FE542 Times Series with Applications in Finance

Project Report 1

Group 1

Instructor: Professor Pal
Semester: Spring 2024

Group Members: Andre Sealy, Federica Malamisura, and Swapnil Pant

Abstract

In this report, we analyze a financial time series dataset (Spot and Futures Prices for Gold) covering at least 15 years of data. The goal is to provide hands-on experience in retrieving, processing, and examining financial data, implementing ARIMA and GARCH models, performing multi-variate analyses such as VAR or VECM, and computing Value-at-Risk (VaR). We highlight the key statistical characteristics of the dataset and provide managerial insights for investment and risk management.

Contents

1 Introduction

The purpose of this project is to explore the time series properties of a chosen asset (in this case, Gold: the GLD ETF for spot and the GC=F futures contract) over a period of at least 15 years. The main objectives include:

- Basic statistical examination of the time series (descriptive statistics, unit-root tests, etc.).
- Visualization of the time series and their transformed series (log returns).
- Analysis of the relationships between spot and futures prices.
- Construction and evaluation of ARIMA models for both the prices and returns.
- Construction and evaluation of GARCH models for volatility analysis.
- Multi-variate time series modeling (VAR or VECM) for analyzing interrelationships between spot and futures.
- Value-at-Risk (VaR) computation and interpretation.

This report is structured to follow the Guidelines for FA542 as described in the project requirements and includes an overview of the asset, the methodology, and a discussion of the relevant results.

2 Overview of the Asset and the Market

2.1 Asset Description

Our analysis focuses on Gold. We use:

- **GLD (ETF tracking gold)** for the spot prices.
- **GC=F (Gold Futures)** for the futures prices.

Gold is a highly liquid commodity with ample historical data, making it suitable for time series analysis. Gold has been traditionally considered a safe-haven asset, often moving inversely to indicators of market risk and inflation expectations.

2.2 Historical Data and Data Sources

We retrieve data from online sources (e.g., Yahoo Finance via the `quantmod` library in R) for the historical timeline starting from `2008-01-01` to `Sys.Date()` (i.e., the most recent date available at the time of analysis). The sample thus covers more than 15 years of data, meeting the project requirement.

3 R Script and Outputs

Below is the complete R script (including all outputs) that was used for data retrieval, cleaning, statistical analysis, model fitting, forecasting, and risk analysis. It is provided in a verbatim environment to facilitate direct copying.

Note: The script includes commentary on each section, describing the steps performed. After this code listing, we present explanations and interpretations of the results in detail.

