Gold Price Forecasting Using Time Series Analysis and Machine Learning

Objective

To analyze and forecast gold prices using historical data by applying time series analysis (ARIMA) and machine learning models (Ridge and Lasso) to identify which approach predicts gold prices more accurately for financial planning and investment analysis.

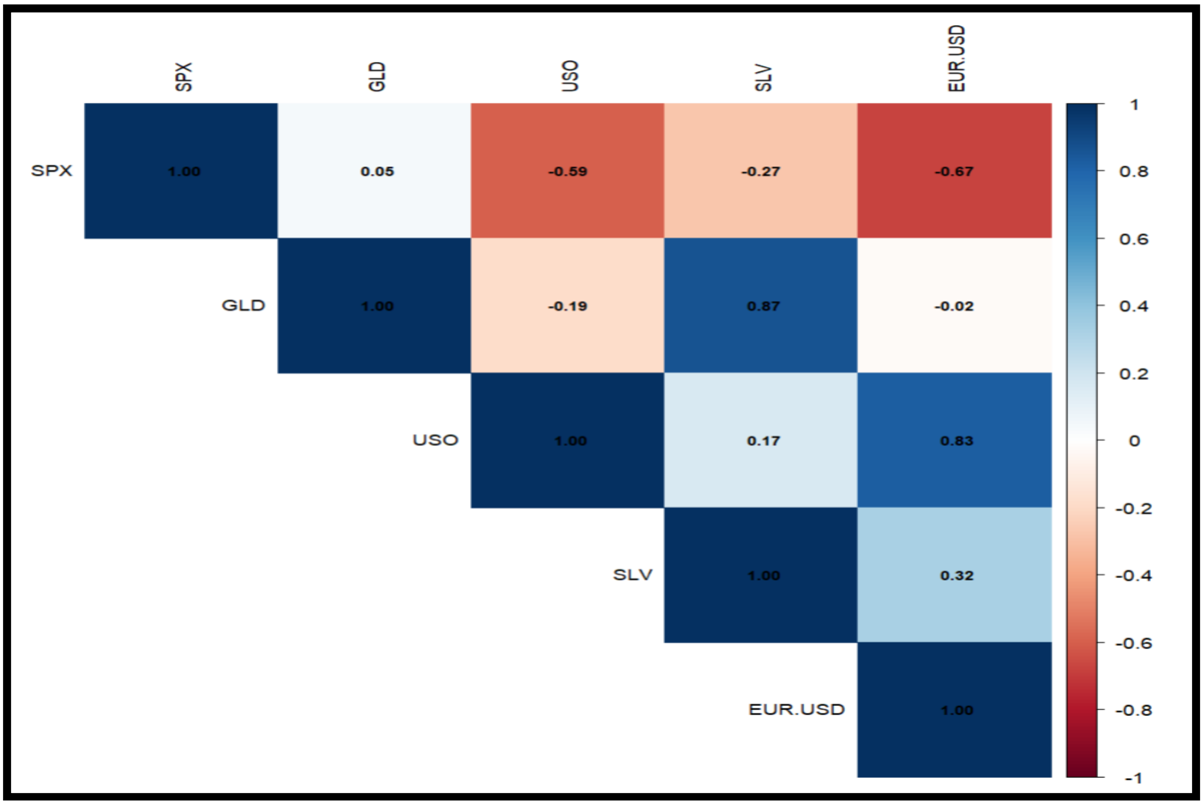
Data Description

- Dataset: Daily gold price dataset with associated economic indicators.
- Period: 2008 to 2020.
- Columns:
    - Date: Date of observation
    - GLD: Gold ETF price (target variable)
    - SPX: S&P 500 index
    - USO: Oil ETF price
    - SLV: Silver ETF price
    - EUR.USD: Euro to USD exchange rate

Data Preprocessing & EDA

- Checked and found no missing values.
- Converted Date to datetime and set as index.
- Analyzed **correlations** using a heatmap:

```
> correlation_matrix <- cor(gold_num)
> correlation_matrix
                SPX         GLD        USO        SLV     EUR.USD
SPX      1.00000000  0.04934504 -0.5915726 -0.2740547 -0.67201742
GLD      0.04934504  1.00000000 -0.1863602  0.8666319 -0.02437547
USO     -0.59157260 -0.18636016  1.0000000  0.1675471  0.82931745
SLV     -0.27405473  0.86663188  0.1675471  1.0000000  0.32163127
EUR.USD -0.67201742 -0.02437547  0.8293175  0.3216313  1.00000000
```



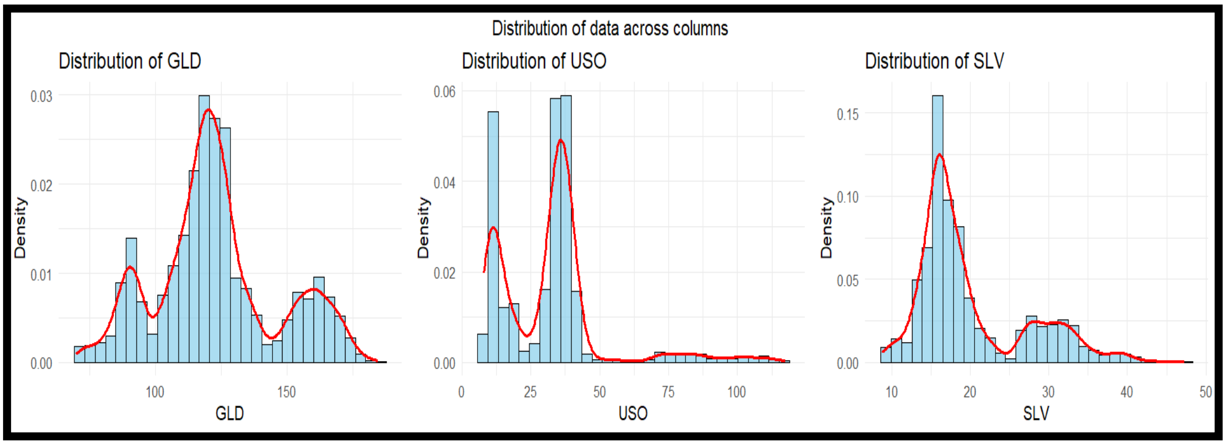
- Strong positive correlation between GLD and SLV (**0.87**) as well as between USO and EUR.USD (**0.83**).
- Strong negative correlations between SPX with USO (**-0.59**) and EUR.USD (**-0.67**).
- **Dropped SPX and EUR.USD** using the **principle of Parsimony** due to low correlation with GLD.

- Checked multicollinearity using **VIF (all < 5) ➔** No multicollinearity present.

