Prediction of Gold Price

Shiyu Wu, Suyeon Song, Xiaowen Wu, Sophia Gan

Executive summary

This report examines how historical gold prices, T-bill rates, and S&P 500 indices can forecast future gold price movements, focusing on the impact of pre- and post-COVID-19 pandemic dynamics. Using data from Yahoo Finance for gold prices and the S&P 500 indices, and from the U.S. department of the Treasury for 13-week T-bill coupon rates, we applied ARIMA, VAR, and ARIMA with external regressors such as T-bills and S&P 500. These models were separately applied on data from January 2014 - December 2019 (pre-pandemic) and January 2020 - December 2023 (post-pandemic) periods to evaluate their accuracy in predicting gold prices under different economic conditions.

Here are our key findings:

- Pre-Pandemic Analysis: ARIMA with external regressors outperformed the other models although it struggled with large prediction intervals.
- Post-Pandemic Analysis: The same model captured the upward trends in gold prices, despite continued wide prediction intervals suggesting uncertainty.
- Model limitations: both models struggled with capturing long-term economic cycles, and residuals from these models did not meet normal distribution assumptions as well as white noise.

To enhance forecast accuracy, future models should include a broader range of economic indicators and employ machine learning to better manage complex relationships between them. Additionally, considering governmental regulations affecting gold price is also crucial. Given the economic volatility triggered by the pandemic, enhancing models with incorporating more comprehensive economic indicators and advanced techniques boost prediction reliability and aid investors in the metal markets.

1 Introduction

In the complex landscape of financial markets, investors are constantly seeking strategies to optimize their portfolios, balancing the pursuit of returns with the mitigation of risk. This report delves into the complex dynamics among three main market components: the S&P 500 index, representing the stock market; Treasury bills (T-bills), representing the bond market; and gold, symbolizing the commodity market. Each of these assets plays a critical role as a market signal and serves as a barometer for varying investor sentiments and macroeconomic trends.

The rationale for selecting these specific assets stems from their historical significance and distinct roles within the financial landscape. The S&P 500 is indicative of stock market health and corporate profitability. T-bills are considered a safe haven during periods of market instability. Gold is traditionally viewed as a hedge against inflation as it stores value in the time period of currency devaluation and geopolitical uncertainty.

Our analysis is focused on examining how gold prices are influenced and can be forecasted by the movements of the other two assets such as S&P 500 and T-bills since gold typically has a negative correlation with other financial assets like stocks and bonds. Particularly, the recent COVID-19 pandemic has dramatically altered economic landscapes, resulting in unprecedented market volatility. This fluctuation has highlighted

gaps in our understanding of how these interactions shift during the significant economic upheavals. Thus, this study aims to address these gaps by assessing whether traditional patterns of gold price movements persists during the pandemic, providing insights that could guide investors in reassessing traditional market behaviors and investment strategies.

In conclusion, the objectives of this report are to answer the following questions:

1. How well can the historical gold prices forecast future gold prices?
2. Can variations in T-bill rates and S&P 500 indices help predict future movements of gold price before and after the COVID-19 pandemic?

2 Methods

2.1 Data

We collected data from three main sources. The first data source consists of the daily closing prices of gold from 2014/01/01 to 2023/12/31, sourced from Yahoo Finance. This dataset contains 2498 rows and will serve as the response variable that we aim to forecast. The second data source comprises the daily closing data for the S&P 500, also from 2014 to 2023 and obtained from Yahoo Finance, with a total of 2516 rows. The third data source is 13 weeks T-bill coupon rate from 2014 to 2023, sourced from U.S. Department of the treasury. This dataset has 2501 rows. After merging all the data, we can have 2497 rows in total.

2.2 EDA

Initially, we conducted time series analysis for each variable by plotting their respective time series to identify any trends or seasonality. This preliminary visualization is crucial for understanding the underlying patterns in the data and determining the preprocessing steps needed for further analysis.

For any time series exhibiting trends or seasonality, we applied differencing—a technique used to make the series stationary. This step is essential because many time series forecasting models, including ARIMA, assume that the data are stationary. Differencing helps remove systematic changes over time, thereby stabilizing the mean of the time series.

After differencing, we examined the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots of the residuals for each time series. These plots are instrumental in checking whether the residuals behave like white noise, which would indicate that all systematic information that can be modeled has been captured, and only random fluctuations are left.

Additionally, we assessed the normality of the residuals using Quantile-Quantile (Q-Q) plots. A normal Q-Q plot compares the distribution of residuals against a perfectly normal distribution. Ideally, the points should fall approximately along a straight line if the residuals are normally distributed. This check is important because many statistical tests used in the analysis and validation of forecasting models rely on the assumption of normally distributed residuals.

Through these steps, we aimed to ensure that each time series was appropriately processed and analyzed to meet the assumptions required for robust statistical forecasting.

Gold

In Figure 1, the time series plot of gold prices displays an increasing trend over time, with a notable further increase during the COVID-19 period.

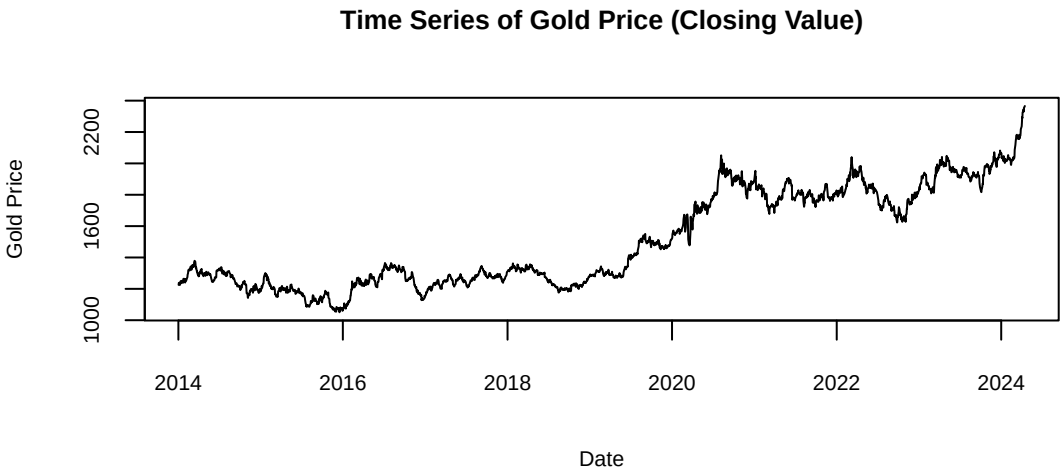

Figure 1: Time Series of Gold Price (Closing Value)

In Figure 2, the ACF plot shows significant spikes outside the error bounds, and PACF has also a spike at lag 1. This suggests the presence of temporal dependencies in the gold price, indicating that past gold prices have a significant influence on its future prices.

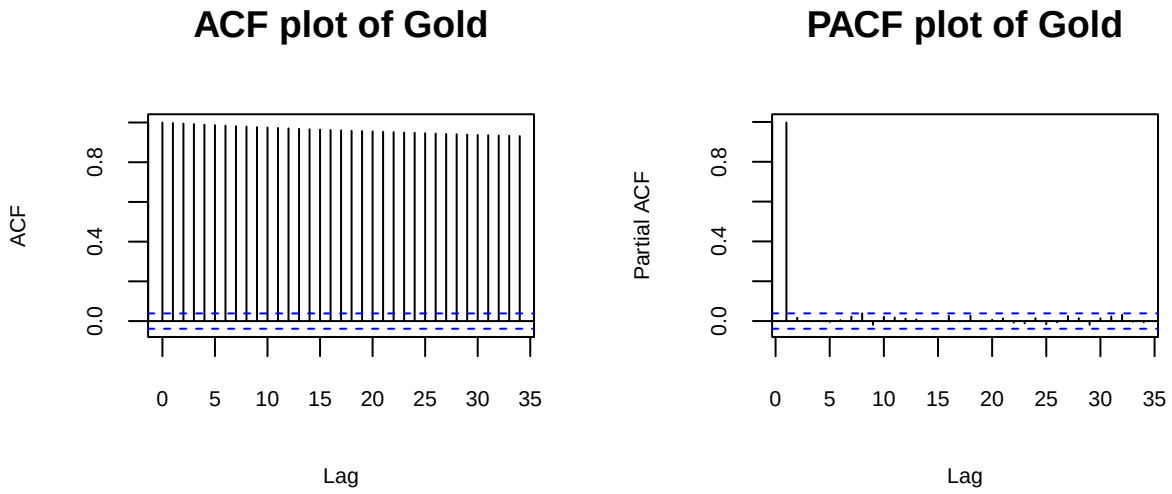

Figure 2: ACF and PACF of Gold Price Time Series

T-bills and S&P 500

To explore the influence of the S&P 500 and Treasury bill (T-bill) rates on gold prices before and after the outbreak of the COVID-19 pandemic, we analyzed time series data spanning from the beginning of 2014 to current. Initially, we plotted the time series for each variable using the original dataset to assess stationarity and identify any trends or seasonal patterns.

As indicated in Figure 3, all three datasets displayed an increasing trend without any obvious seasonal variations. Specifically, gold prices fluctuate within a range of approximately $1,100 to $1,500 before COVID and increases after. Notably, there was a drop in the price of gold at the beginning of 2020. Similarly, the S&P 500 index showed a steady increase from 2014 until the end of 2019, followed by a drop in early 2020. While for T-bill rates, the data indicated a gradual rise from 2014 up to the end of 2019. However, there was a noticeable decline in T-bill rates beginning in 2020, likely in anticipation of or in response to economic slowdown triggered by the pandemic.

These trends highlight the interconnectedness of these financial variables and underscore the importance of considering external economic factors when analyzing commodity prices like gold. The analysis further provides a baseline understanding of how these assets interact under pre-pandemic conditions, offering valuable insights for forecasting or historical comparison.

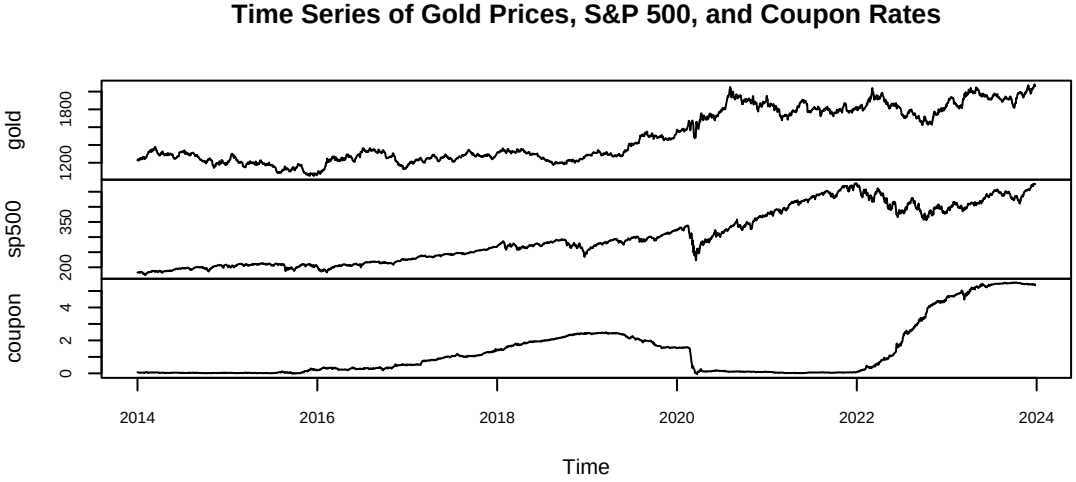

Figure 3: Time Series of Gold, S&P500, and Coupon Rates

2.3 Prediction

The forecasting of gold prices has been divided into two distinct phases: pre-pandemic (January 2014 to December 2019) and post-pandemic (January 2020 to December 2023). To prepare these data for further analysis and to address non-stationarity, we applied differencing to all three datasets (gold prices, S&P 500, and T-bill rates) by the maximum number required to achieve stationarity across the series. This method ensures consistency in the treatment of the datasets and comparability of results.

As illustrated in Figure 4, after differencing, the data sets exhibited a stationary trend, characterized by reduced trends or cycles, though with occasional spikes. Notably, in the differenced series for coupon rates, there is a significant spike in 2020. This particular spike reflects inherent volatility in financial data, which can be pronounced when differencing is used to achieve stationarity. The number of differences applied was determined to be two for each series based on the initial analysis, which was sufficient to stabilize the mean of the series without excessively distorting the data.