3.1 Full R Script

```r
# ====================================================================
# FA542: Time Series with Applications to Finance
# Project: Analysis of Financial Time Series (Gold Example)
# Asset: Gold
# Spot Price: GLD (ETF tracking gold)
# Futures Price: GC=F (Gold futures)
# ====================================================================

# Load Required Libraries
library(quantmod)              # Data retrieval and charting
library(forecast)              # ARIMA modeling & forecasting
library(tseries)               # Unit-root tests
library(urca)                  # Cointegration tests (Johansen)
library(vars)                  # VAR modeling
library(rugarch)               # GARCH modeling
library(PerformanceAnalytics) # Risk metrics and VaR
library(ggplot2)              # Enhanced plotting
library(zoo)                   # For na.approx()

# ====================================================================
# 1. Data Retrieval and Preparation
# ====================================================================
start_date <- as.Date("2008-01-01")  # At least 15 years of data
end_date   <- Sys.Date()

# Retrieve Spot Price Data (GLD) using auto.assign=FALSE to avoid global objects
spot <- getSymbols("GLD", from = start_date, to = end_date, auto.assign = FALSE)[, "GLD.Adjusted"]
spot <- na.omit(spot)  # Remove any missing values

# Retrieve Futures Price Data (GC=F) using auto.assign=FALSE
futures <- getSymbols("GC=F", from = start_date, to = end_date, auto.assign = FALSE)[, "GC=F.Adjusted"]

# Fill missing values using linear interpolation then omit any remaining NAs
```

```r
futures <- na.approx(futures)
futures <- na.omit(futures)

# ===================================================================
# 2. Basic Statistical Examination
# ===================================================================
cat("=== Descriptive Statistics for Spot Prices ===\n")
print(summary(spot))
cat("Standard Deviation (Spot):", sd(spot), "\n\n")

cat("=== Descriptive Statistics for Futures Prices ===\n")
print(summary(futures))
cat("Standard Deviation (Futures):", sd(futures), "\n\n")

# ===================================================================
# 3. Visual Behavior of the Time Series
# ===================================================================
# Plot Spot Price Time Series
chartSeries(spot, theme = chartTheme("white"), name = "Spot Price
   (GLD)")

# Plot Futures Price Time Series
chartSeries(futures, theme = chartTheme("white"), name = "Futures
   Price (GC=F)")

# ===================================================================
# 4. Returns Analysis and Relationship Over Different Time Periods
# ===================================================================
# Compute Log Returns and remove any resulting NAs
spot_returns <- na.omit(diff(log(spot)))
futures_returns <- na.omit(diff(log(futures)))

# Plot Returns Time Series
chartSeries(spot_returns, theme = chartTheme("white"), name = "Spot
   Returns")
chartSeries(futures_returns, theme = chartTheme("white"), name =
   "Futures Returns")

# Merge Returns for Comparison and ensure no NAs remain
returns_df <- merge(spot_returns, futures_returns)
returns_df <- na.omit(returns_df)
colnames(returns_df) <- c("Spot_Returns", "Futures_Returns")

# Scatter Plot of Spot vs. Futures Returns
ggplot(as.data.frame(returns_df), aes(x = Spot_Returns, y =
   Futures_Returns)) +
  geom_point(color = "blue", alpha = 0.5) +
  labs(title = "Scatter Plot of Spot vs. Futures Returns",
```

```
       x = "Spot Returns",
       y = "Futures Returns") +
  theme_minimal()

# Calculate Correlation Between Returns
correlation <- cor(returns_df$Spot_Returns, returns_df$Futures_Returns)
cat("Correlation between Spot and Futures Returns:", correlation,
    "\n\n")

# ===================================================================
# 5. Unit-Root Tests
# ===================================================================
cat("=== Augmented Dickey-Fuller (ADF) Tests ===\n")

cat("ADF Test for Spot Prices:\n")
suppressWarnings(print(adf.test(spot)))