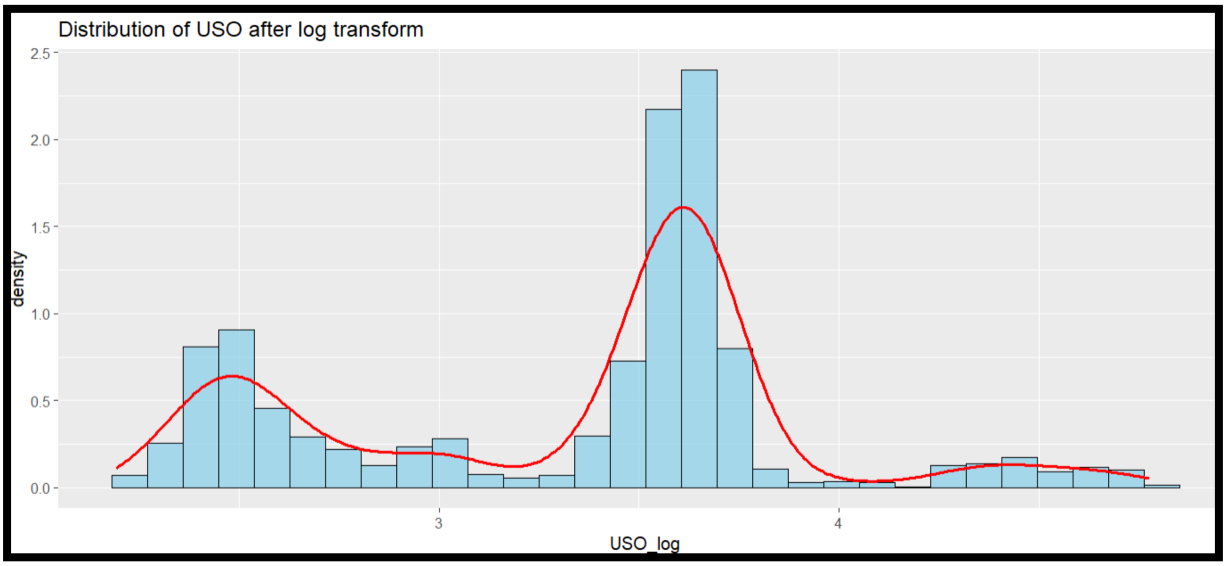
```
> # Checking Multicollinearity
> model <- lm(GLD ~ SLV + SPX + USO + EUR.USD, data=gold_data)
> vif(model)
     SLV      SPX      USO  EUR.USD
1.167741 1.854727 3.353192 4.128340
```

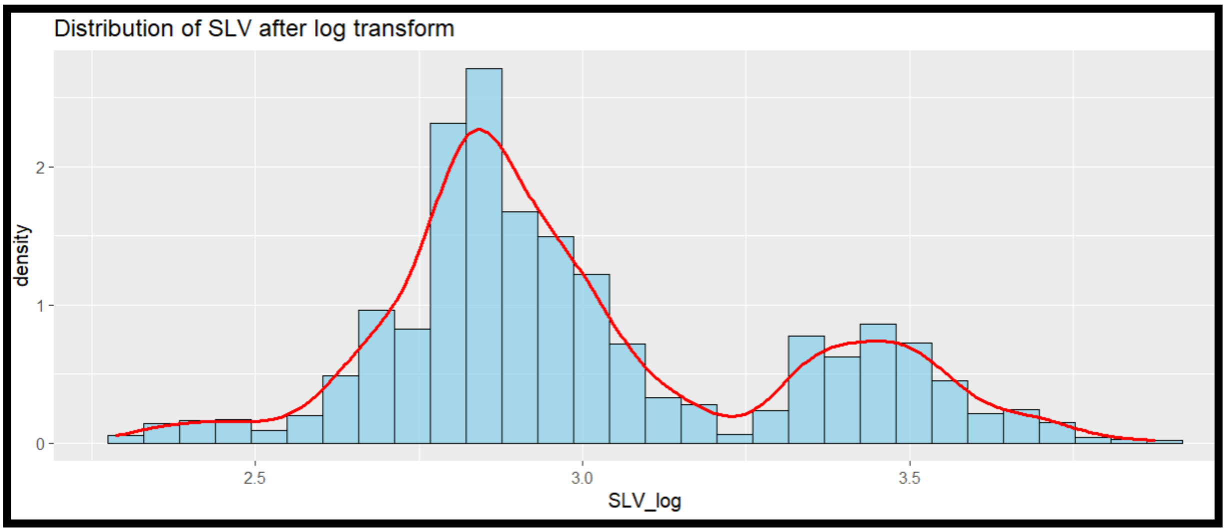
- Distribution Analysis:

```
> # to check the skewness of data
> skewness_values <- sapply(gold_data_new[temp_cols], skewness, na.rm=TRUE)
> skewness_values
      GLD       USO       SLV
0.3337007 1.6971054 1.1521300
```



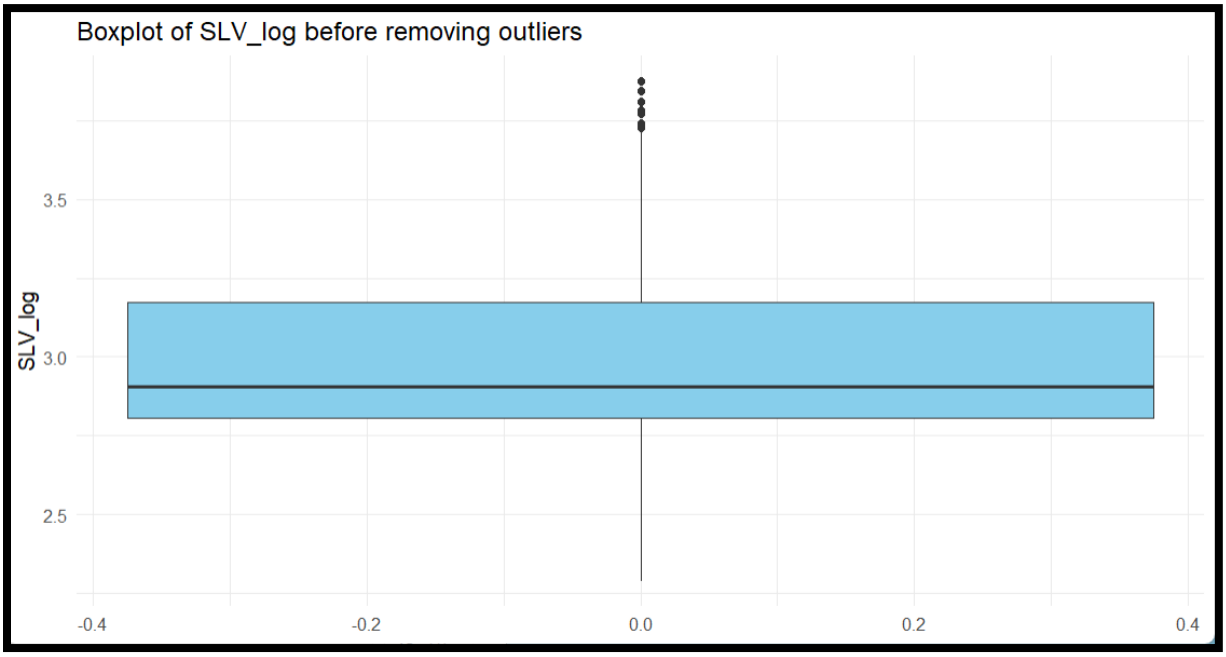
- USO and SLV were **positively skewed (skewness > 1)**.
- Applied **log transformation** to reduce skewness.