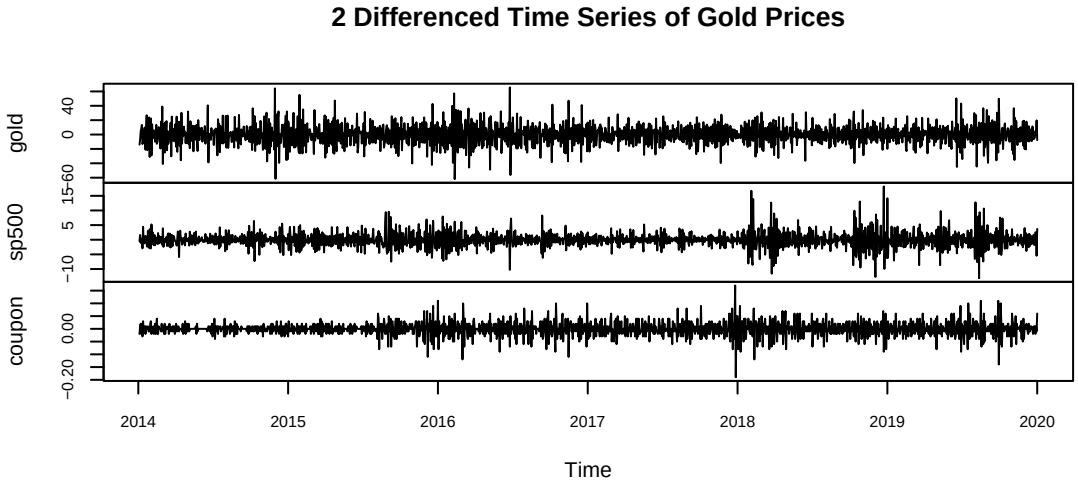

Figure 4: Time Series of All Variables after 2 Differenced

Using data from the pre-pandemic period, we predicted gold prices from January 2020 to December 2023 and then compared these predictions with the actual gold prices for the same period. Similarly, we forecasted gold prices from January 2024 up to the present and compared these forecasts with the actual data from the corresponding period.

For each time frame, three forecasting methodologies were employed: ARIMA (AutoRegressive Integrated Moving Average), VAR (Vector Autoregression), and ARIMA with external regressors such as T-bills and S&P 500. Since the data used to train each model covers different time periods, the models differ from each other. Finally, a comparative analysis of the forecasts generated by the different models across the two periods was conducted.

In conclusion, our analysis employed six distinct models, divided into two main categories based on the period they cover—pre-pandemic and post-pandemic

1. Pre-pandemic period

a) Gold Price forecast using ARIMA model

b) Gold Price forecast based on T-bill and S&P 500 using VAR model

c) Gold Price forecast based on T-bill and S&P 500 using ARIMA model with external regressors

2. Post-pandemic period

a) Gold Price forecast using ARIMA model

b) Gold Price forecast based on T-bill and S&P 500 using VAR model

c) Gold Price forecast based on T-bill and S&P 500 using ARIMA model with external regressors

3 Results

3.1 Pre-pandemic period

3.1.1 Gold Price forecast using ARIMA model

We fitted an ARIMA (5,1,5) model to the pre-pandemic gold prices and found a high correlation of 0.994 between the predicted and actual values in Figure 5.

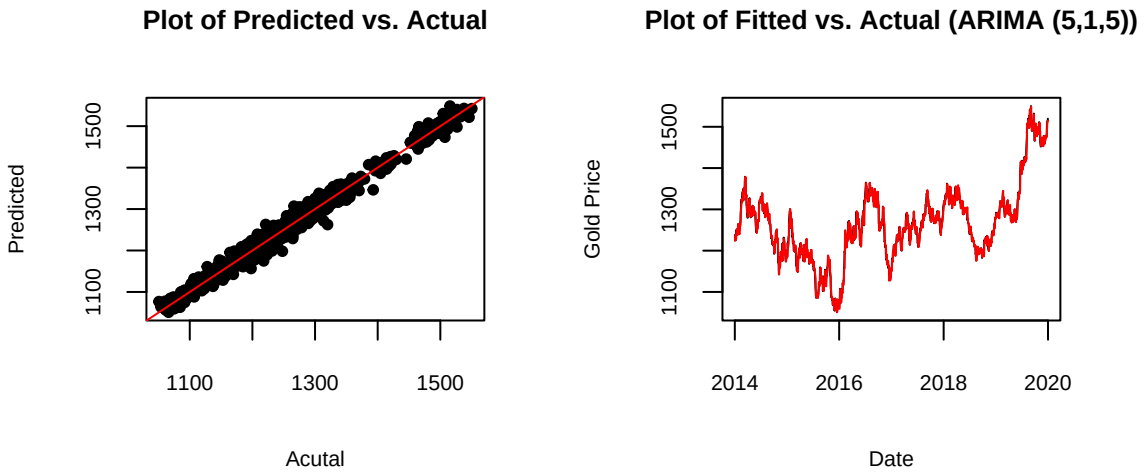

Figure 5: Plot of Predicted vs. Actual value with Correlation

In Figure 6, we were able to see the residuals appear to meet white noise assumption as spikes remain within the error bounds in both the ACF and PACF plots. However, in Figure 7, the Q-Q plot shows a fat tail at the edges, indicating the residuals are not normally distributed.

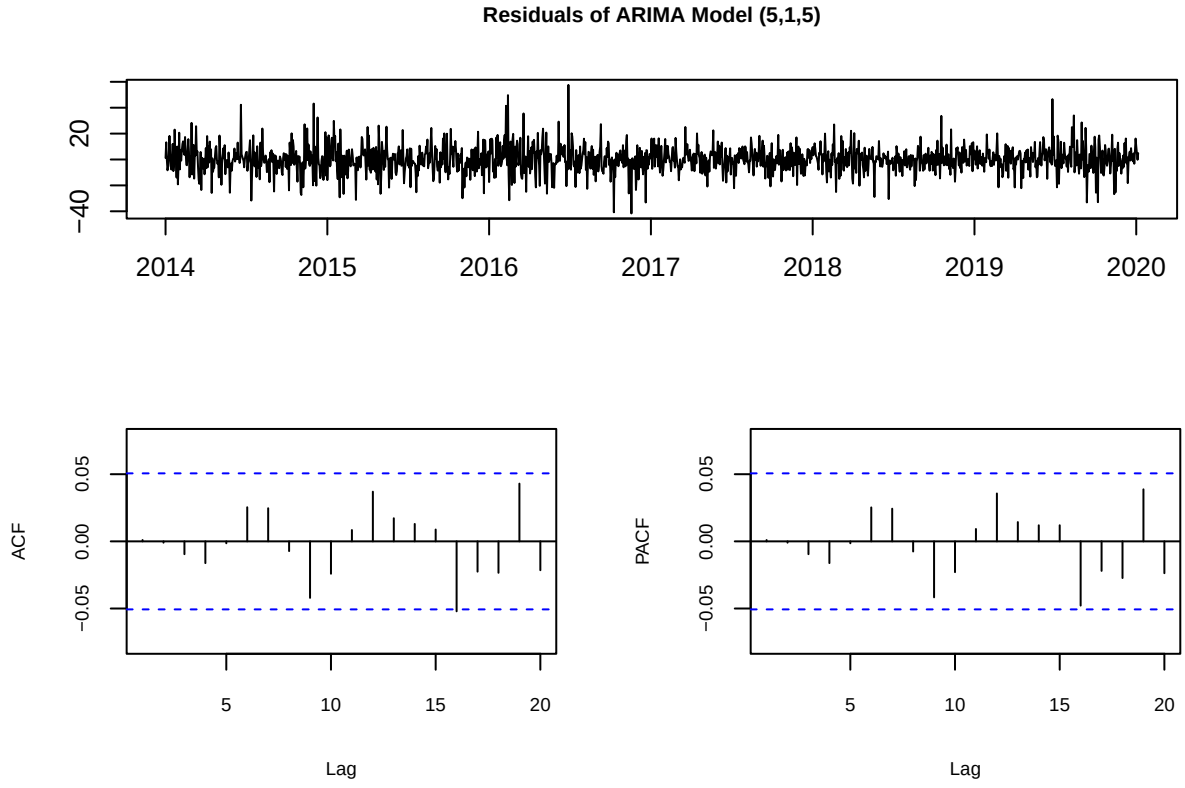

Figure 6: Residual plots of ARIMA (5,1,5) Model

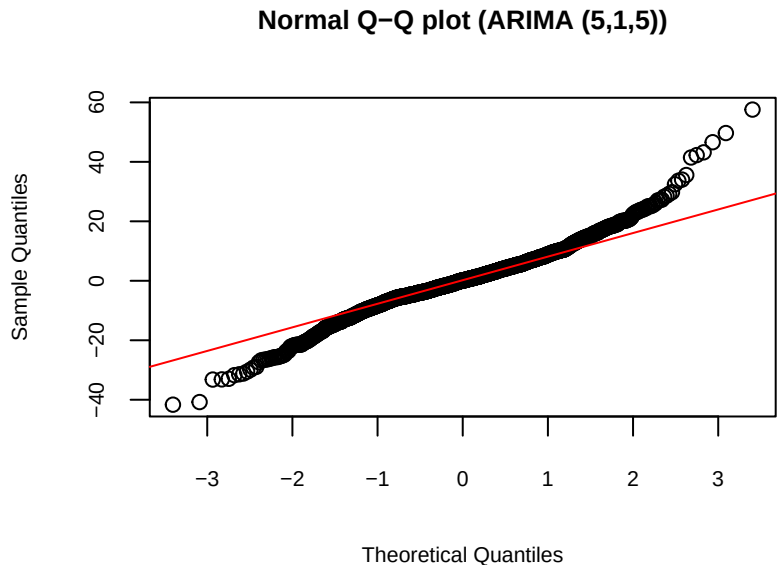

Figure 7: Q-Q plot of ARIMA (5,1,5) Model Residuals

In Figure 8, it is evident that the model fails to capture the fluctuations in gold prices. The actual gold price exceeds both the forecasted and past values, indicating that the model struggles to account for the volatility observed during unprecedented economic downturns. This discrepancy is particularly notable during periods of economic instability where gold often serves as a safe haven, thereby driving prices higher than expected. The model's training data did not adequately represent these extreme market conditions, resulting in less accurate predictions. Based on this model, we can conclude that the historical pre-pandemic gold prices do not help forecast future gold prices from January 2020 to December 2023 because of significant shifts in economic conditions and market behaviors during the pandemic.

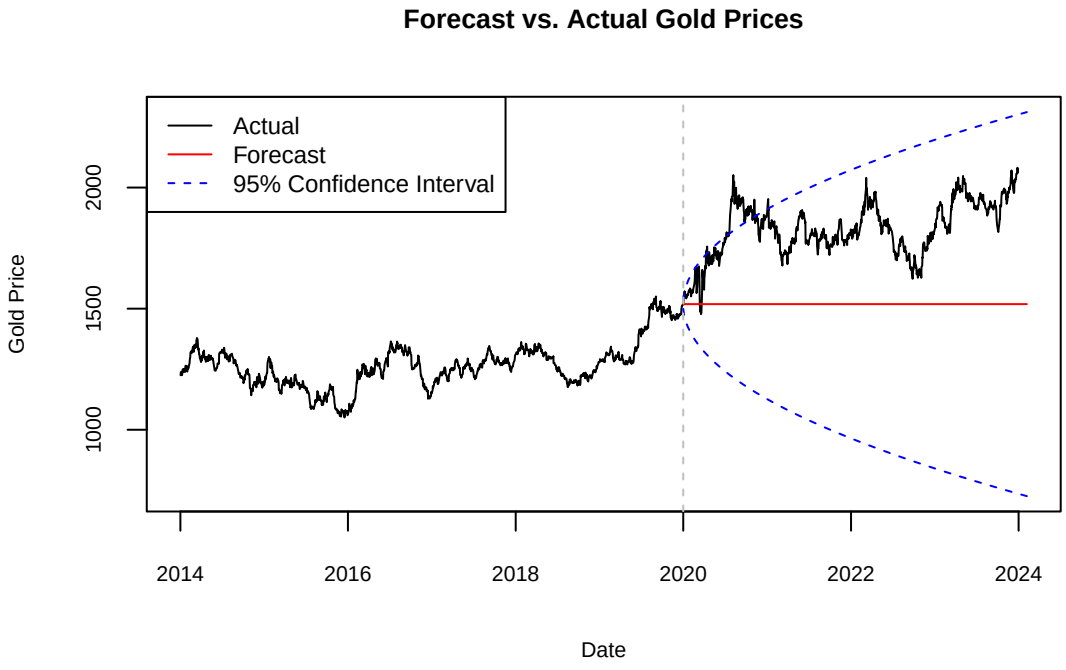

Figure 8: Forecast vs. Actual Gold Price from ARIMA (5,1,5) Model

3.1.2 Gold Price forecast using VAR model

A VAR model is a powerful tool for forecasting a vector of time series data and is particularly effective in elucidating the relationships among multivariate series by capturing dynamic changes in coefficients over time. For instance, to forecast the future price of gold using the relationship between historical data on Treasury bills, bond coupons, and gold, a VAR model is the optimal choice. The model outputs the differences across all imputed time series. Based on the Akaike Information Criterion (AIC), the VAR(10) model was selected for its fit. Forecast results from this model, illustrated in Figure 9, initially display minor fluctuations. However, as the forecasting horizon extends, these differences gradually stabilize and tend to converge towards zero, indicating a stabilization in the predictive variance of the model.