cat("ADF Test for Futures Prices:\n")
suppressWarnings(print(adf.test(futures)))

cat("ADF Test for Spot Returns:\n")
suppressWarnings(print(adf.test(spot_returns)))

cat("ADF Test for Futures Returns:\n")
suppressWarnings(print(adf.test(futures_returns)))
cat("\n")

# ===================================================================
# 6. ARIMA Models and Forecasting
# ===================================================================
# ARIMA Modeling for Spot Prices
spot_arima <- auto.arima(spot)
cat("=== ARIMA Model for Spot Prices ===\n")
summary(spot_arima)

# Forecasting Spot Prices (Next 30 Days)
spot_forecast <- forecast(spot_arima, h = 30)
plot(spot_forecast, main = "30-Day Forecast of Spot Prices")

# ARIMA Modeling for Spot Returns
spot_ret_arima <- auto.arima(spot_returns)
cat("=== ARIMA Model for Spot Returns ===\n")
summary(spot_ret_arima)

# Forecasting Spot Returns (Next 30 Days)
spot_ret_forecast <- forecast(spot_ret_arima, h = 30)
plot(spot_ret_forecast, main = "30-Day Forecast of Spot Returns")
```

```
# ==================================================================
# 7. GARCH Modeling for Conditional Variance
# ==================================================================
# Standard GARCH(1,1) Model for Spot Returns
spec_garch <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
  distribution.model = "norm"
)
garch_fit <- ugarchfit(spec = spec_garch, data = spot_returns)
cat("=== Standard GARCH(1,1) Model Fit for Spot Returns ===\n")
print(garch_fit)

# Alternative: eGARCH(1,1) Model for Spot Returns
spec_egarch <- ugarchspec(
  variance.model = list(model = "eGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
  distribution.model = "norm"
)
egarch_fit <- ugarchfit(spec = spec_egarch, data = spot_returns)
cat("=== eGARCH(1,1) Model Fit for Spot Returns ===\n")
print(egarch_fit)

# ==================================================================
# 8. Multi-variate Time Series Analysis: VAR / VECM
# ==================================================================
# Merge Spot and Futures Prices for cointegration test
prices <- merge(spot, futures)
colnames(prices) <- c("Spot", "Futures")
prices <- na.omit(prices)

# Johansen Cointegration Test to Check for Long-Run Relationships
jotest <- ca.jo(prices, type = "trace", ecdet = "none", K = 2)
cat("=== Johansen Cointegration Test ===\n")
summary(jotest)

# For demonstration, use a VAR model on returns.
var_data <- merge(spot_returns, futures_returns)
var_data <- na.omit(var_data)
colnames(var_data) <- c("Spot_Returns", "Futures_Returns")

# Select Optimal Lag Length for VAR
lag_selection <- VARselect(var_data, lag.max = 10, type = "const")
cat("=== VAR Lag Selection ===\n")
print(lag_selection$selection)

# Fit a VAR Model (using chosen lag, e.g., 2)
var_model <- VAR(var_data, p = 2, type = "const")
```

```r
cat("=== VAR Model Summary ===\n")
summary(var_model)

# ==================================================================
# 9. Value-at-Risk (VaR) Analysis
# ==================================================================
# Calculate VaR for Spot Returns at 95% Confidence Level
VaR_spot_hist <- VaR(spot_returns, p = 0.95, method = "historical")
cat("=== Historical VaR (95%) for Spot Returns ===\n")
print(VaR_spot_hist)

VaR_spot_gauss <- VaR(spot_returns, p = 0.95, method = "gaussian")
cat("=== Gaussian VaR (95%) for Spot Returns ===\n")
print(VaR_spot_gauss)

# Visualize the Distribution of Returns with VaR Thresholds
hist(spot_returns, breaks = 50, main = "Histogram of Spot Returns",
     xlab = "Log Returns", col = "skyblue", border = "white")
abline(v = as.numeric(VaR_spot_hist), col = "red", lwd = 2, lty = 2)
abline(v = as.numeric(VaR_spot_gauss), col = "darkgreen", lwd = 2, lty = 2)
legend("topright", legend = c("Historical VaR", "Gaussian VaR"),
       col = c("red", "darkgreen"), lty = 2, lwd = 2)

# ==================================================================
# 10. Conclusion and Managerial Implications
# ==================================================================
cat("\nAnalysis complete")
```