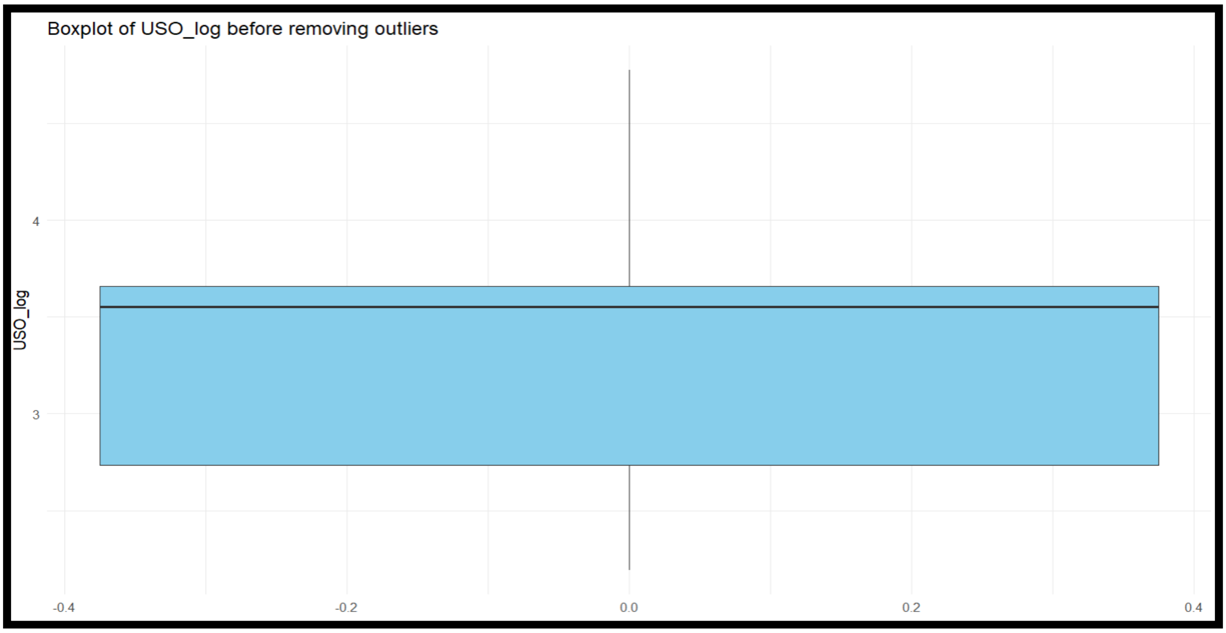


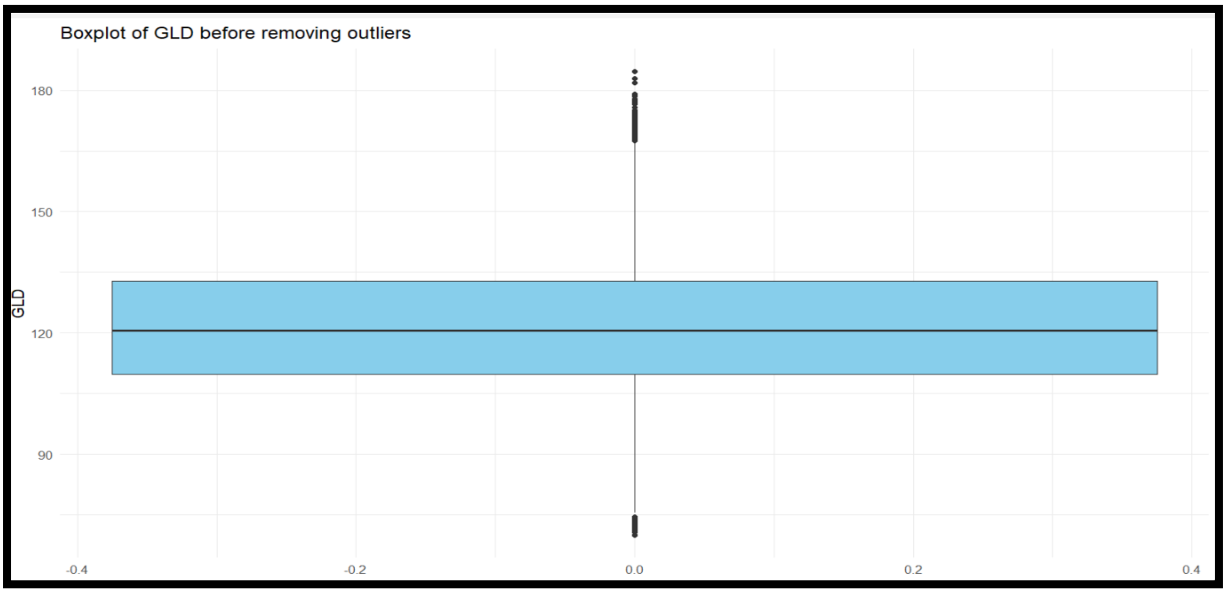


- Outlier Handling:

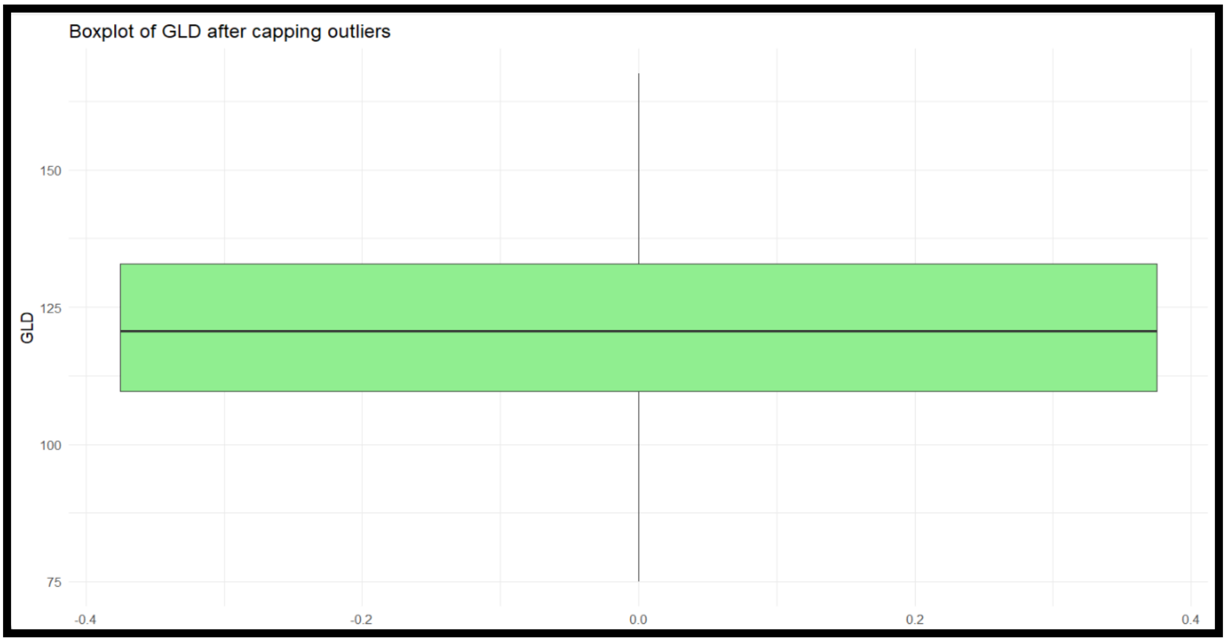
```
> outliers_list <- lapply(gold_data_new[cols_to_check], detect_outliers)
> outliers_count_before <- sapply(outliers_list, length)
> print(outliers_count_before)
    GLD USO_log SLV_log
    115       0      20
```

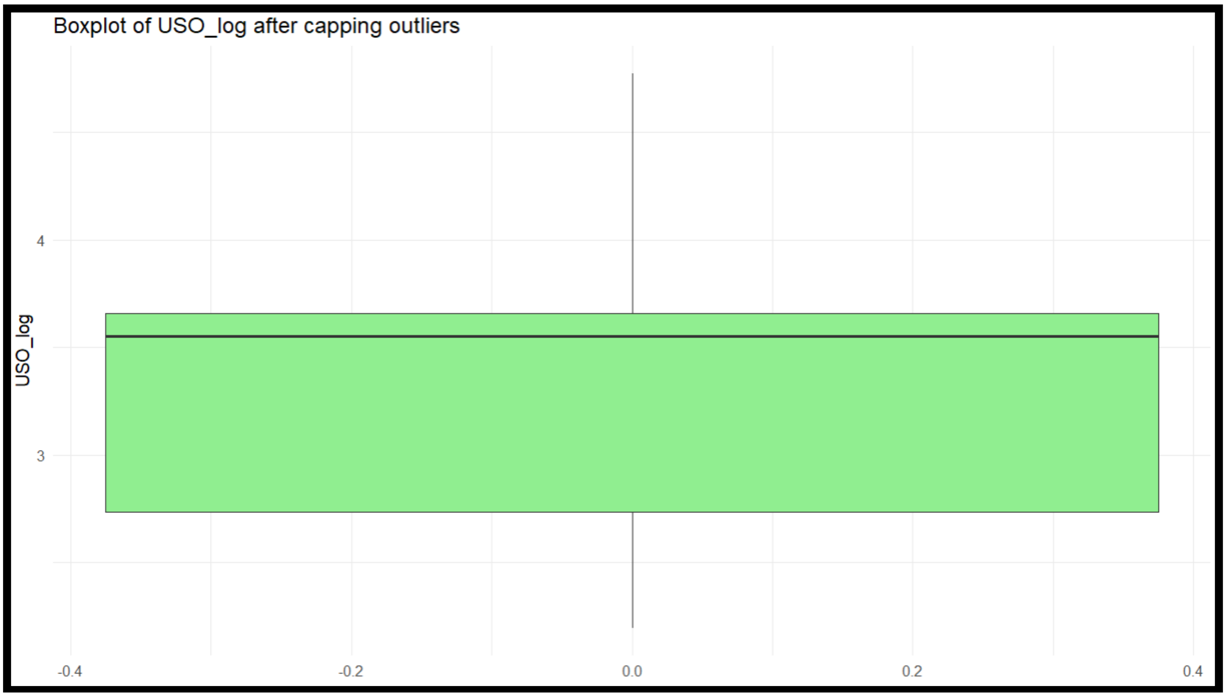


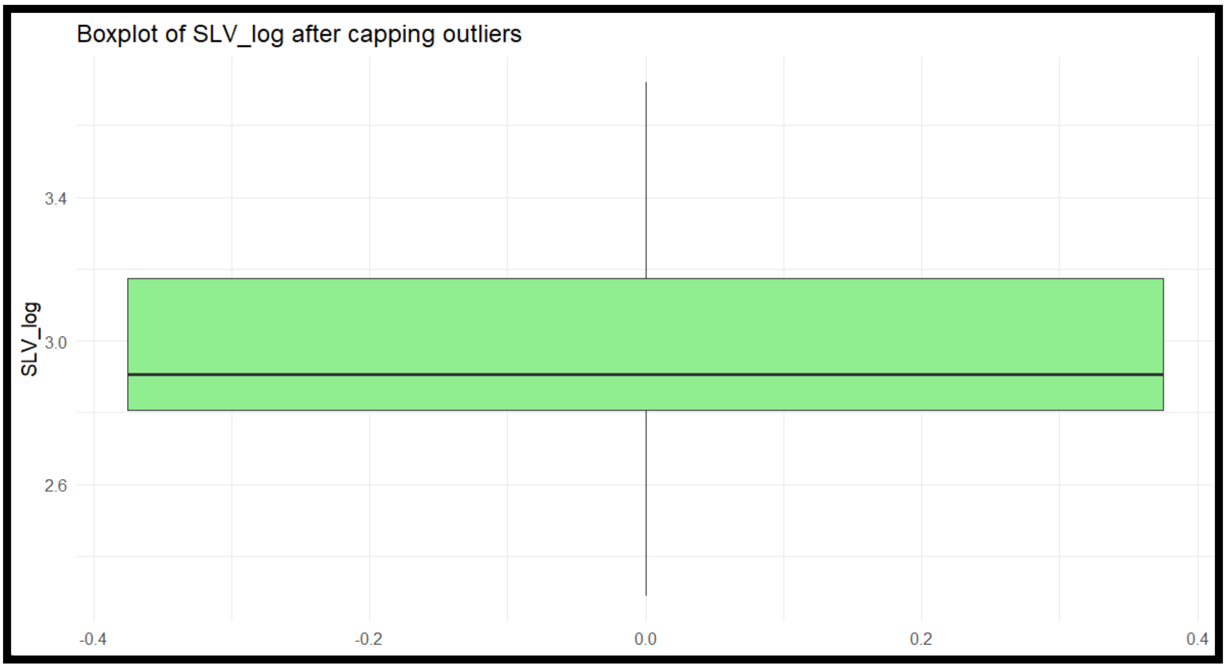




- Used **IQR capping** to reduce extreme outlier influence.