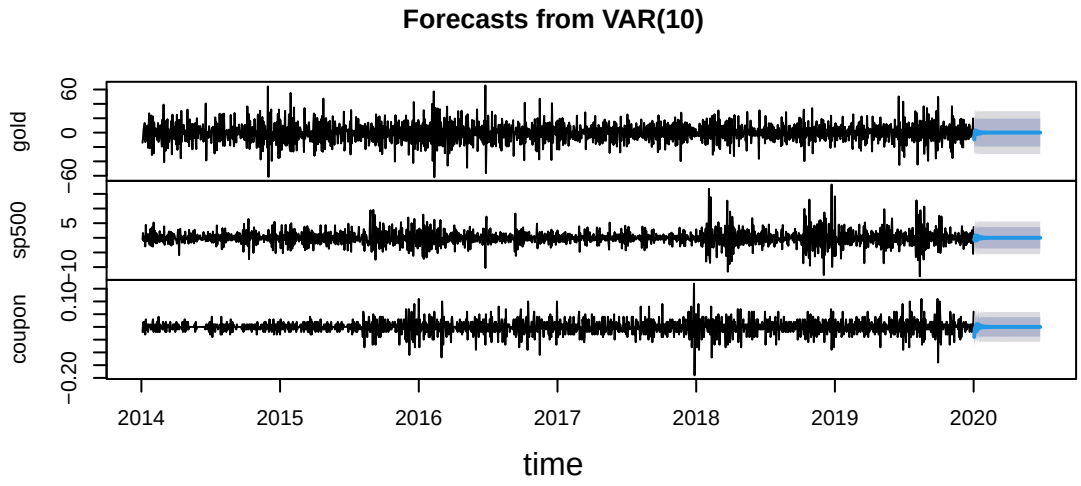

Figure 9: Forecast from VAR(10) Model (after second differencing)

Subsequently, the undifferenced forecast data and their corresponding confidence intervals were computed from the prediction outcomes. As depicted in Figure 10, the VAR(10) model demonstrates commendable accuracy in short-term forecasting. The accompanying graph, which zooms in on the forecasted data over a four-month period, underscores the model's efficacy in capturing the direction of data trends.

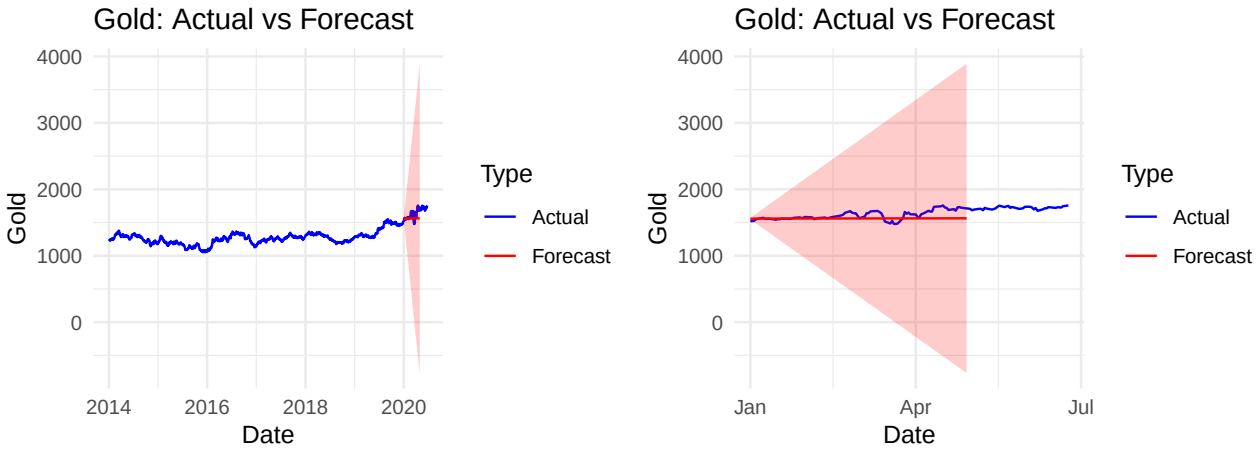

Figure 10: Forecast from VAR(10) Model (the undifferenced forecast data)

To evaluate the performance of the model, both a residual plot and a Q-Q plot were generated. The residual plot, shown in Figure 11 on the left, indicates that the residuals exhibit neither obvious trends nor

seasonality, confirming that the model has successfully captured the primary structure of the data. The Q-Q plot, displayed in Figure 11 on the right, further illustrates that the residuals generally adhere to a normal distribution, except for the tails, which are noticeably flatter. This analysis helps validate the statistical assumptions underlying the model and underscores its robustness in forecasting.

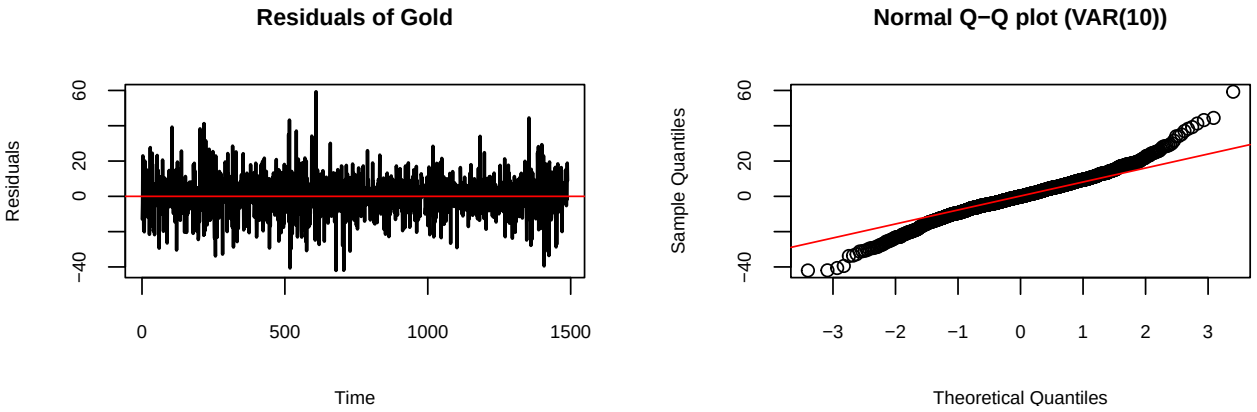

Figure 11: Residuals for validation of VAR(10)

However, its accuracy diminishes over the long term, failing to capture the actual trend of the data. Additionally, even though the forecasted confidence interval covers the actual price of gold, the forecast confidence intervals significantly widen, indicating increased uncertainty in the predictions. Throughout the visualizations and analysis, it underscores the model's limitations in long-term forecasting accuracy.

3.1.3 Gold Price forecast using ARIMA model with external regressors

In addition to the VAR(10) model, we also implemented the ARIMA(5,1,4) model augmented with external regressors, specifically the S&P 500 and Treasury bills, to predict gold prices. This approach leverages historical data to forecast future values while incorporating the influence of external factors that may impact predictions.

To accommodate the ARIMA model's requirements, we disregarded the need for data stationarity and seasonality, extending the maximum function order to 10, with the maximum parameters for autoregressive (p) and moving average (q) components set at 5. As shown in Figure 12, the model predicted a decline in gold prices. However, when compared with actual gold price movements, the observed values did not align with the predictions, highlighting a discrepancy between the forecasted and actual prices.

This deviation can largely be attributed to the model's reliance on historical data and established relationships between the datasets. The unexpected spike in actual gold prices, driven by the unforeseen impacts of the COVID-19 pandemic, was not captured by the model. Although the upper confidence interval managed to encompass some of the actual price movements during the forecasted period, it also widened as the forecasting horizon extended, signaling increased uncertainty in future predictions.

This outcome indicates that while the model effectively mirrors trends based on historical data, its predictive accuracy diminishes in the face of unforeseen events. Nevertheless, by analyzing the predicted trend relative to actual prices prior to the unexpected event, it is evident that the model fluctuates around historical values, underscoring its utility under stable conditions.

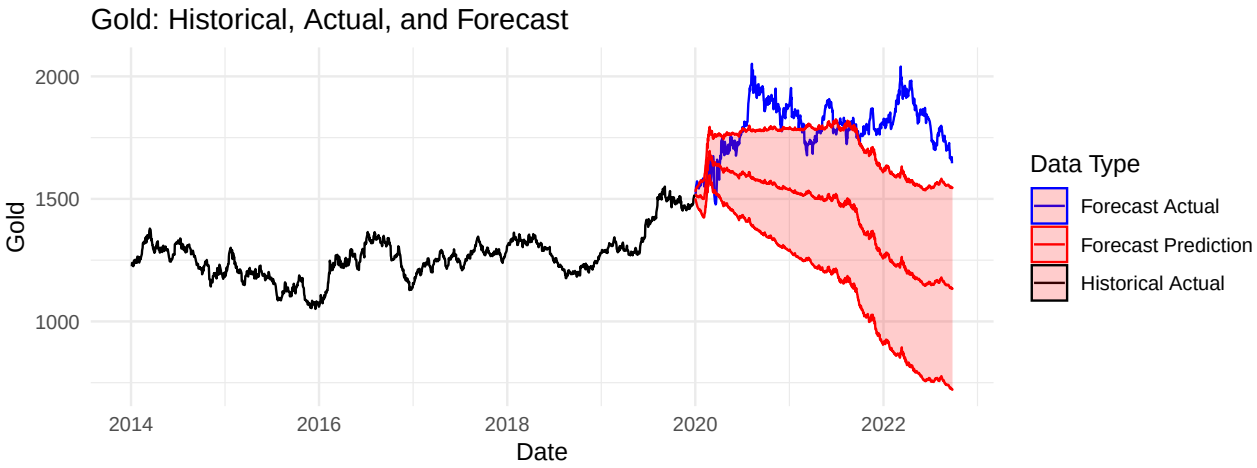

Figure 12: Forecast Gold Price for ARIMA Model with External Regressors

To evaluate the performance of the ARIMA model, we analyzed the residual plots as well as the ACF and PACF plots. As depicted in Figure 13, the residual plot on the left demonstrates that the residuals fluctuate around zero, which suggests that the model effectively captures the primary structure of the time series. This indicates that the key dynamics influencing gold prices are well-represented within the model.

The right side of Figure 13 presents the Q-Q plot for the model. This plot shows that the residuals align closely with the theoretical normal distribution line, confirming that the residuals are generally normally distributed. The exception, as noted, is in the tails of the distribution, which appear flatter than expected. This characteristic might indicate the presence of outliers or extreme values that the model does not fully account for, which could be a result of abrupt market changes such as those seen during the COVID pandemic.

