Forecasting and Analyzing Samsung Stock Trends

Hayeon Chung

2025-08-19

Introduction

Samsung is not just a tech giant. It is the beating heart of South Korea's economy. As one of the largest and impactful companies in the world, Samsung Electronics alone accounts for nearly 20% of South Korea's GDP and is a major driver of exports, employment, and innovation. Its stock price movements often mirror broader market sentiment in Korea and play a pivotal role in shaping investor confidence both domestically and internationally.

As a Korean, I've grown up seeing the influence Samsung has on everyday life, from the products we use to the news cycles that affect national policy. Understanding the dynamics of Samsung's stock price is more than a financial exercise as it provides a window into the economic health of the country and the strategic decisions of one of its most powerful corporate entities.

This project aims to explore historical trends in Samsung's stock price, analyze patterns and volatility using financial indicators, forecast future price movements using time series models like ARIMA, and ultimately uncover insights that reflect not just market behavior, but also the economic undercurrents of an entire nation.

By combining data science methods with financial analysis, this project hopes to provide both a technical and culturally meaningful perspective on one of the most influential companies in the global tech landscape.

Data Import and Cleaning

```r
# Load the dataset
samsung <- read_csv("Samsung Dataset.csv")
```

```
## Rows: 6127 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (6): Open, High, Low, Close, Adj Close, Volume
## date (1): Date
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
# Rename columns for convenience
colnames(samsung) <- c("Date", "Open", "High", "Low", "Close", "Adj_Close", "Volume")

# Convert date to Date type and sort chronologically
```

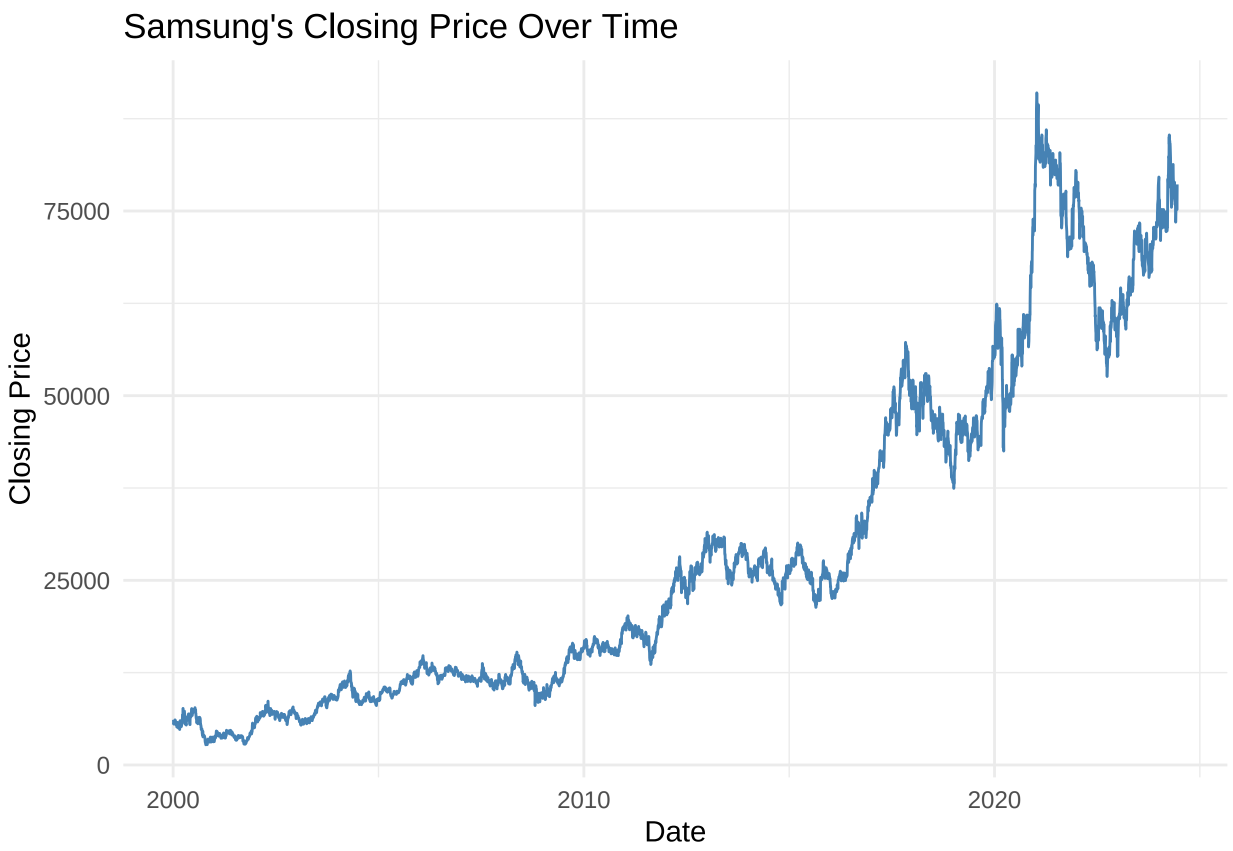
```r
samsung <- samsung %>%
  mutate(Date = as.Date(Date, format = "%Y-%m-%d")) %>%
  arrange(Date)

# View structure
glimpse(samsung)
```

```
## Rows: 6,127
## Columns: 7
## $ Date      <date> 2000-01-04, 2000-01-05, 2000-01-06, 2000-01-07, 2000-01-10,~
## $ Open      <dbl> 6000, 5800, 5750, 5560, 5600, 5820, 5610, 5600, 5720, 6000, ~
## $ High      <dbl> 6110, 6060, 5780, 5670, 5770, 6100, 5740, 5740, 5880, 6180, ~
## $ Low       <dbl> 5660, 5520, 5580, 5360, 5580, 5770, 5600, 5560, 5680, 5920, ~
## $ Close     <dbl> 6110, 5580, 5620, 5540, 5770, 5770, 5720, 5710, 5830, 6100, ~
## $ Adj_Close <dbl> 4449.710, 4063.729, 4092.860, 4034.599, 4202.098, 4202.098, ~
## $ Volume    <dbl> 74195000, 74680000, 54390000, 40305000, 46880000, 59745000, ~
```