Boxplot of GLD after capping outliers
GLD
150
125
100
75
-0.4
-0.2
0.0
0.2
0.4

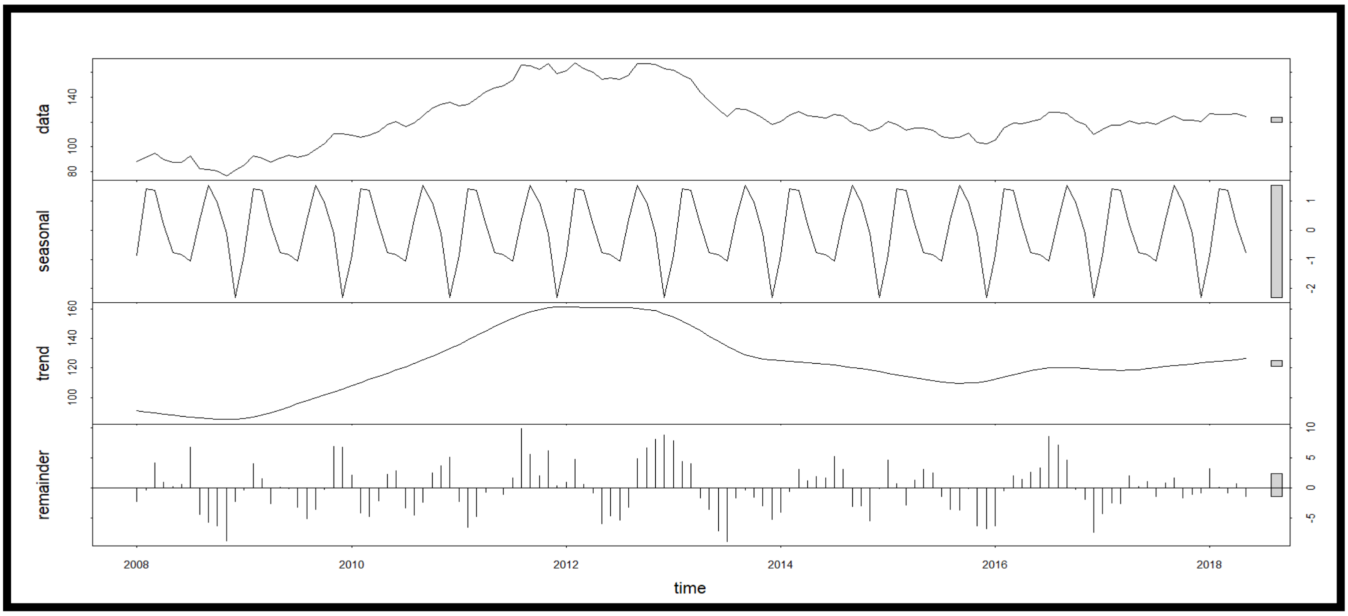

Boxplot of USO_log after capping outliers
USO_log
4
3
-0.4
-0.2
0.0
0.2
0.4


Boxplot of SLV_log after capping outliers
SLV_log
3.4
3.0
2.6
-0.4
-0.2
0.0
0.2
0.4

- Converted to monthly aggregation for time series modeling.

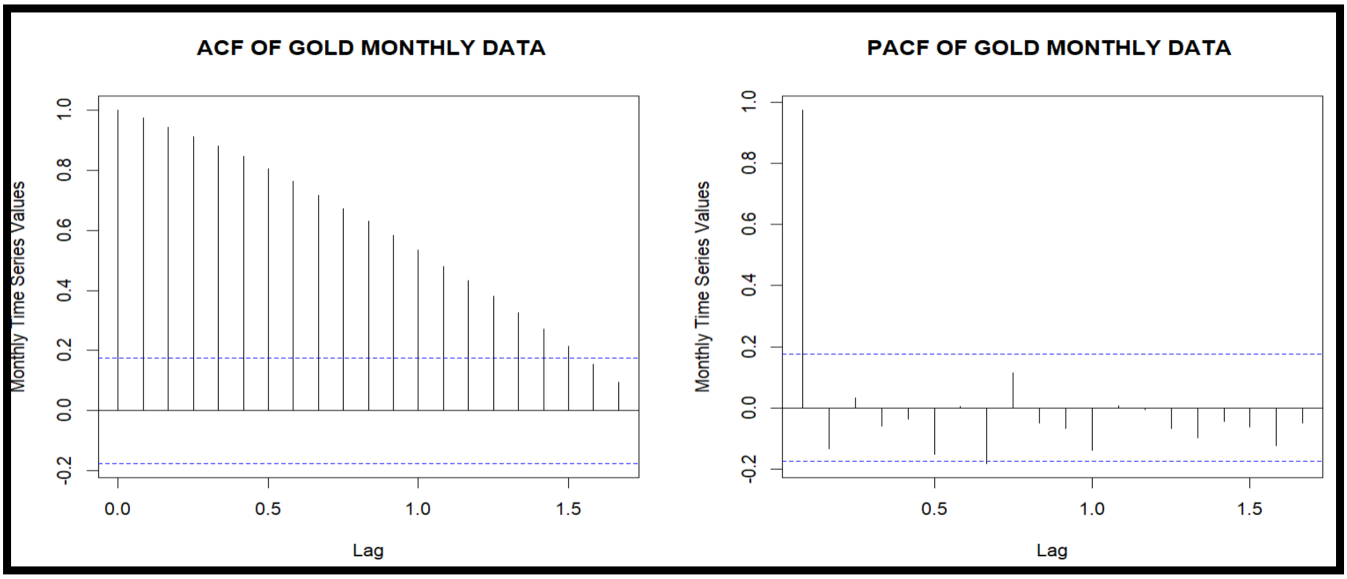
Time Series Analysis (ARIMA)

- STL Decomposition



  - Clear **upward trend (2008–2012)**, moderation, and mild recovery post-2016.
  - **Annual seasonality observed**, making ARIMA appropriate.