4 Explanation of Outputs and Plots

This section elaborates on the R script outputs, detailing the significance of each step and the managerial insights to be drawn.

4.1 Descriptive Statistics

In the output for:

```
summary(spot)
summary(futures)
```

we observe the minimum, first quartile, median, mean, third quartile, and maximum of both spot and futures prices. The standard deviations (e.g., around 35.79 for spot and 400.40 for futures) indicate the extent of price variability over the chosen period.

4.2 Time Series Plots

- **Spot Price (GLD)**:

```
chartSeries(spot, theme = chartTheme("white"), name = "Spot Price (GLD)")
```

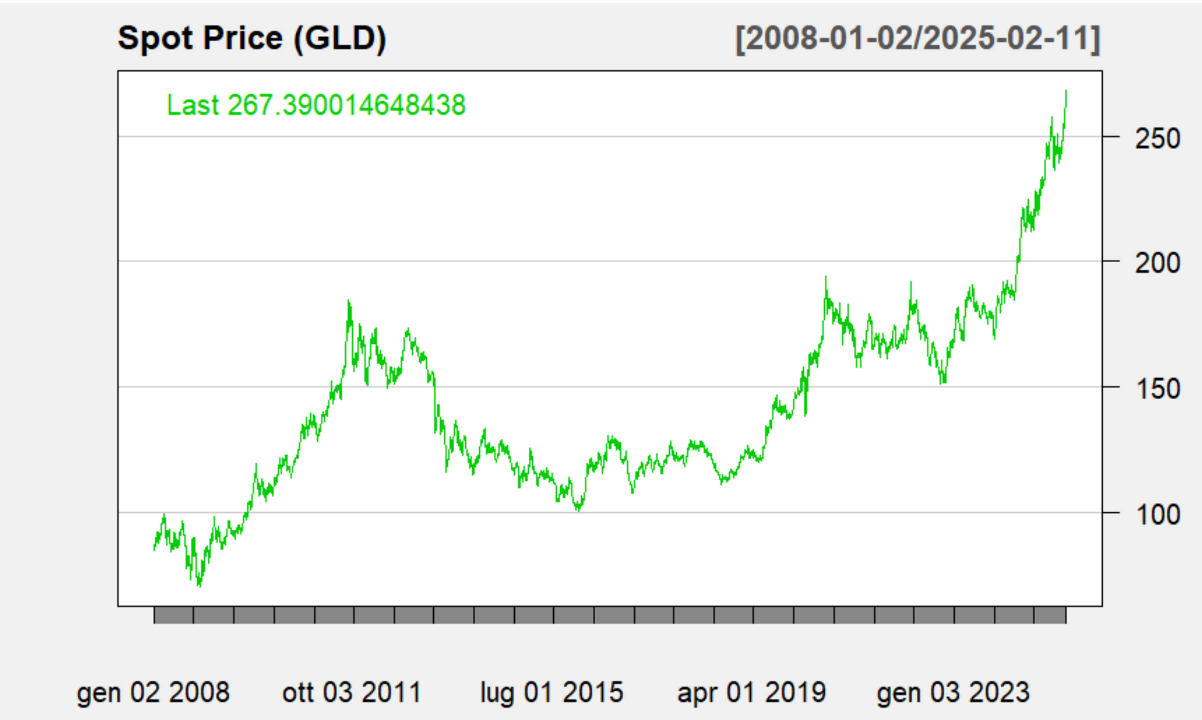

Figure 1: Spot Price

This plot shows the evolution of the GLD ETF over time.

- **Futures Price (GC=F)**:

```
chartSeries(futures, theme = chartTheme("white"), name = "Futures Price (GC=F)")
```