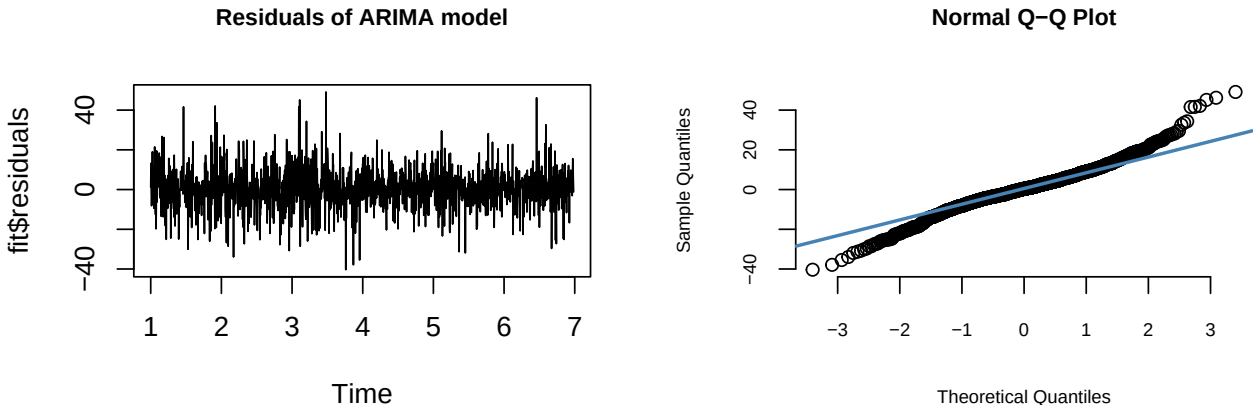

Figure 13: Residual Plot for ARIMA Model with External Regressors

Figure 14, which illustrates the ACF and PACF for the ARIMA(5,1,4) model, provides additional insights into the model's adequacy. The absence of significant spikes in either graph is a positive indication, suggesting that the model parameters have been appropriately selected to effectively forecast the future values. This lack of significant autocorrelation at various lags implies that the model has successfully captured the dependencies in the data, leaving no systematic pattern unaccounted for in the residuals.

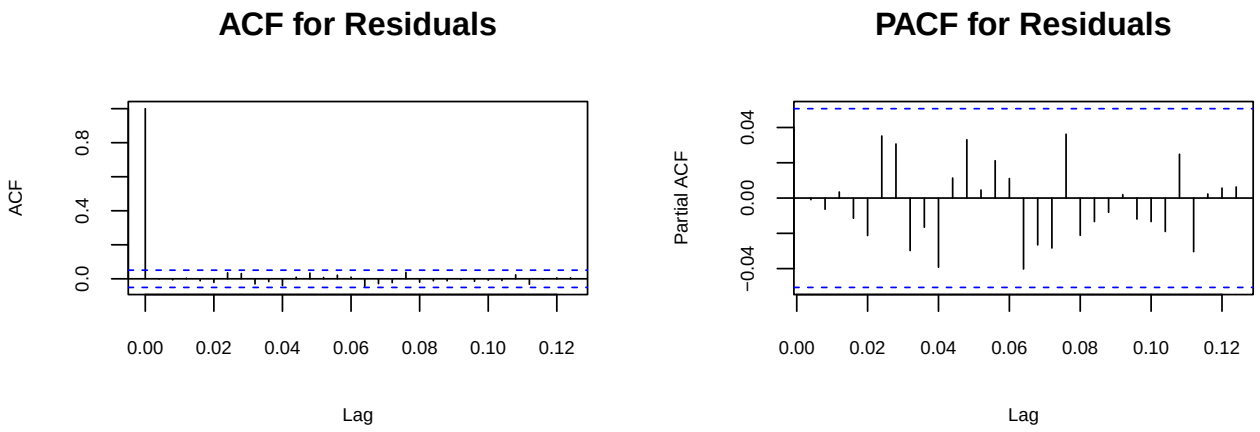

Figure 14: ACF and PACF Plot for ARIMA Model with External Regressors

Together, Figures 13 and 14 reinforce the effectiveness of the ARIMA(5,1,4) model with external regressors in capturing the essential characteristics of historical data. These diagnostics confirm that the model has been well-tuned to the data's underlying structure, making it a robust tool for forecasting gold prices under conditions consistent with the historical data. The demonstrated normal distribution of residuals and the proper parameter selection are critical in ensuring the reliability and accuracy of the model's predictions.

3.2 Post-pandemic period

3.2.1 Gold Price forecast using ARIMA model

We fitted an ARIMA (2,1,2) model to the post-pandemic gold prices based on AIC and found a high correlation of 0.986 between the predicted and actual values in Figure 15.

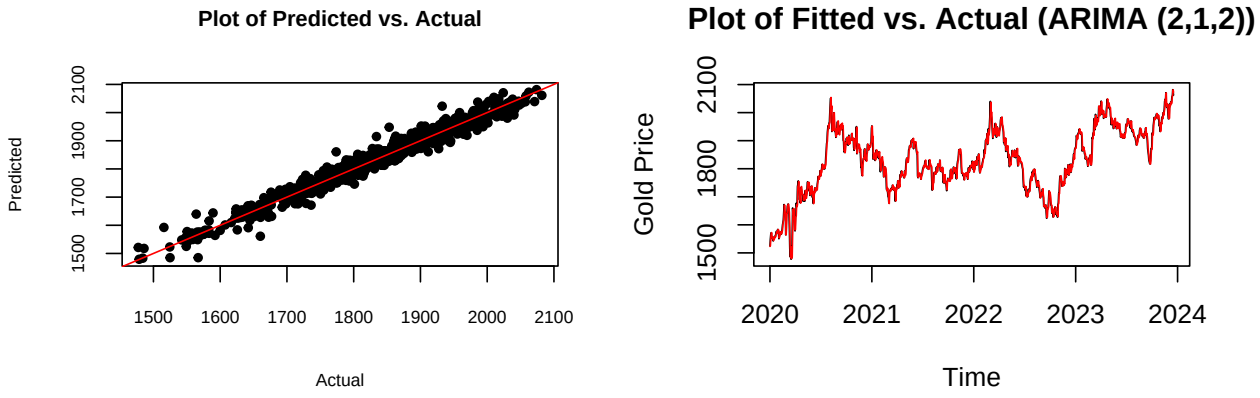

Figure 15: Plot of Predicted vs. Actual value with Correlation

In Figure 16, the analysis suggests that the residuals conform to the white noise assumption since the spikes are confined within the error boundaries in both the ACF and Partial PACF charts. Conversely, Figure 17 presents a Q-Q plot where the presence of fat tails at the extremes suggests that the residuals do not follow a normal distribution.

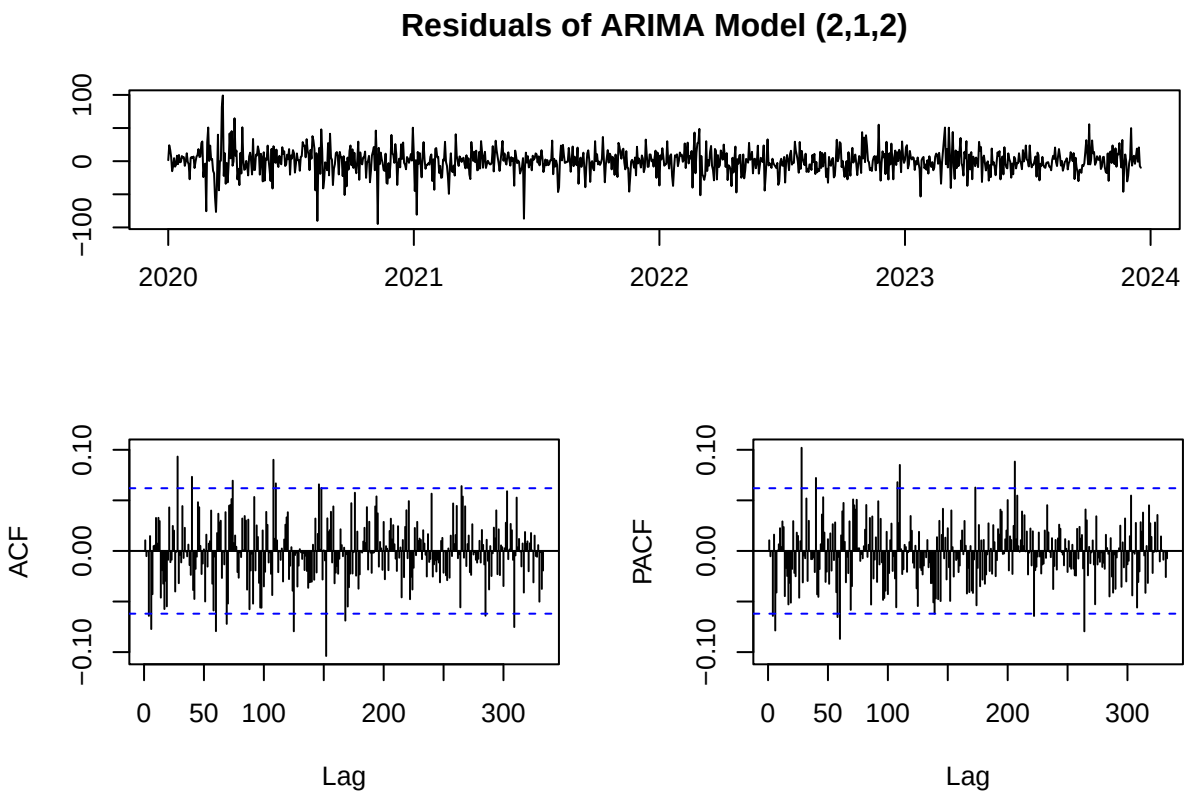

Figure 16: Residual plots of ARIMA (2,1,2) Model

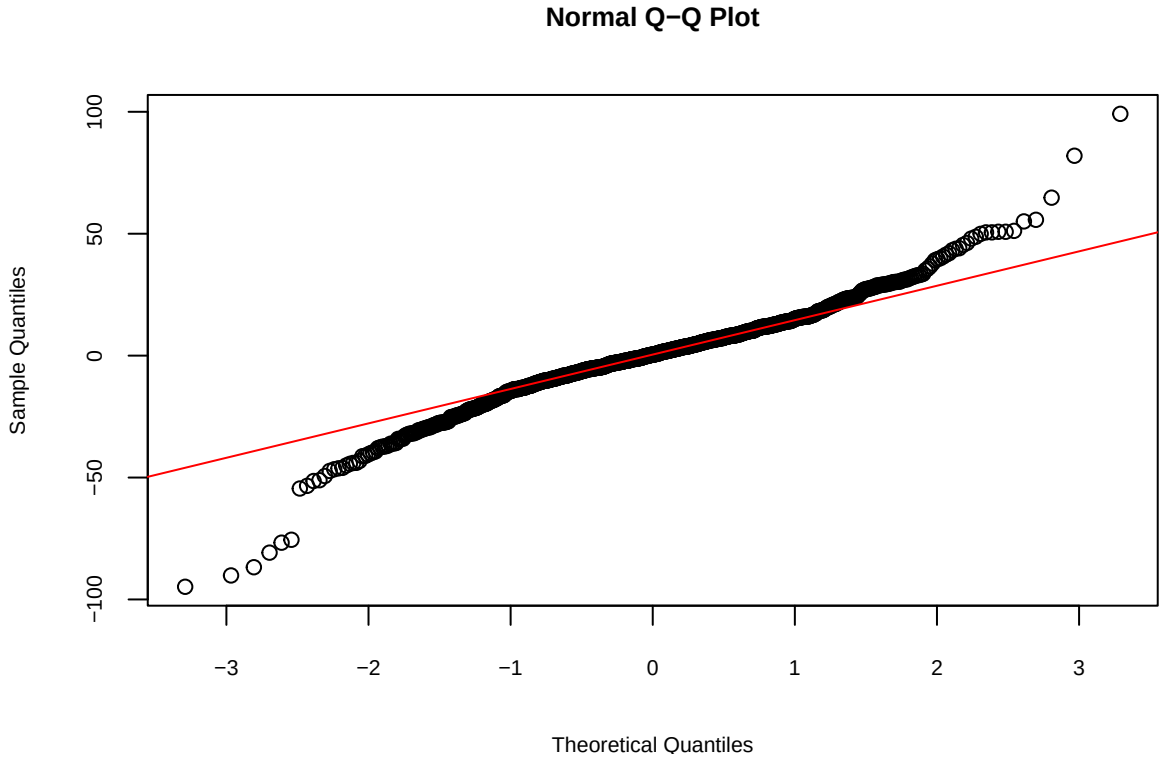

Figure 17: Q-Q plot of ARIMA (2,1,2) Model Residuals

In Figure 18, it is clear that the model has captured the trend in gold prices, successfully predicting an increase. However, the actual gold prices exceeded the forecasted values, indicating that the model's prediction was not very accurate in magnitude. In other words, although the predicted confidence interval includes the actual price, it still shows a certain degree of uncertainty. This may be due to gold prices being influenced by many factors, and during times of external instability, relying solely on historical data to predict gold prices may capture the general trend but cannot predict with high accuracy.