- Stationarity Checks



  - **ACF slow decay, PACF sharp cutoff ➔ non-stationary.**

```
> adf.test(gold_monthly_ts)

        Augmented Dickey-Fuller Test

data:  gold_monthly_ts
Dickey-Fuller = -1.5654, Lag order = 4, p-value = 0.7571
alternative hypothesis: stationary
```

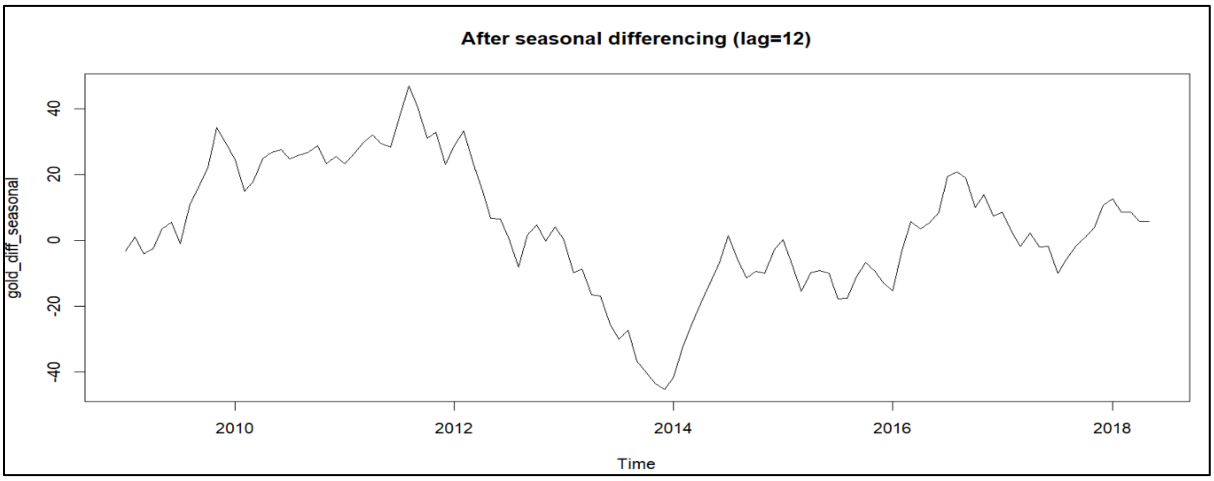
```
> PP.test(gold_monthly_ts)

        Phillips-Perron Unit Root Test

data:  gold_monthly_ts
Dickey-Fuller = -1.4544, Truncation lag parameter = 4, p-value = 0.8033
```

H_0: The Time series has a unit root (i.e. non-stationary)

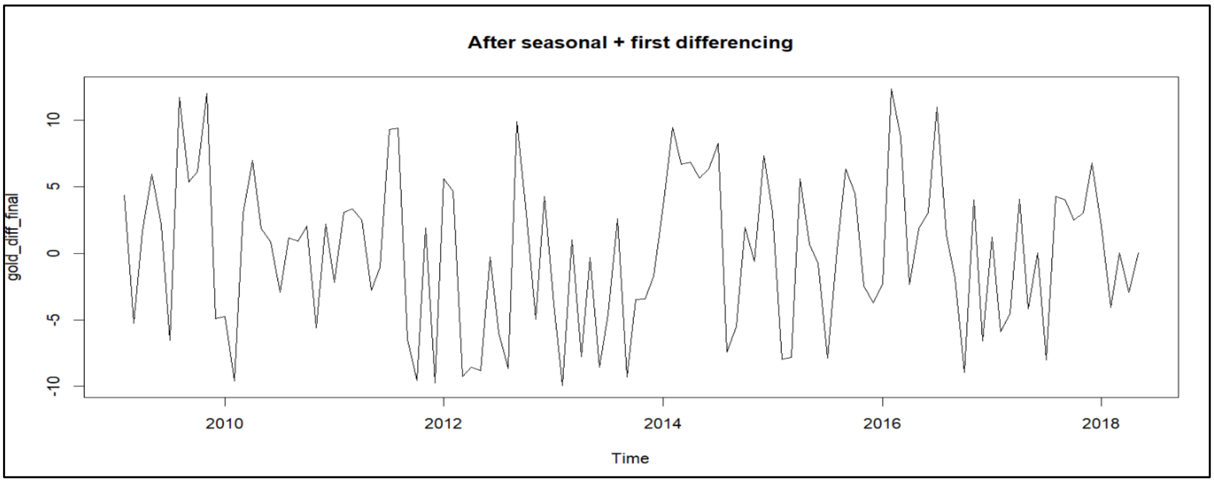
H_1: The Time series has a non-unit root (i.e. stationary)

- **ADF & PP tests**: p-value > 0.05 ➔ non-stationary.

```
gold_diff_seasonal <- diff(gold_monthly_ts, lag=12)
plot(gold_diff_seasonal, main="After seasonal differencing (lag=12)")
```




```
# First difference to remove trend
gold_diff_final <- diff(gold_diff_seasonal, differences=1)
plot(gold_diff_final, main="After seasonal + first differencing")
```



- Applied **seasonal (lag 12) + first differencing (lag 1)** to achieve stationarity.

```
> adf.test(na.omit(gold_diff_final))

        Augmented Dickey-Fuller Test

data:  na.omit(gold_diff_final)
Dickey-Fuller = -3.7483, Lag order = 4, p-value = 0.02397
alternative hypothesis: stationary
```

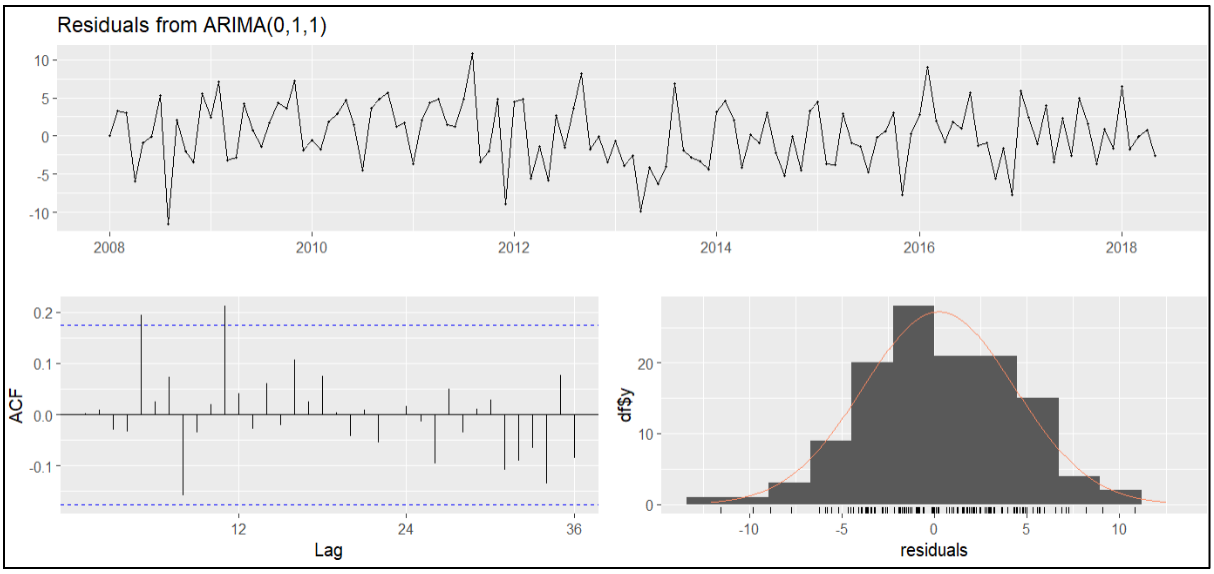
```
> PP.test(gold_diff_final)

        Phillips-Perron Unit Root Test

data:  gold_diff_final
Dickey-Fuller = -8.889, Truncation lag parameter = 4, p-value = 0.01
```