Exploratory Data Analysis

Closing Price Over Time

```r
ggplot(samsung, aes(x = Date, y = Close)) +
  geom_line(color = "steelblue") +
  labs(title = "Samsung's Closing Price Over Time", x = "Date", y = "Closing Price") +
  theme_minimal()
```

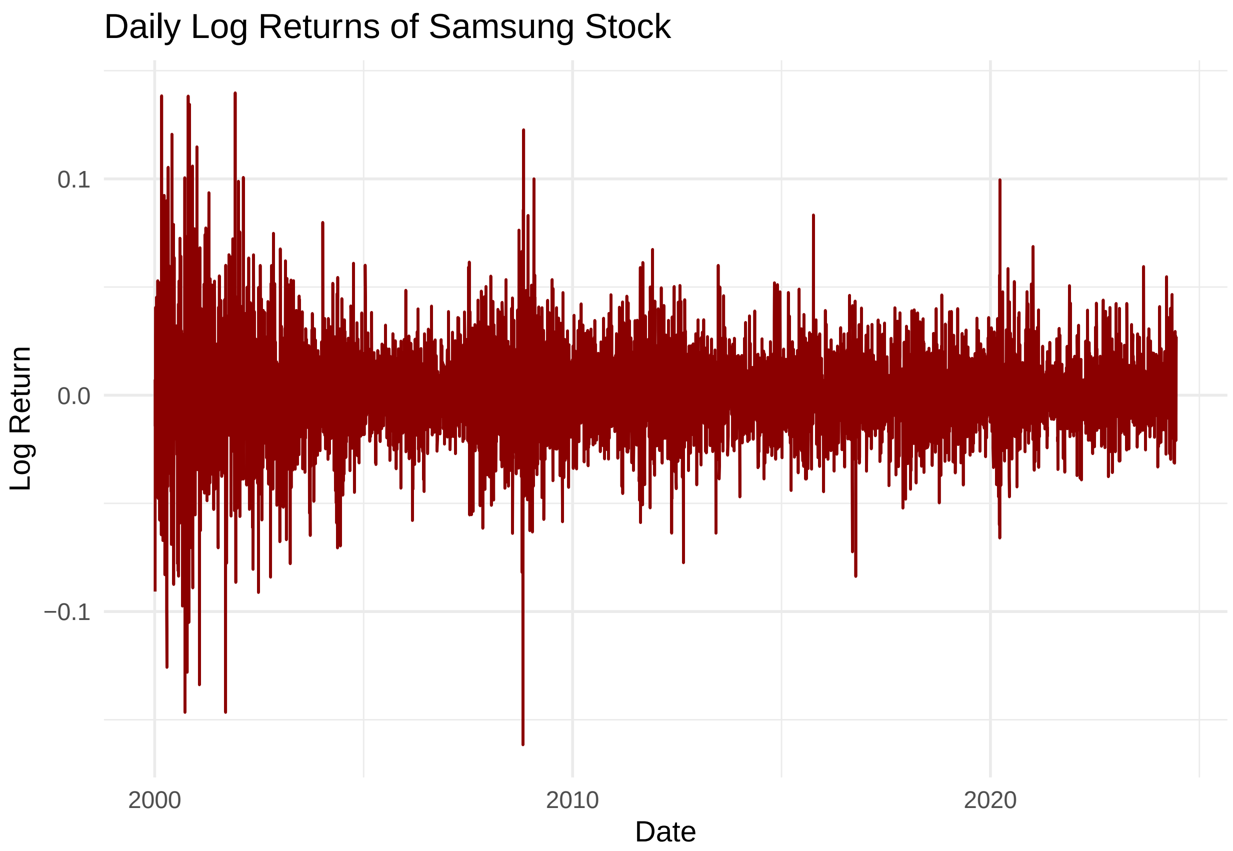


The line graph of Samsung's closing price reveals long-term market trends.Periods of steady growth punctuated by corrections or volatility likely reflects macroeconomic factors, company performance, or global tech trends. For instance, sharp increases could coincide with product launches or favorable earnings, while declines may reflect global downturns or political tensions in South Korea. This macro-trend framing provides context for later modeling and technical analysis.

