted=true&sort=asc&apiKey=MQkwjxSNUJKutT1VbndjgzjoFljeXluI" response = requests.get(url) first = response.json() opens = [i["o"] for i in first["results"]]

R- opens <- py$opens save(opens, file = "opens.rda")

load("opens.rda") opens <- unlist(opens) start_date <- as.Date("2023-05-28") end_date <- as.Date("2025-02-25") all_dates <- seq.Date(start_date, end_date, by = "day") weekdays_only <- all_dates[!weekdays(all_dates) %in% c("Saturday", "Sunday")] dates <- weekdays_only[1:length(opens)] plot.ts(opens, xlab = "Market Days", ylab = "U.S. Dollars", main = "SPDR S&P 500 ETF(SPY) Price", xaxt ="n") axis(1, at = seq(1, length(opens), by = 100), labels = format(dates[seq(1, length(opens), by = 100)], "%b %Y"))

library(reticulate) library(forecast) library(astsa) library(tseries)

plot.ts(log(opens), xlab = "Market Days", ylab = "U.S. Dollars", main = "SPDR S&P 500 ETF(SPY) Price(log-transformed)", xaxt ="n") axis(1, at = seq(1, length(opens), by = 100), labels = format(dates[seq(1, length(opens), by = 100)], "%b %Y")) adf.test(log(opens))

tidyopens <- diff(log(opens)) plot.ts(tidyopens, xlab = "Market Days", ylab = "U.S. Dollars", main = "SPDR S&P 500 ETF(SPY) Price(log+diff-transformed)", xaxt ="n") axis(1, at = seq(1, length(opens), by = 100), labels = format(dates[seq(1, length(opens), by = 100)], "%b %Y")) adf.test(log(opens)) acf <- acf(tidyopens, lag.max = 10, type="correlation") pacf <- pacf(tidyopens, lag.max = 10)

auto.arima(tidyopens, stepwise = FALSE, seasonal=TRUE) sarima(tidyopens, 0,0,0) res <- residuals(auto.arima(tidyopens)) pacf(res, main="PACF of Residuals")

library(fGarch) garch_mod <- garchFit(~arma(0,0)+garch(1,1), trace =FALSE, tidyopens) summary(garch_mod) fore <- predict(garch_mod, n.ahead = 10) set.seed(10) preds <- rnorm(10, mean = fore$meanForecast, sd = fore$standardDeviation) log_fore <- c() for (i in 1:length(preds)){ a = sum(preds[1:i]) log_fore <- append(log_fore, a) } last <- tail(opens,1) conv_vals <- exp(log_fore)*last all <- c(opens, conv_vals) plot.ts(all, xlab = "Market Days", ylab = "U.S. Dollars", main = "SPDR S&P 500 ETF(SPY) Price", xaxt ="n") axis(1, at = seq(1, length(opens), by = 100), labels = format(dates[seq(1, length(opens), by = 100)], "%b %Y")) lines((length(opens) + 1):length(all), conv_vals, col = "red", pch = 19)

##Sources:

https://www.investing.com/news/stock-market-news/spy-etf-shows-seasonal-performance-patterns-analysis-reveals-93CH-3818986 https://polygon.io/docs/stocks/get_v2_aggs_ticker__stocksticker__range__multiplier__timespan__from__to