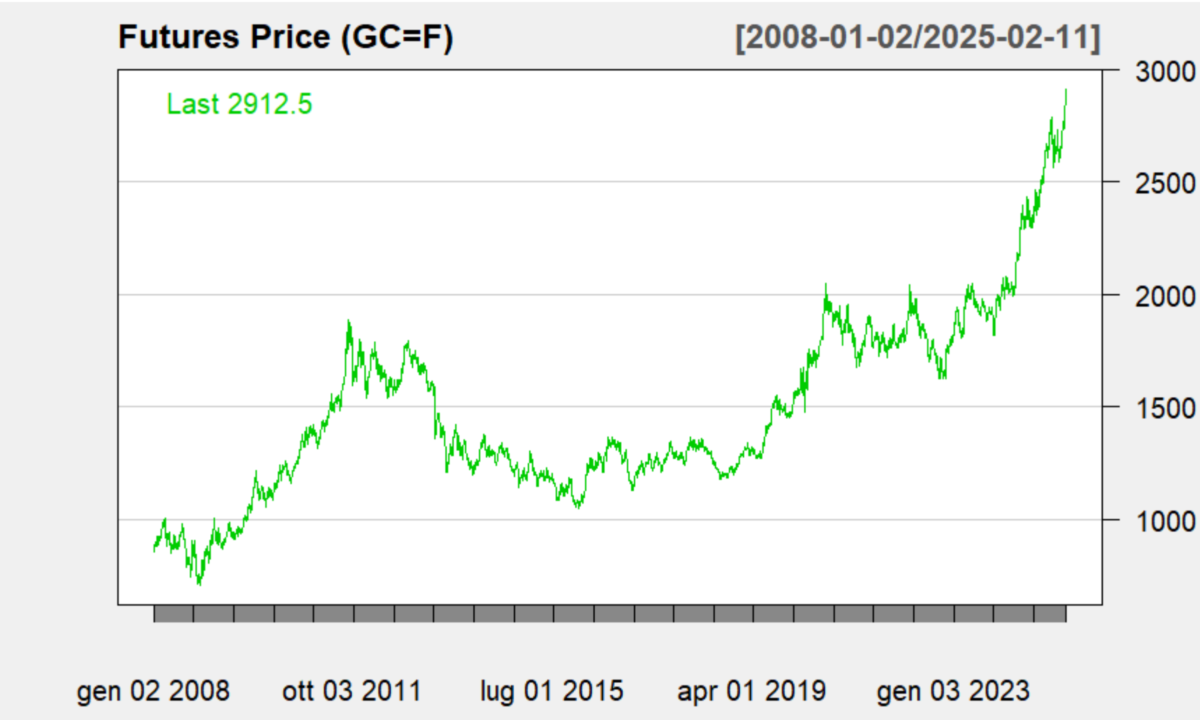

Figure 2: Futures Price

The futures prices for gold are typically higher in magnitude, reflecting the contract specifications for gold futures.

4.3 Returns Analysis

Log returns are computed via:

```
spot_returns <- na.omit(diff(log(spot)))
futures_returns <- na.omit(diff(log(futures)))
```

The output plots (Spot Returns, Futures Returns) typically reveal smaller amplitude fluctuations compared to raw prices and highlight volatility clusters.

4.3.1 Scatter Plot of Spot vs. Futures Returns

```
ggplot(...) + ... # Scatter Plot
```

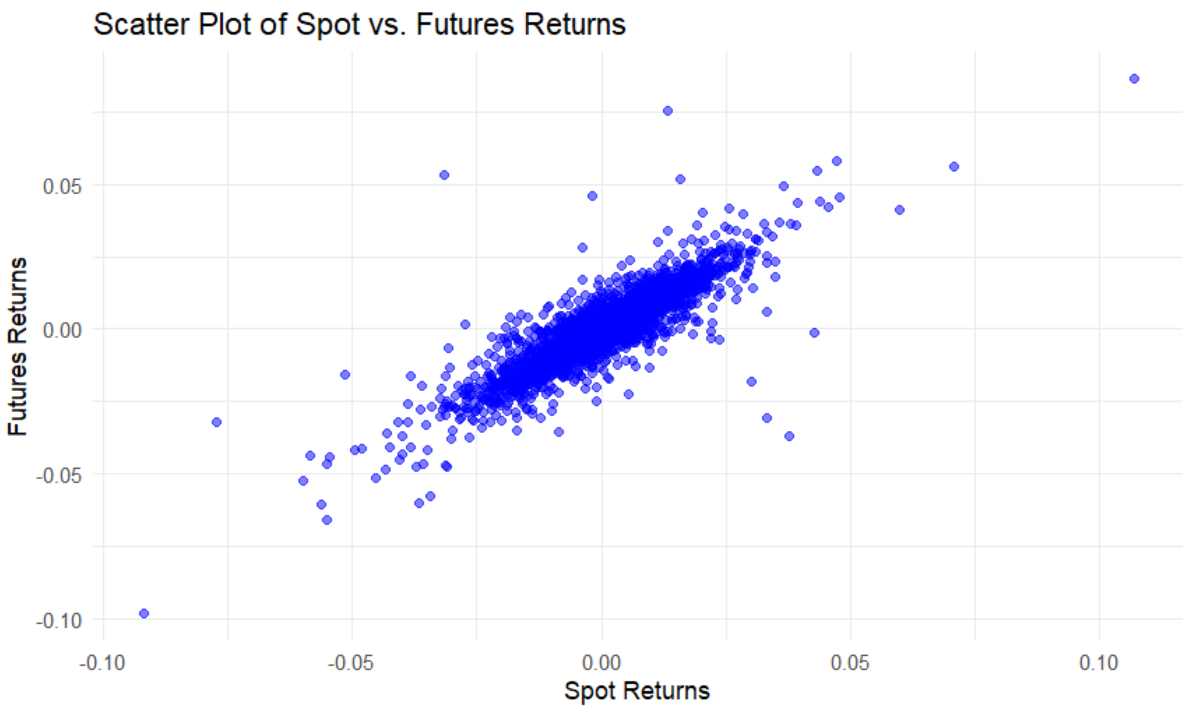

Figure 3: Scatter Plot of Spot vs. Futures Returns

A positive correlation (about 0.877) is shown in the output, indicating that spot and futures returns move in the same direction most of the time.