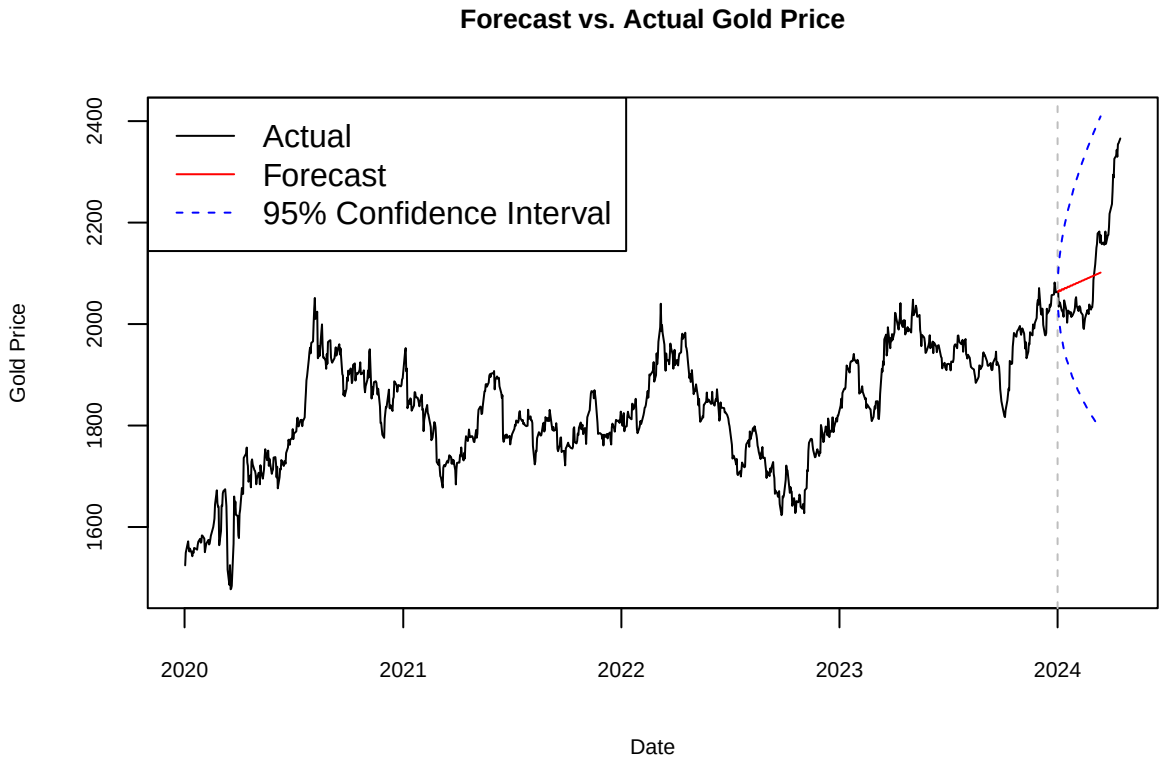

Figure 18: Forecast vs. Actual Gold Price from ARIMA (2,1,2) Model

3.2.2 Gold Price forecast using VAR model

Based on the AIC, the VAR(9) model was selected for fitting. The model's forecast results depicted in Figure 19. It is observable that after minor fluctuations, the forecasted values tend to converge towards zero (these forecasted results represent the outcomes after second differencing).

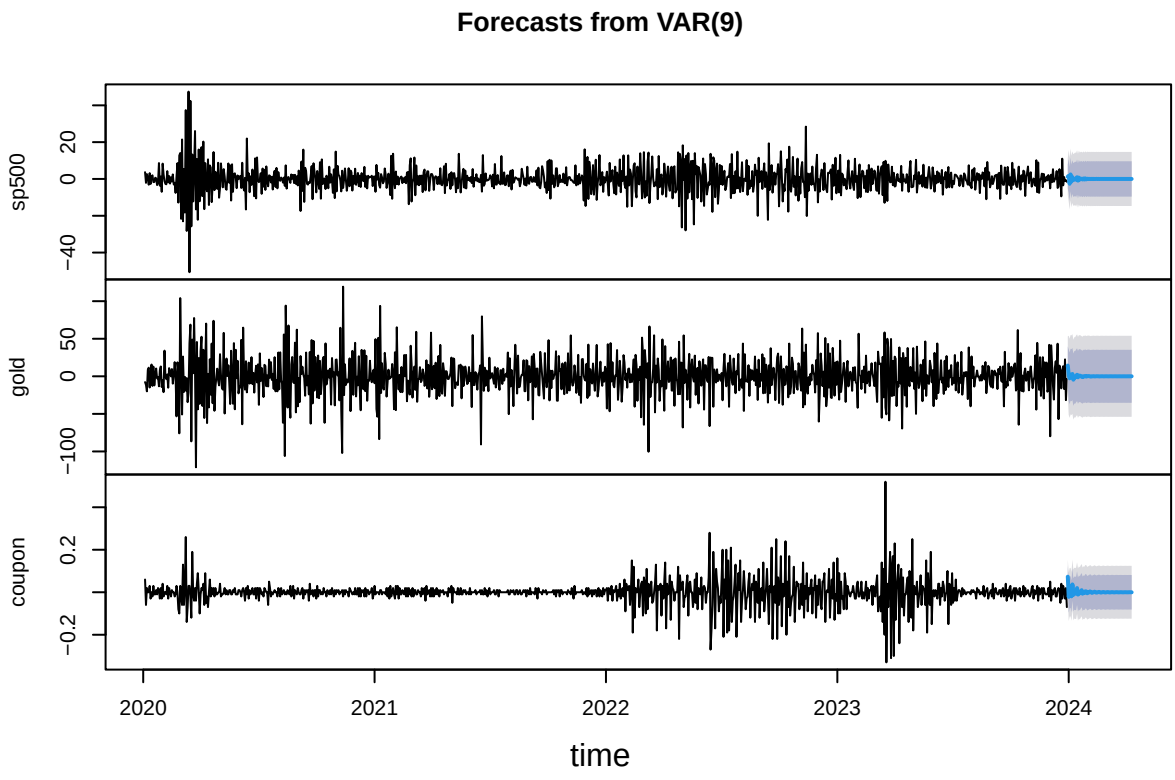

Figure 19: Forecast fromVAR(9) Model (after second differencing)

Subsequently, the undifferenced forecast data were calculated based on the prediction results. Figures 20 illustrate that the VAR(9) model achieves relatively accurate results in short-term forecasting, successfully capturing the direction of the data trends. However, its accuracy diminishes over the long term, failing to capture the sustained trends of the data. Additionally, the forecast confidence intervals significantly widen, indicating increased uncertainty in the predictions. This analysis underscores the model's limitations in long-term forecasting accuracy.

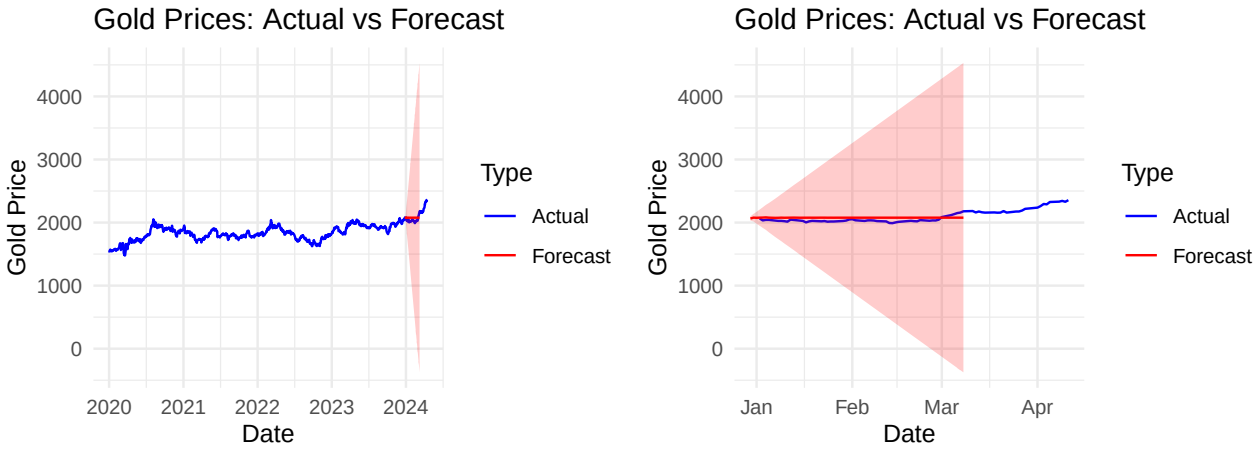

Figure 20: Forecast from VAR(9) Model (the undifferenced forecast data)

Figure 21 indicates that the residuals do not exhibit any obvious patterns or trends, suggesting that the model has captured the main structure of the data.

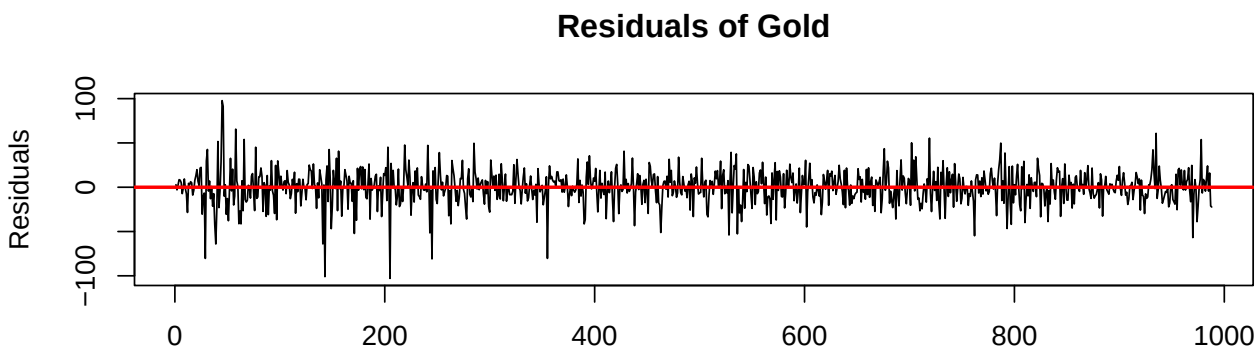

Figure 21: Residuals Plot for VAR(9) Model (Gold)

Figure 22 shows that the points largely adhere to the line, with some deviations at the beginning and end, suggesting that the residuals are mostly normally distributed.

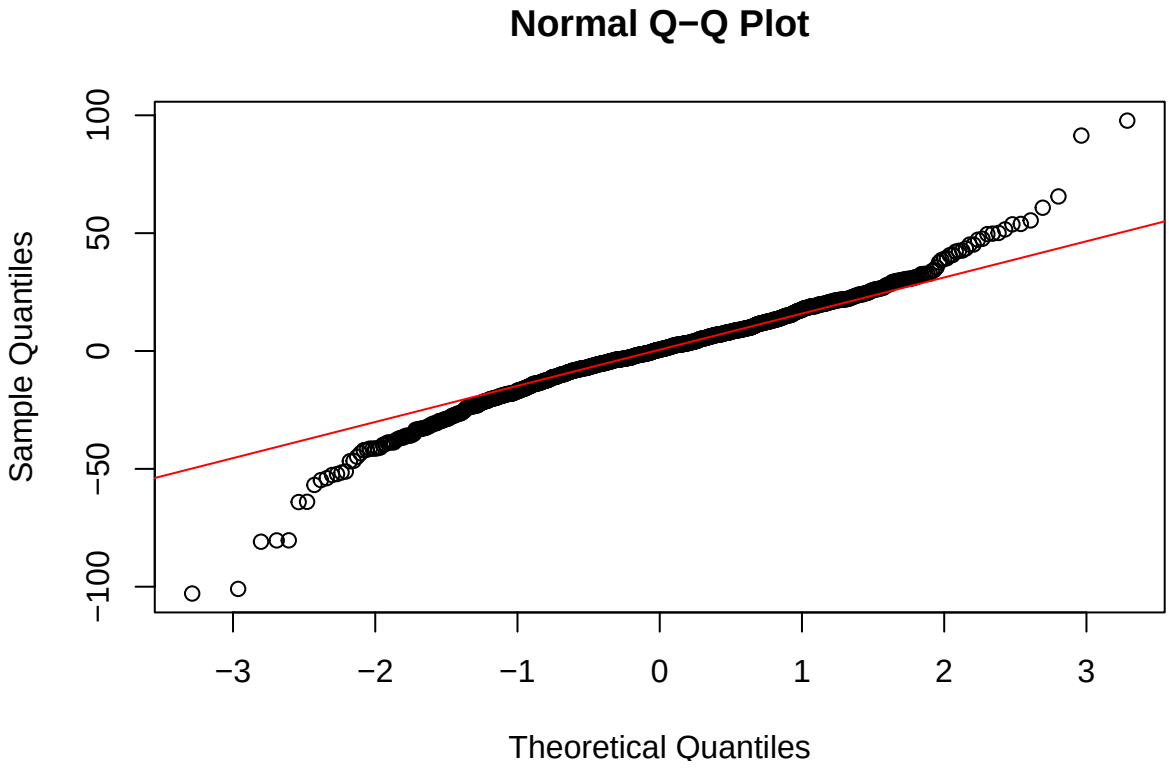

Figure 22: Q-Q Plot for VAR(9) Model (Gold)

3.2.3 Gold Price forecast using ARIMA model with external regressors

We selected the ARIMA(4,1,3) model with external regressors, SP500 and T-bills, for predicting gold prices. As illustrated in Figure 23, the model forecast an uptrend in gold prices. When compared to the actual gold prices, the observed values fall within our predictive model. This alignment suggests that the model is effectively capturing the underlying price dynamics of the gold market.