- Post differencing: p-value < 0.05 ➔ stationary.

- ARIMA Modeling

```
> fit_model <- auto.arima(gold_monthly_ts, stepwise=FALSE, approximation=FALSE)
> summary(fit_model)
Series: gold_monthly_ts
ARIMA(0,1,1)

Coefficients:
         ma1
      0.2368
s.e.  0.0862

sigma^2 = 16.96:  log likelihood = -350.99
AIC=705.98   AICc=706.08   BIC=711.62

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set 0.2289691 4.085259 3.349196 0.1718651 2.808417 0.2136145 0.001599452
```

- Used **auto.arima ➔ ARIMA(0,1,1)** selected.
- Diagnostic checks:



- ○ Residuals ~ white noise (no autocorrelation, normal distribution).

```
> checkresiduals(fit_model)

        Ljung-Box test

data:  Residuals from ARIMA(0,1,1)
Q* = 20.255, df = 23, p-value = 0.6265

Model df: 1.   Total lags used: 24
```

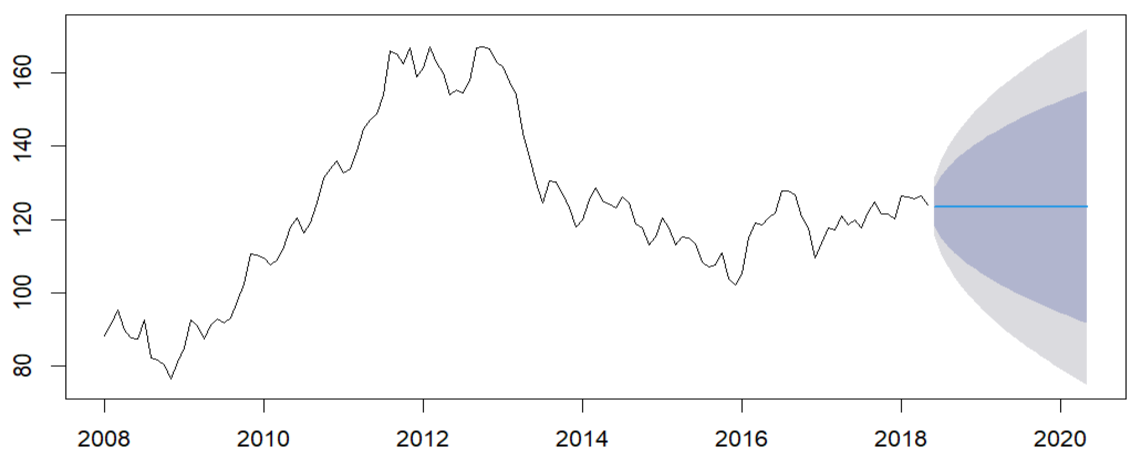
```
> Box.test(residuals(fit_model), lag=log(length(residuals(fit_model))), type="Ljung-Box", fitdf =1)

        Box-Ljung test

data:  residuals(fit_model)
X-squared = 0.27633, df = 3, p-value = 0.9644
```

H_0: The residuals are independently distributed (i.e., no autocorrelation)
H_1: The residuals are not independently distributed (i.e., autocorrelation).

- ○ **Ljung-Box test p > 0.05 ➔ residuals are independently distributed.**

- Forecast for **2 years**:

Forecast of Gold Prices for Next 2 Years



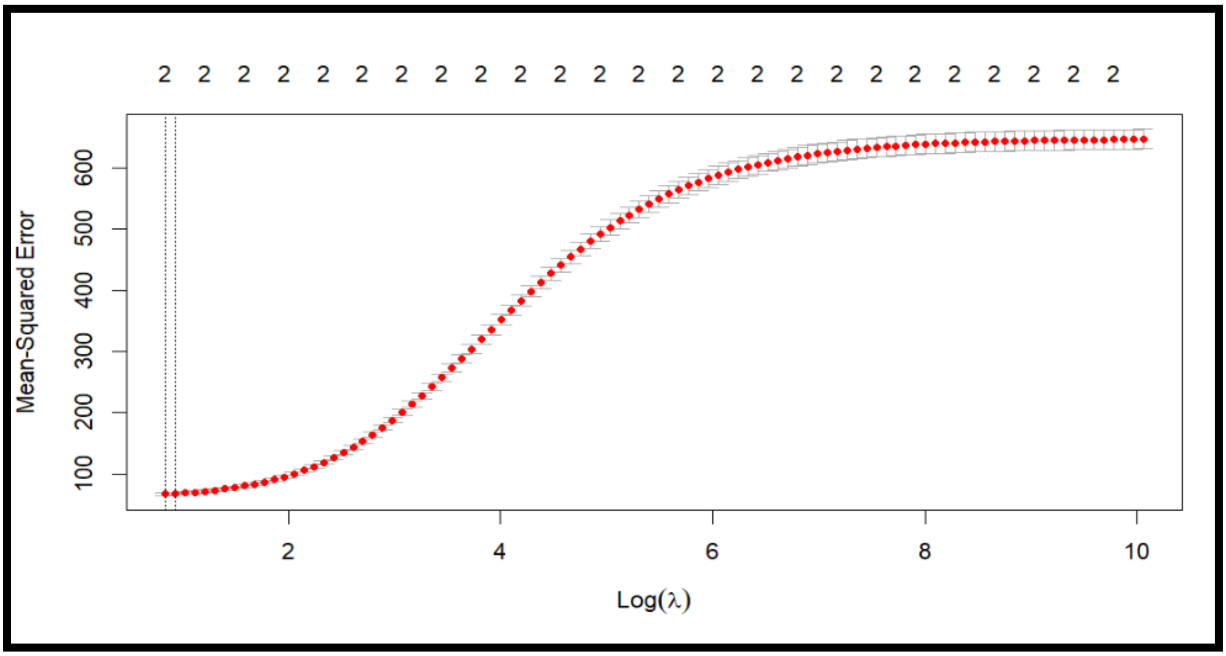
- Flat forecast with widening confidence intervals (reflecting increasing uncertainty while maintaining short-term accuracy).