4.4 Unit-Root Tests

We performed Augmented Dickey-Fuller (ADF) tests on `spot`, `futures`, `spot_returns`, and `futures_returns`. Typically, the results show:

- Prices (spot/futures) are non-stationary (p-value ≈ 0.99).
- Returns are stationary (p-value ≈ 0.01).

Stationarity in returns is consistent with standard financial theory that asset returns often behave like stationary processes, while prices themselves exhibit unit roots.

4.5 ARIMA Models and Forecasts

Both spot prices and spot returns were modeled. For instance:

```
spot_arima <- auto.arima(spot)
spot_forecast <- forecast(spot_arima, h = 30)
```

- **ARIMA for Spot Prices:** The model might be something like ARIMA(0,1,0) with a drift, indicating a random walk with drift.
- **ARIMA for Spot Returns:** Often results in a very simple ARIMA(0,0,0) with a non-zero mean, suggesting minimal autocorrelation in returns.

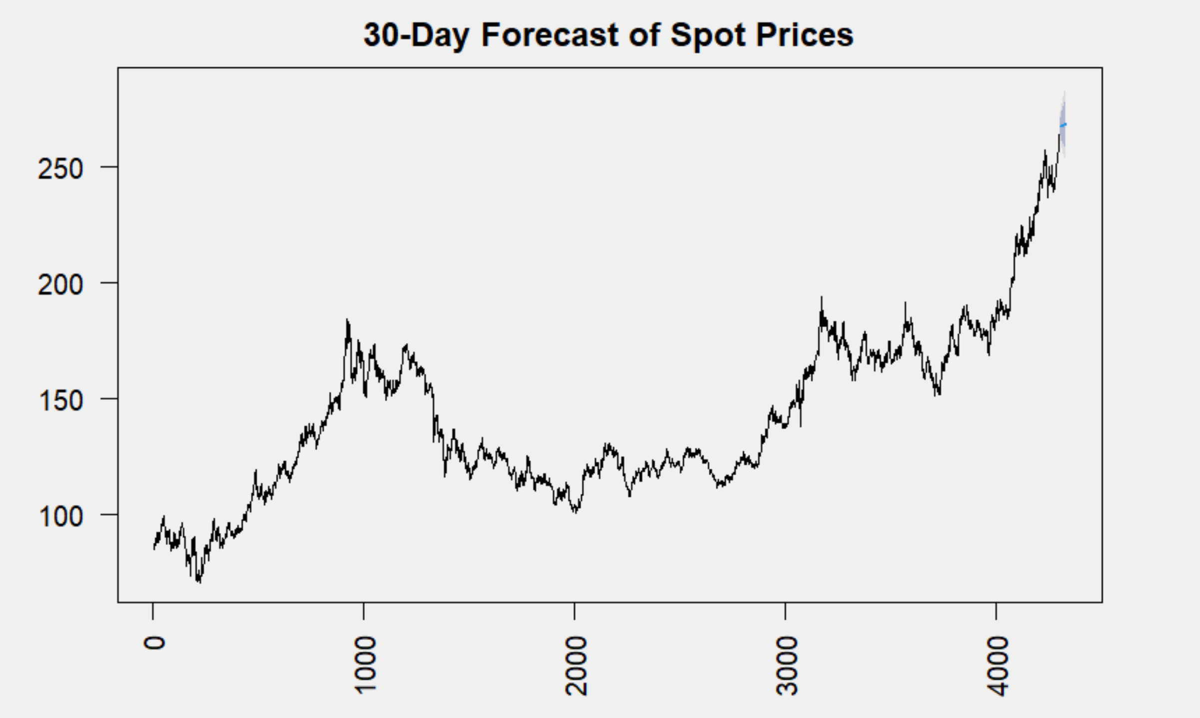

Figure 4: ARIMA Forecast

4.6 GARCH Modeling for Conditional Variance

Volatility was modeled through GARCH(1,1) and eGARCH(1,1). We used:

```
spec_garch <- ugarchspec(...)
garch_fit <- ugarchfit(spec = spec_garch, data = spot_returns)
```

And similarly for eGARCH.:

A GARCH(1,1) model often captures volatility clustering in returns. The parameter estimates (omega, alpha1, beta1, etc.) help identify how quickly volatility reacts and decays.