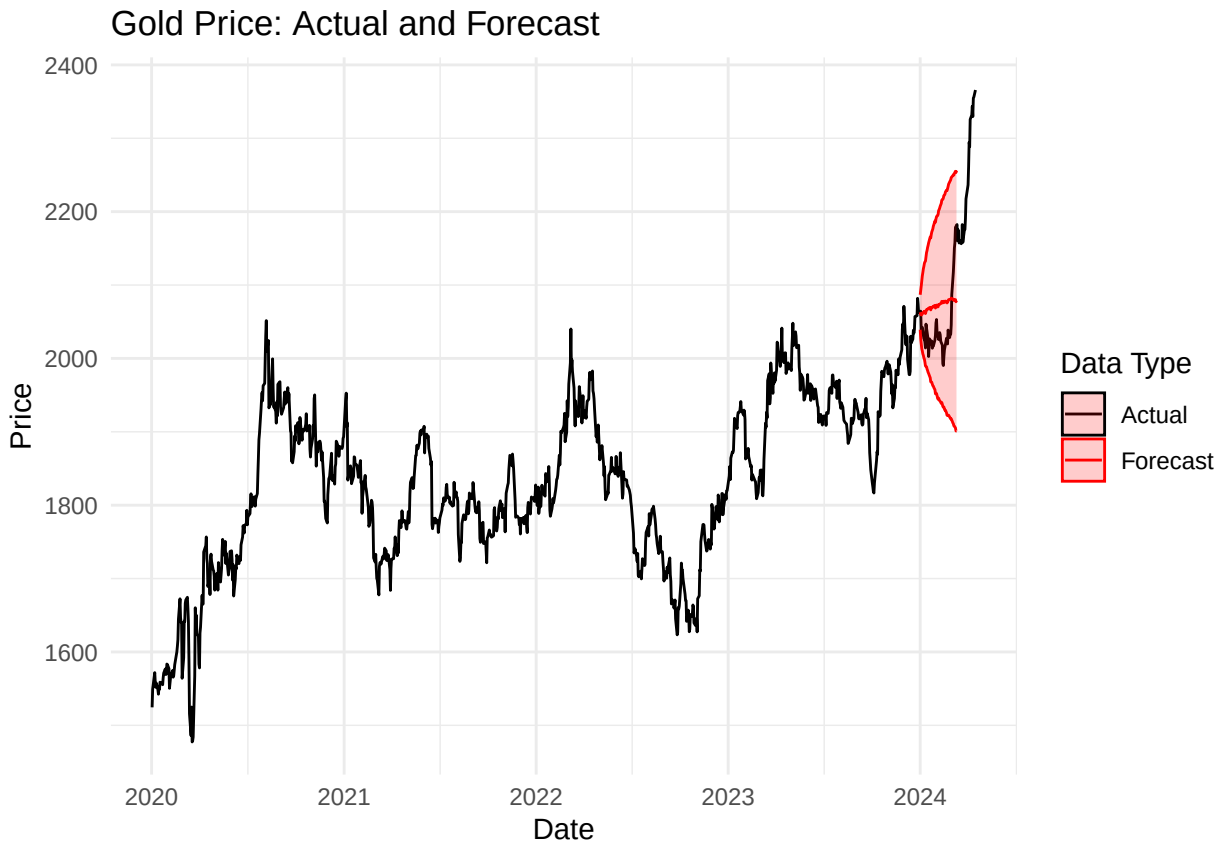

Figure 23: Forecast Gold Price for ARIMA Model with External Regressors

Form Figure 24, the residuals fluctuate around the zero line, suggesting no significant bias in the predictions. Figure 25 demonstrates that the standardized residuals mostly align with the theoretical quantiles represented by the dashed line, pointing to normality in the distribution of residuals. However, some deviations from the line, particularly at the ends, suggest that extreme values are not entirely consistent with a normal distribution. Both plots show that ARIMA(4,1,3) models with external regressors capture the characteristics of data.

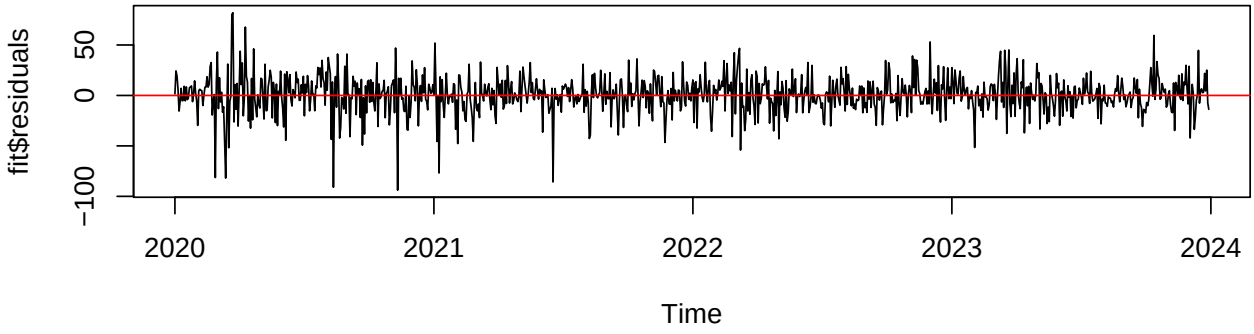

Figure 24: Residual Plot for ARIMA Model with External Regressors

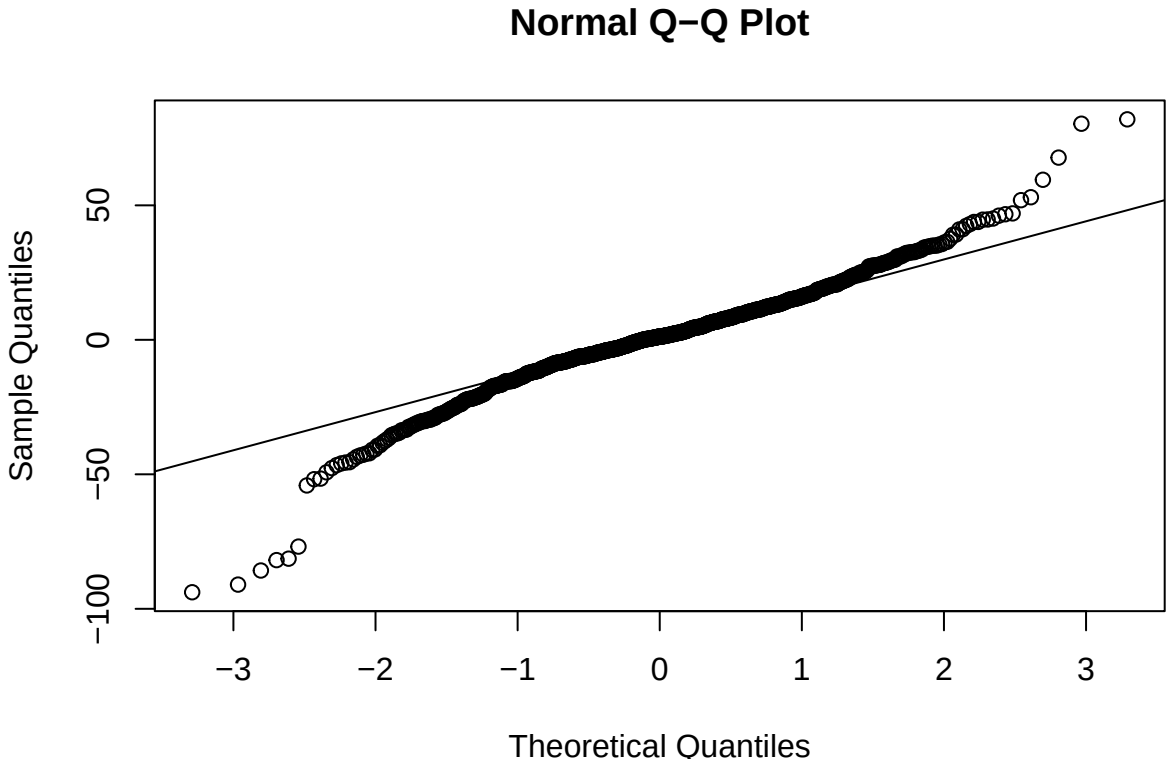

Figure 25: Q-Q Plot for ARIMA Model with External Regressors

4 Discussion

In conclusion, before COVID-19, the ARIMA model with external regressors such as T-bills and S&P 500 index variables outperformed both VAR and ARIMA models. It captured trends effectively but struggled with initial fluctuations triggered by COVID-19. Specifically, the forecasted values from the model followed the historical trajectory of the gold prices but failed to account for dramatic changes during the pandemic.

After COVID-19, the prediction intervals of the ARIMA model with external regressors such as T-bills and S&P 500 variables captured the actual time series of gold. The forecasts predicted the trend of the gold price; however, the wide confidence intervals, while generally covering the actual values, indicated a degree of uncertainty in these predictions.

Both models demonstrate limitations in capturing long-term economic cycles or reacting to the instant changes. The residual plots from the financial time series data do not meet the criteria for normal distribution and white noise, as evidenced by sharp peaks and heavy tails. Furthermore, gold prices can be influenced by other macroeconomic factors and market sentiment beyond the impacts of T-bills and S&P 500.

To enhance the accuracy and comprehensiveness of our analysis, we need to incorporate additional economic indicators that could influence gold prices into our model. Employing machine learning techniques will also allow us to better handle the complex relationships between variables, thereby improving the model's predictive accuracy. Additionally, considering the impact of governmental regulations on gold prices is crucial for refining our forecasting model further.