- **ARIMA Performance**
  - **RMSE: 4.09**
  - **MAE: 3.35**
  - Indicates **strong predictive accuracy** leveraging temporal patterns.

Machine Learning Modeling (Ridge & Lasso)

- **Setup:**
  - Predictors: `USO_log`, `SLV_log`
  - Target: `GLD`
  - **Train-test split: 80%-20% (time-based, no shuffling)**
  - Used **cross-validation** for hyperparameter tuning.

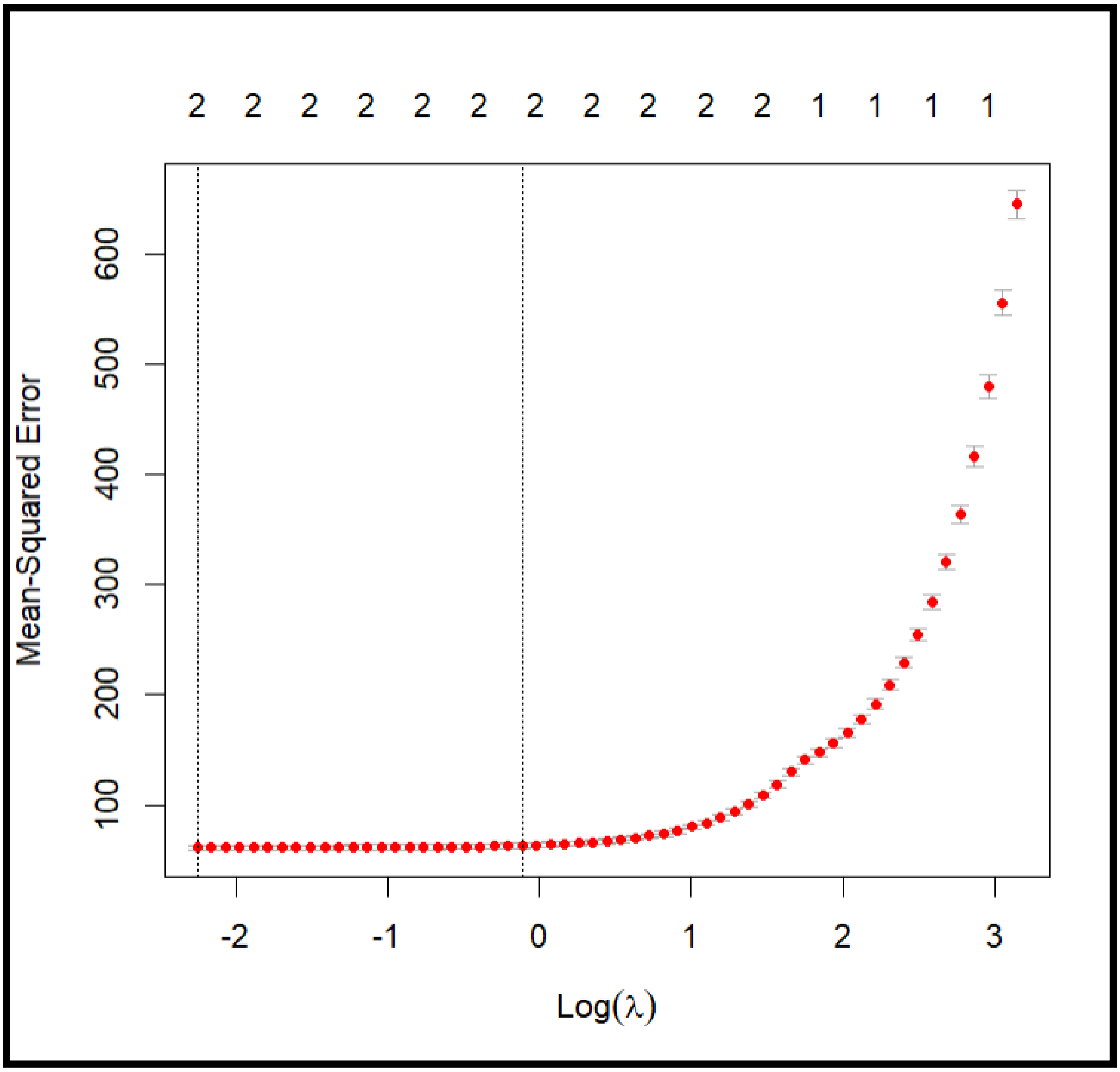
- **Ridge Regression (α = 0)**
  - Optimal λ (lambda.min): **2.322**, MSE: **66.31**
  - All predictors retained, no zeroing due to Ridge's nature.
  - **Performance:**
    - **RMSE: 6.05**
    - **MAE: 5.31**



```
> ridge_rmse <- sqrt(mean((ridge_pred - y_test)^2))
> ridge_rmse
[1] 6.051019
> ridge_mae  <- mean(abs(ridge_pred - y_test))
> ridge_mae
[1] 5.311087
```

- **Lasso Regression (α = 1)**

  - Optimal λ (lambda.min): **0.1053**, MSE: **61.08**
  - Both predictors retained at optimal λ.
  - Slightly higher error on the test set compared to Ridge:
    - **RMSE: 6.71**
    - **MAE: 5.98**



```
> lasso_rmse <- sqrt(mean((lasso_pred - y_test)^2))
> lasso_rmse
[1] 6.706304
> lasso_mae  <- mean(abs(lasso_pred - y_test))
> lasso_mae
[1] 5.97625
```

Results & Comparison

Model	RMSE	MAE
ARIMA	**4.09**	**3.35**
Ridge	6.05	5.31
Lasso	6.71	5.98

The **ARIMA model outperformed Ridge and Lasso**, demonstrating that **gold prices are better predicted using their historical patterns rather than using correlated economic indicators alone**.

This aligns with the strength of **time series models in capturing temporal dependencies** compared to regularized linear models which rely solely on cross-sectional relationships.

Conclusion

This project demonstrates **practical end-to-end financial time series forecasting**, covering:

- Data cleaning and preprocessing
- Exploratory Data Analysis (EDA)
- Feature engineering
- ARIMA time series forecasting
- Ridge and Lasso regression comparison
- Model evaluation using RMSE/MAE
- Diagnostic validation for residual behavior

Key takeaway:

Gold prices exhibit strong temporal structures best captured by ARIMA, making it a reliable tool for forecasting in financial planning and investment scenarios.