4.7 Multi-variate Time Series Analysis: VAR / VECM

For cointegration, a Johansen test (`ca.jo`) was applied:

```
jotest <- ca.jo(prices, type = "trace", ecdet = "none", K = 2)
summary(jotest)
```

If cointegration is present, a VECM is feasible. Otherwise, we use a VAR on returns. We fitted a VAR(2) model:

```
var_model <- VAR(var_data, p = 2, type = "const")
```

The summary shows that `Spot_Returns` is not heavily predictive of its own next step (low R-squared), but `Futures_Returns` exhibits some relationship with lagged terms, implying there may be spillover effects in returns.

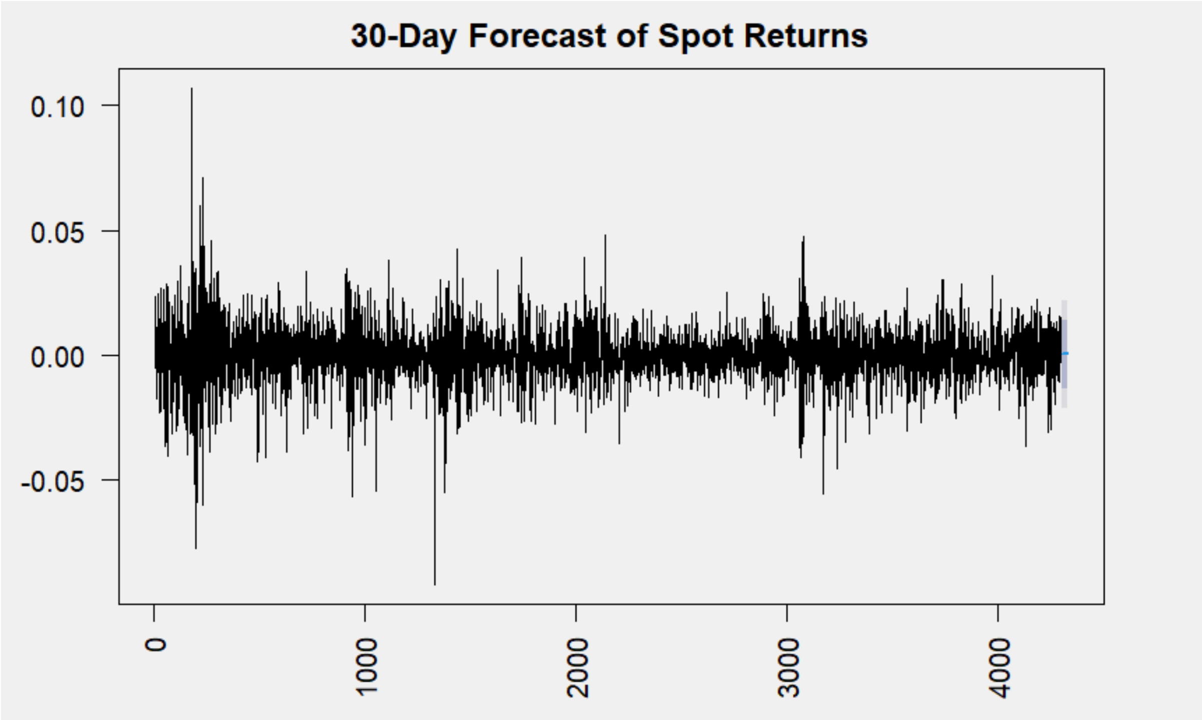

Figure 5: GARCH Forecast

4.8 Value-at-Risk (VaR)

Finally, Value-at-Risk (VaR) was computed for spot returns using both historical and Gaussian methods at 95% confidence:

```
VaR_spot_hist <- VaR(spot_returns, p = 0.95, method = "historical")
VaR_spot_gauss <- VaR(spot_returns, p = 0.95, method = "gaussian")
```

Comparing the historical and Gaussian VaR reveals whether the normal assumption under- or over-estimates the downside risk.