Appendix

Notice how the appendix below gathers all the code blocks above and nicely pastes them together.

```
############################
# STYLE EDITS: IGNORE THIS
############################

knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
knitr::opts_chunk$set(echo = FALSE)

library(quantmod)
gold_data <- getSymbols("GC=F", src = "yahoo", from = "2014-01-01",
                        to = "2024-04-16", auto.assign = FALSE)
gold <- na.omit(Cl(gold_data))
colnames(gold) <- "close"
gold_df <- data.frame(
  date = index(gold),
  close = as.numeric(gold$close))
gold_df$date <- as.Date(gold_df$date, format = "%m/%d/%Y")
gold_df <- gold_df[order(gold_df$date), ]
gold_ts <- ts(gold)
plot(gold_df$date, gold_ts,
  main = "Time Series of Gold Price (Closing Value)",
  type = "l", xlab = "Date", ylab = "Gold Price",
  cex.main = 0.8,
  cex.lab = 0.7,
  cex.axis = 0.7)
par(mfrow=c(1,2))
acf(gold_ts, main = "ACF plot of Gold",
    cex.main = 0.8,
    cex.lab = 0.7,
    cex.axis = 0.7)
pacf(gold_ts, main = "PACF plot of Gold",
     cex.main = 0.8,
    cex.lab = 0.7,
    cex.axis = 0.7)
data <- read.csv("three_before_covid.csv")
data$date <- as.Date(data$date, format="%Y-%m-%d")
names(data)[names(data) == "gold"] <- "temp"
names(data)[names(data) == "sp500"] <- "gold"
names(data)[names(data) == "temp"] <- "sp500"

data_ts <- ts(data[, c("gold", "sp500", "coupon")], start=c(2014, 1), frequency=250)
plot(data_ts, main = "Time Series of Gold Prices, S&P 500, and Coupon Rates",
     cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
library(tseries)
library(forecast)
library(ggplot2)
library(dplyr)
library(lubridate)
library(vars)
library(urca)
```

```r
library(rugarch)
library(readr)
data_ts <- ts(data[, c("gold", "sp500", "coupon")], start=c(2014, 1),
              end = c(2020, 1), frequency=250)

num_diff <- max(c(ndiffs(data_ts[, "gold"]),
                  ndiffs(data_ts[, "sp500"]), ndiffs(data_ts[, "coupon"])))

gold_diff <- diff(data_ts, differences = num_diff)
plot(gold_diff, main = "2 Differenced Time Series of Gold Prices",
     cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)

library(quantmod)
b <- getSymbols("SPY",
  src = "yahoo",
  from = "2014-01-01",
  to = "2024-04-16",
  auto.assign = FALSE)
sp500 <- na.omit(Cl(b))
colnames(sp500) <- "close"
sp500_df <- data.frame(
  date = index(sp500),
  close = as.numeric(sp500$close))
sp500_df$date <- as.Date(sp500_df$date, format = "%m/%d/%Y")
sp500_df <- sp500_df[order(sp500_df$date), ]

tb_2014 <- read.csv("t-bill/2014.csv")
tb_2015 <- read.csv("t-bill/2015.csv")
tb_2016 <- read.csv("t-bill/2016.csv")
tb_2017 <- read.csv("t-bill/2017.csv")
tb_2018 <- read.csv("t-bill/2018.csv")
tb_2019 <- read.csv("t-bill/2019.csv")
tb_2020 <- read.csv("t-bill/2020.csv")
tb_2021 <- read.csv("t-bill/2021.csv")
tb_2022 <- read.csv("t-bill/2022.csv")
tb_2023 <- read.csv("t-bill/2023.csv")

# extract a column of "X13.WEEKS.COUPON.EQUIVALENT"
cp_2014 <- (tb_2014[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2015 <- (tb_2015[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2016 <- (tb_2016[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2017 <- (tb_2017[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2018 <- (tb_2018[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2019 <- (tb_2019[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2020 <- (tb_2020[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2021 <- (tb_2021[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2022 <- (tb_2022[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])
cp_2023 <- (tb_2023[, c("Date", "X13.WEEKS.COUPON.EQUIVALENT")])

# set the date type in ascending order
cp_2014 <- cp_2014[order(cp_2014$Date), ]
cp_2015 <- cp_2015[order(cp_2015$Date), ]
cp_2016 <- cp_2016[order(cp_2016$Date), ]
```

```r
cp_2017 <- cp_2017[order(cp_2017$Date), ]
cp_2018 <- cp_2018[order(cp_2018$Date), ]
cp_2019 <- cp_2019[order(cp_2019$Date), ]
cp_2020 <- cp_2020[order(cp_2020$Date), ]
cp_2021 <- cp_2021[order(cp_2021$Date), ]
cp_2022 <- cp_2022[order(cp_2022$Date), ]
cp_2023 <- cp_2023[order(cp_2023$Date), ]

# list them all
cp_list <- list(
  cp_2014 = cp_2014, cp_2015 = cp_2015, cp_2016 = cp_2016,
  cp_2017 = cp_2017, cp_2018 = cp_2018, cp_2019 = cp_2019,
  cp_2020 = cp_2020, cp_2021 = cp_2021, cp_2022 = cp_2022,
  cp_2023 = cp_2023)

library(dplyr)
df_merge <- lapply(cp_list, function(df) {
  # sort by date
  df <- df[order(as.Date(df$Date, format = "%m/%d/%Y")), ]
  # rename columns
  colnames(df) <- c("date", "coupon")
  # convert date type
  df$date <- as.Date(df$date, format = "%m/%d/%Y")
  return(df)})

names(df_merge) <- names(cp_list)
combined_df <- bind_rows(df_merge)

tmp <- inner_join(sp500_df, gold_df, by = "date")
df <- inner_join(tmp, combined_df, by = "date")
colnames(df) <- c("date", "sp500", "gold", "coupon")

df_filtered <- df %>%
  filter(date >= "2014-01-01" & date <= "2020-01-01")

gold <- df_filtered$gold
gold_ts <- ts(gold,
  start = c(2014, 1),
  frequency = 249)

gold.dat <- df$gold
gold.dat_ts <- ts(
  gold.dat,
  start = c(2014, 1),
  frequency = 249)

gold_model <- auto.arima(
  gold_ts,
  max.p = 5, max.q = 5,
  max.order = 10, stationary = F, seasonal = F,
  trace = F, stepwise = F, approximation = F)

fitted1 <- gold_ts - gold_model$resid
```

```r
range_df <- data.frame(
  Date = df_filtered$date,
  Gold = gold_ts, Fitted = fitted1)
par(mfrow=c(1,2))

cor_value <- round(cor(gold, fitted1, use = "complete.obs"), 3)
plot(gold, fitted1,
  xlab = "Acutal",
  ylab = "Predicted", col = "black", pch = 20,
  main = "Plot of Predicted vs. Actual",
  cex.main = 0.8,
  cex.lab = 0.7,
  cex.axis = 0.7)
abline(0, 1, col = "red")


plot(range_df$Date, range_df$Gold,
  type = "l", col = "black",
  main = "Plot of Fitted vs. Actual (ARIMA (5,1,5))",
  xlab = "Date", ylab = "Gold Price",
  cex.main = 0.8,
  cex.lab = 0.7,
  cex.axis = 0.7)
lines(range_df$Date, range_df$Fitted, col = "red")
tsdisplay(gold_model$resid,
  points = F,
  main = "Residuals of ARIMA Model (5,1,5)",
  lag.max = 20,
  cex.main = 0.8,
  cex.lab = 0.7,
  cex.axis = 0.7)
qqnorm(gold_model$resid, main = "Normal Q-Q plot (ARIMA (5,1,5))",
       cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
qqline(gold_model$resid, col = "red")
forecast_values <- forecast(gold_model, h = 1498)
forecast_dates <- seq(
  from = max(df_filtered$date) + 1, by = "day",
  length.out = 1498)
actual_data <- gold_df[gold_df$date >= as.Date("2014-01-01") &
                         gold_df$date <= as.Date("2023-12-31"), ]
combined_x_range <- range(actual_data$date, forecast_dates, na.rm = TRUE)
combined_y_range <- range(actual_data$close, forecast_values$mean,
  forecast_values$lower,
  forecast_values$upper,
  na.rm = TRUE)
plot(actual_data$date, actual_data$close,
  type = "l", lwd = 1, col = "black",
  xlab = "Date", ylab = "Gold Price",
  main = "Forecast vs. Actual Gold Prices",
  xlim = combined_x_range, ylim = combined_y_range,
  cex.main = 0.8,
  cex.lab = 0.7,
  cex.axis = 0.7)
```

```r
lines(forecast_dates, forecast_values$mean, col = "red", lwd = 1)
lines(forecast_dates, forecast_values$lower[, 2],
  col = "blue", lty = 2) # 95% lower CI
lines(forecast_dates, forecast_values$upper[, 2],
  col = "blue", lty = 2) # 95% upper CI
abline(v = as.Date("2020-01-01"), col = "grey", lwd = 1, lty = 2)
legend("topleft",
  legend = c("Actual", "Forecast", "95% Confidence Interval"),
  col = c("black", "red", "blue"),
  lty = c(1, 1, 2), lwd = c(1, 1, 1), cex=0.75)
lag_selection <- VARselect(gold_diff, lag.max=10, type="both")
optimal_lag <- lag_selection$selection["AIC(n)"]

var_model <- VAR(gold_diff, p=optimal_lag, type="both")

serial_test <- serial.test(var_model, lags.pt=optimal_lag, type="PT.asymptotic")
forecast_length <- 120
var_forecast <- forecast(var_model, h=forecast_length)
res <- residuals(var_model)
plot(var_forecast)


gold_data_var <- data %>%
  filter(date <= as.Date("2019-12-31")) %>%
  dplyr::select(date, gold)

future_data_var <- data %>%
  filter(date > as.Date("2019-12-31")) %>%
  dplyr::select(date, gold)

additional_data <- future_data_var %>%
  slice(1:forecast_length)

extended_gold_data <- bind_rows(gold_data_var, additional_data)

extended_gold_data <- extended_gold_data %>%
  mutate(
    lower_ci = as.numeric(NA),
    upper_ci = as.numeric(NA),
    Type = "Actual"
  )

last_value <- tail(data_ts[, "gold"], 1)
undiff_forecast <- rep(last_value, length(var_forecast$forecast[[1]]$mean))
undiff_forecast <- undiff_forecast + cumsum(var_forecast$forecast[[1]]$mean)
undiff_forecast[1] <- last_value + var_forecast$forecast[[1]]$mean[1]
forecasted_gold_var <- undiff_forecast

last_date <- max(gold_data_var$date)
forecast_dates <- seq.Date(from = last_date + 1,
                           by = "day", length.out = forecast_length)

undiff_lower_ci <- rep(last_value, length(var_forecast$forecast[[1]]$lower))
```

```
undiff_lower_ci <- undiff_lower_ci + cumsum(var_forecast$forecast[[1]]$lower)
undiff_lower_ci[1] <- last_value + var_forecast$forecast[[1]]$lower[1]

undiff_upper_ci <- rep(last_value, length(var_forecast$forecast[[1]]$upper))
undiff_upper_ci <- undiff_upper_ci + cumsum(var_forecast$forecast[[1]]$upper)
undiff_upper_ci[1] <- last_value + var_forecast$forecast[[1]]$upper[1]

sliced_lower_ci <- undiff_lower_ci[1:forecast_length]
sliced_upper_ci <- undiff_upper_ci[1:forecast_length]

forecast_data_var <- data.frame(
  date = forecast_dates,
  gold = forecasted_gold_var,
  lower_ci = sliced_lower_ci,
  upper_ci = sliced_upper_ci,
  Type = "Forecast"
)

gold_total_var <- rbind(extended_gold_data, forecast_data_var)

future_data_var_1 <- data %>%
  filter(date > as.Date("2019-12-31")) %>%
  dplyr::select(date, gold)

additional_data_1 <- future_data_var_1 %>%
  slice(1:forecast_length)

additional_data_1 <- additional_data_1 %>%
  mutate(
    lower_ci = as.numeric(NA),
    upper_ci = as.numeric(NA),
    Type = "Actual"
  )

gold_total_var_1 <- rbind(additional_data_1, forecast_data_var)

library(patchwork)

p1 <- ggplot(gold_total_var, aes(x = date, y = gold)) +
  geom_line(aes(color = Type), size = 0.5) +
  geom_ribbon(data = filter(forecast_data_var, Type == "Forecast"),
              aes(ymin = lower_ci, ymax = upper_ci), fill = "red", alpha = 0.2) +
  scale_color_manual(values = c("Actual" = "blue", "Forecast" = "red")) +
  labs(title = "Gold: Actual vs Forecast",
       x = "Date",
       y = "Gold",
       color = "Type") +
  theme_minimal()

p2 <- ggplot(gold_total_var_1, aes(x = date, y = gold)) +
  geom_line(aes(color = Type), size = 0.5) +
  geom_ribbon(data = filter(forecast_data_var, Type == "Forecast"),
              aes(ymin = lower_ci, ymax = upper_ci), fill = "red", alpha = 0.2) +
```

```
  scale_color_manual(values = c("Actual" = "blue", "Forecast" = "red")) +
  labs(title = "Gold: Actual vs Forecast",
       x = "Date",
       y = "Gold",
       color = "Type") +
  theme_minimal()

p1 + p2

residuals_var <- residuals(var_model)

residuals_df <- as.data.frame(residuals_var)

time_index <- as.numeric(rownames(residuals_df))