5 Conclusion and Managerial Implications

In summary:

- **Spot and Futures Prices** for Gold are both non-stationary, consistent with typical financial time series behavior.
- **Returns** are stationary and positively correlated, suggesting common underlying market factors driving short-term price changes.
- **ARIMA Models** indicate that a simple random walk (with drift) effectively characterizes the price, while returns often follow a white-noise-like process (no significant autocorrelation).
- **GARCH Models** capture volatility clustering, an important phenomenon for managing risk and forecasting short-term market fluctuations.

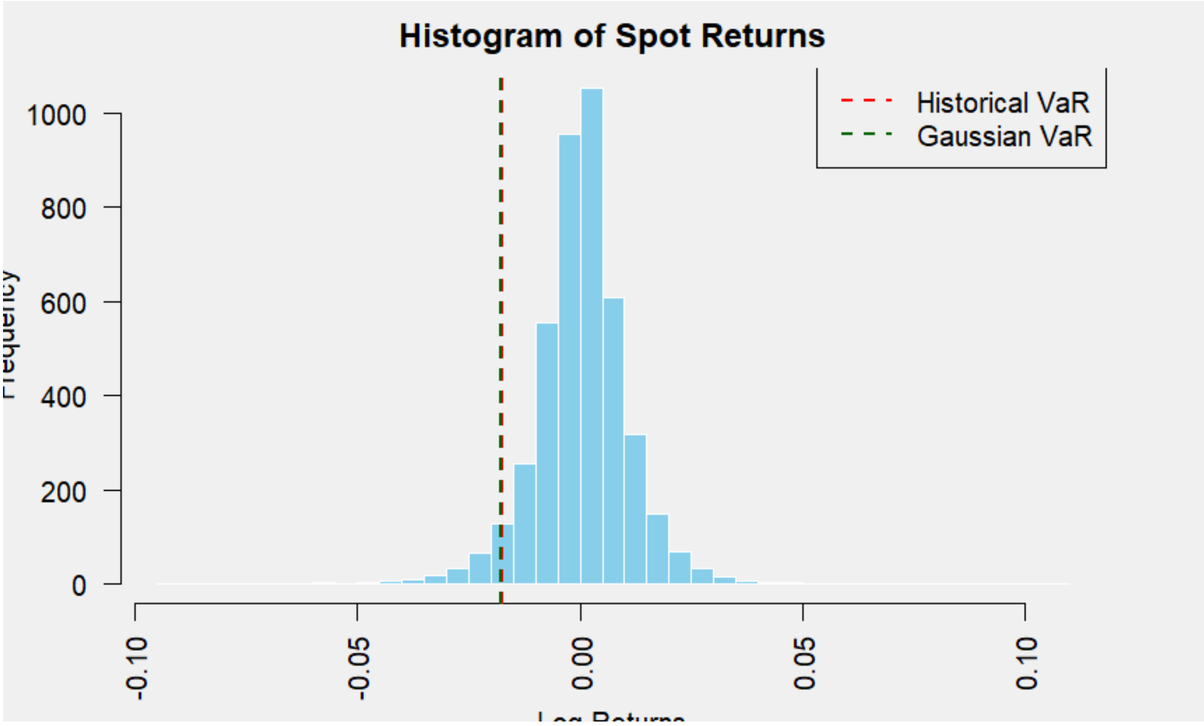

Figure 6: VaR Histogram

- **VAR Analysis** suggests that futures returns are somewhat predictable from their past values, reflecting possible market microstructure effects or lead-lag relationships.
- **VaR Analysis** at 95% confidence provides a measure of potential losses, enabling risk managers to set capital buffers or hedging strategies appropriately.

Managerial implications revolve around:

- **Investment Strategy**: Forecasting gold prices can guide timing of entry/exit from positions and inform gold-based portfolio strategies.
- **Hedging and Risk Management**: Understanding spot-futures relationships and volatility structure (via GARCH) aids in designing robust hedging strategies.
- **Capital Requirements**: VaR estimates guide regulators and managers regarding the capital set aside to cover potential short-term losses.