par(mfrow=c(1,2))
plot(time_index, residuals_df$gold, type = "l", col = "black",
     main = "Residuals of Gold",
     xlab = "Time", ylab = "Residuals", lwd = 2,
     cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
abline(0, 0, col = "red")

qqnorm(residuals_df$gold, main = "Normal Q-Q plot (VAR(10))",
       cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
qqline(gold_model$resid, col = "red")
filtered_data <- data %>%
  filter(date <= as.Date("2019-12-31")) %>%
  dplyr::select(sp500, coupon)

gold_data <- data %>%
  filter(date <= as.Date("2019-12-31")) %>%
  dplyr::select(date, gold)

gold_ts <- ts(gold_data$gold, frequency = 250)

regressor <- as.matrix(filtered_data)

fit <- auto.arima(gold_ts, max.p = 5, max.q = 5, max.order = 10,
                  stationary = F, seasonal = F, trace = F, stepwise = F,
                  approximation = F, xreg = regressor)


filtered_data_future <- data %>%
  filter(date > as.Date("2019-12-31")) %>%
  dplyr::select(date, sp500, coupon)
future_regressor <- as.matrix(filtered_data_future[, c("sp500", "coupon")])

actual_gold_price <- data %>%
  filter(date > as.Date("2019-12-31")) %>%
  dplyr::select(date, gold)

# Forecast gold prices using the model
future_gold_price <- forecast(fit, xreg = future_regressor)
```

```r
actual_gold_ts <- ts(actual_gold_price$gold, start = c(2020, 1), frequency = 250)

gold_data_arima <- data %>%
  filter(date <= as.Date("2019-12-31")) %>%
  dplyr::select(date, gold) %>%
  mutate(
    Lower = as.numeric(NA),
    Upper = as.numeric(NA),
    Type = "Historical Actual"
  )

forecast_length <- length(future_gold_price$mean)
forecast_dates <- seq(from = as.Date("2020-01-01"),
                      length.out = forecast_length, by = "day")

forecast_df <- data.frame(
  date = forecast_dates,
  gold = future_gold_price$mean,
  Lower = future_gold_price$lower[,1],
  Upper = future_gold_price$upper[,1],
  Type = "Forecast Prediction"
)

actual_forecast_period <- data %>%
  filter(date > as.Date("2019-12-31") & date <= max(forecast_dates)) %>%
  dplyr::select(date, gold) %>%
  mutate(
    Lower = as.numeric(NA),
    Upper = as.numeric(NA),
    Type = "Forecast Actual"
  )

combined_data <- bind_rows(gold_data_arima, actual_forecast_period, forecast_df)

ggplot(combined_data, aes(x = date, y = gold, group = Type, color = Type)) +
  geom_line(size = 0.5) +
  geom_ribbon(data = filter(combined_data, Type == "Forecast Prediction"),
              aes(ymin = Lower, ymax = Upper), fill = "red", alpha = 0.2) +
  scale_color_manual(values = c("Historical Actual" = "black",
                                "Forecast Actual" = "blue",
                                "Forecast Prediction" = "red")) +
  labs(title = "Gold: Historical, Actual, and Forecast",
       x = "Date",
       y = "Gold",
       color = "Data Type") +
  theme_minimal()

par(mfrow=c(1,2))
plot(fit$residuals, main = "Residuals of ARIMA model",
     cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)

qqnorm(fit$residuals, pch = 1, frame = FALSE,
       cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
```

```r
qqline(fit$residuals, col = "steelblue", lwd = 2)
par(mfrow=c(1,2))
acf(fit$residuals, main = "ACF for Residuals",
    cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
pacf(fit$residuals, main = "PACF for Residuals",
     cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
library(readxl)
df <- read_excel("data.xlsx")
df_filtered <- df %>%
  filter(date >= "2020-01-01" & date <= "2023-12-31")

data1 <- df_filtered[,c(1,3)]
gold <- data1$gold
gold_ts <- ts(gold,frequency = 252, start = c(2020,1))

fit <- auto.arima(gold_ts, max.p = 5, max.q = 5, max.order = 10,
                  stationary = F, seasonal = F, trace = F, stepwise = F, approximation = F)

fitted2 = gold-fit$resid
par(mfrow=c(1,2))
cor_value <- round(cor(gold, fitted2, use = "complete.obs"), 3)
plot(gold, fitted2, xlab="Actual",
     ylab="Predicted", col="black", pch=20,
     main = "Plot of Predicted vs. Actual",
     cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
abline(0, 1, col="red")
plot(gold_ts, type ='l', ylab = "Gold Price",
     main = "Plot of Fitted vs. Actual (ARIMA (2,1,2))")
lines(fitted2,col="red")
tsdisplay(fit$resid, points = F,
         main="Residuals of ARIMA Model (2,1,2)")

qqnorm(fit$resid, cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
qqline(fit$resid, col = "red")
forecast_values <- forecast(fit, h = 72)
forecast_dates <- seq(from = as.Date("2024-01-01") + 1,
                      by = "day", length.out = 72)

actual_data <- gold_df[gold_df$date >= as.Date("2020-01-01") &
                         gold_df$date <= as.Date("2024-04-16"),]

combined_y_range <- range(actual_data$close, forecast_values$mean,
                          forecast_values$lower[,2], forecast_values$upper[,2],
                          na.rm = TRUE)

plot(actual_data$date, actual_data$close, type = "l", lwd = 1, col = "black",
     ylim = combined_y_range, xlab = "Date", ylab = "Gold Price",
     main = "Forecast vs. Actual Gold Price",
     cex.main = 0.8, cex.lab = 0.7, cex.axis = 0.7)
lines(forecast_dates, forecast_values$mean, col = "red", lwd = 1)
abline(v = as.Date("2024-01-01"), col = "grey", lwd = 1, lty = 2)
lines(forecast_dates, forecast_values$lower[, 2], col = "blue", lty = 2)  # 95% lower CI
lines(forecast_dates, forecast_values$upper[, 2], col = "blue", lty = 2)  # 95% upper CI
```

```r
legend("topleft",
       legend = c("Actual", "Forecast", "95% Confidence Interval"),
       col = c("black", "red", "blue"),
       lty = c(1, 1, 2), lwd = c(1, 1, 1))
data_origin <- read_xlsx("data.xlsx")
data_origin <- subset(data_origin, year %in% c(2020, 2021, 2022, 2023))
data_origin$date <- as.Date(data_origin$date)

gold <- data_origin[, c(1,3)]

data <- ts(data_origin[, c("sp500", "gold", "coupon")],
           frequency = 250, start = c(2020, 1, 2))

data_diff <- diff(data, differences = 2)
var_lag <- VARselect(data_diff, lag.max = 20, type = "both", season = NULL)
var_model <- VAR(data_diff, p = 9, type = "both")

forecast_length <- 70
forecast_results <- forecast(var_model, h = 70)
plot(forecast_results)
last_value <- tail(data[, "gold"],1)
last_date <- max(data_origin$date)
forecast_date <- seq.Date(from = last_date + 1, by = "day", length.out = forecast_length)

undiff_forecast <- rep(last_value, length(forecast_results$forecast[[2]]$mean))
undiff_forecast <- undiff_forecast + cumsum(forecast_results$forecast[[2]]$mean)
undiff_forecast[1] <- last_value + forecast_results$forecast[[2]]$mean[2]

undiff_lower_ci <- rep(last_value, length(forecast_results$forecast[[2]]$lower))
undiff_lower_ci <- undiff_lower_ci + cumsum(forecast_results$forecast[[2]]$lower)
undiff_lower_ci[1] <- last_value + forecast_results$forecast[[2]]$lower[[2]]

undiff_upper_ci <- rep(last_value, length(forecast_results$forecast[[2]]$upper))
undiff_upper_ci <- undiff_upper_ci + cumsum(forecast_results$forecast[[2]]$upper)
undiff_upper_ci[1] <- last_value + forecast_results$forecast[[2]]$upper[[2]]

sliced_lower_ci <- undiff_lower_ci[1:forecast_length]
sliced_upper_ci <- undiff_upper_ci[1:forecast_length]

forecast_data_var <- data.frame(
  date = forecast_date,
  gold = undiff_forecast,
  lower_ci = sliced_lower_ci,
  upper_ci = sliced_upper_ci,
  Type = "Forecast"
)

gold_2024 <- read.csv("2024gold.csv")
gold_2024 <- gold_2024[, 2:3]
colnames(gold_2024)[2] <- "gold"
gold_2024$date <- as.Date(gold_2024$date)

gold_before <- data_origin[, c(1,3)]
```

```r
gold_all <- bind_rows(gold_before, gold_2024)
gold_all <- gold_all %>%
  mutate(
    lower_ci = as.numeric(NA),
    upper_ci = as.numeric(NA),
    Type = "Actual"
  )

gold_total_var <- rbind(gold_all, forecast_data_var)

p1 <- ggplot(gold_total_var, aes(x = date, y = gold)) +
  geom_line(aes(color = Type), linewidth = 0.5) +
  geom_ribbon(data = filter(gold_total_var, Type == "Forecast"),
              aes(ymin = lower_ci, ymax = upper_ci), fill = "red", alpha = 0.2) +
  scale_color_manual(values = c("Actual" = "blue", "Forecast" = "red")) +
  labs(title = "Gold Prices: Actual vs Forecast",
       x = "Date",
       y = "Gold Price",
       color = "Type") +
  theme_minimal()

gold_2024_real <- gold_2024 %>%
  mutate(
    lower_ci = as.numeric(NA),
    upper_ci = as.numeric(NA),
    Type = "Actual"
  )
gold_2024_real <- gold_2024_real[1:70,]
gold_pred <- rbind(forecast_data_var, gold_2024_real)

p2 <- ggplot(gold_pred, aes(x = date, y = gold)) +
  geom_line(aes(color = Type), linewidth = 0.5) +
  geom_ribbon(data = filter(gold_pred, Type == "Forecast"),
              aes(ymin = lower_ci, ymax = upper_ci), fill = "red", alpha = 0.2) +
  scale_color_manual(values = c("Actual" = "blue", "Forecast" = "red")) +
  labs(title = "Gold Prices: Actual vs Forecast",
       x = "Date",
       y = "Gold Price",
       color = "Type") +
  theme_minimal()

p1+p2
residuals <- residuals(var_model)

plot(residuals[,2], type = "l", ylab = "Residuals", xlab = " ",
     main = "Residuals of Gold")
abline(h = 0, col = "red", lwd = 2)
qqnorm(residuals[,2])
qqline(residuals[,2], col = "red")
data_diff1 <- diff(data, differences = 1)
gold_2024 <- read.csv("2024gold.csv")
gold_2024 <- gold_2024[, -1]
gold_2024$date <- as.Date(gold_2024$date)
```

```r
c <- getSymbols("SPY",src = "yahoo", from = "2024-01-02",
                to = "2024-04-15", auto.assign = FALSE)
sp500_2024 <- na.omit(Cl(c))
colnames(sp500_2024) <- "sp500"
sp500_2024 <- fortify.zoo(sp500_2024)
colnames(sp500_2024)[1] <- "date"

coupon_2024 <- read_xlsx("coupon_2024.xlsx")

future_regressor <- inner_join(sp500_2024, coupon_2024, by = "date")
future_regressor <- ts(future_regressor)
future_regressor <- future_regressor[,2:3]
gold_ts <- data[, "gold", drop = FALSE]

regressor <- data[,c(1,3)]
colnames(future_regressor)[1] <- "gold"

fit <- auto.arima(gold_ts, xreg = regressor, max.p = 5, max.q = 5,
                  max.order = 10, stationary = F, seasonal = F, trace = F,
                  stepwise = F, approximation = F)
future_gold <- forecast(fit, xreg = future_regressor)
forcast_length <- length(future_gold$mean)
forecast_dates <- seq(from = as.Date("2024-01-01"),
                      length.out = forecast_length, by = "day")

forecast_df <- data.frame(
  date = forecast_dates,
  gold = future_gold$mean[1:70],
  lower_ci = future_gold$lower[1:70],
  upper_ci = future_gold$upper[1:70],
  Type = "Forecast"
)

combined_data <- bind_rows(forecast_df, gold_all)

ggplot(combined_data, aes(x = date, y = gold, group = Type, color = Type)) +
  geom_line(aes(color = Type), linewidth = 0.5) +
  geom_ribbon(data = filter(combined_data, Type == "Forecast"),
              aes(ymin = lower_ci, ymax = upper_ci), fill = "red", alpha = 0.2) +
  scale_color_manual(values = c("Actual" = "black",
                                "Forecast" = "red")) +
  labs(title = "Gold Price: Actual and Forecast",
       x = "Date",
       y = "Price",
       color = "Data Type") +
  theme_minimal()
plot(fit$residuals)
abline(h = 0, col = "red")
qqnorm(fit$residuals)
qqline(fit$residuals)
```