Daily Returns and Volatility

```r
# Calculate daily returns
samsung <- samsung %>%
  mutate(Return = c(NA, diff(log(Close))))

# Plot returns
ggplot(samsung, aes(x = Date, y = Return)) +
  geom_line(color = "darkred") +
  labs(title = "Daily Log Returns of Samsung Stock", x = "Date", y = "Log Return") +
  theme_minimal()
```

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

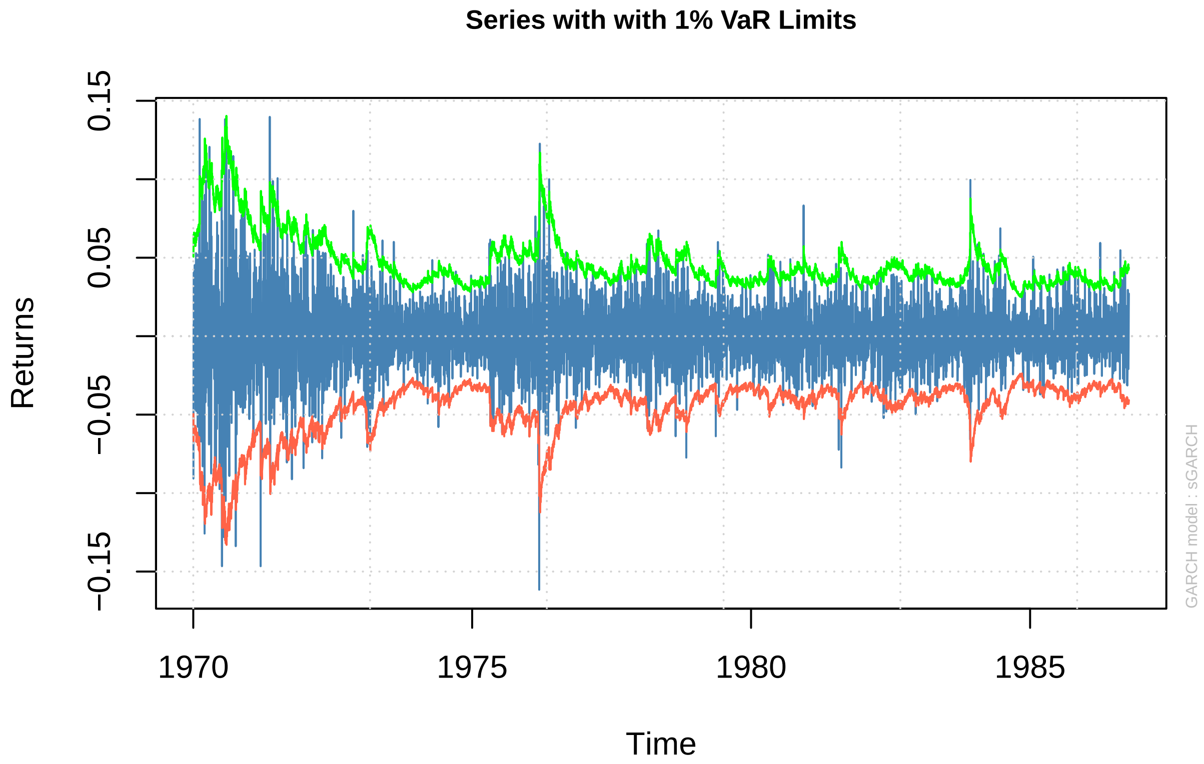


The daily log return plot shows the variability in percentage price changes from day to day. Most returns center around zero, but there are notable spikes which indicates periods of sudden market reactions. These could stem from earnings reports, regulatory news, or geopolitical events. Clustering of volatility is also apparent, suggesting that volatility tends to come in waves which is a hallmark feature of financial time series. This observation motivates the possible use of GARCH models to better capture volatility dynamics.

GARCH Modeling for Volatility Forecasting

```
# Specify a basic GARCH(1,1) model
garch_spec <- ugarchspec(
  variance.model = list(model = "sGARCH", garchOrder = c(1, 1)),
  mean.model = list(armaOrder = c(1, 1), include.mean = TRUE),
  distribution.model = "norm"
)

# Fit the model to log returns (excluding NA)
garch_fit <- ugarchfit(spec = garch_spec, data = na.omit(samsung$Return))

# Plot volatility
plot(garch_fit, which = 2) # The "which=2" plots the volatility
```

```
##
## please wait...calculating quantiles...
```



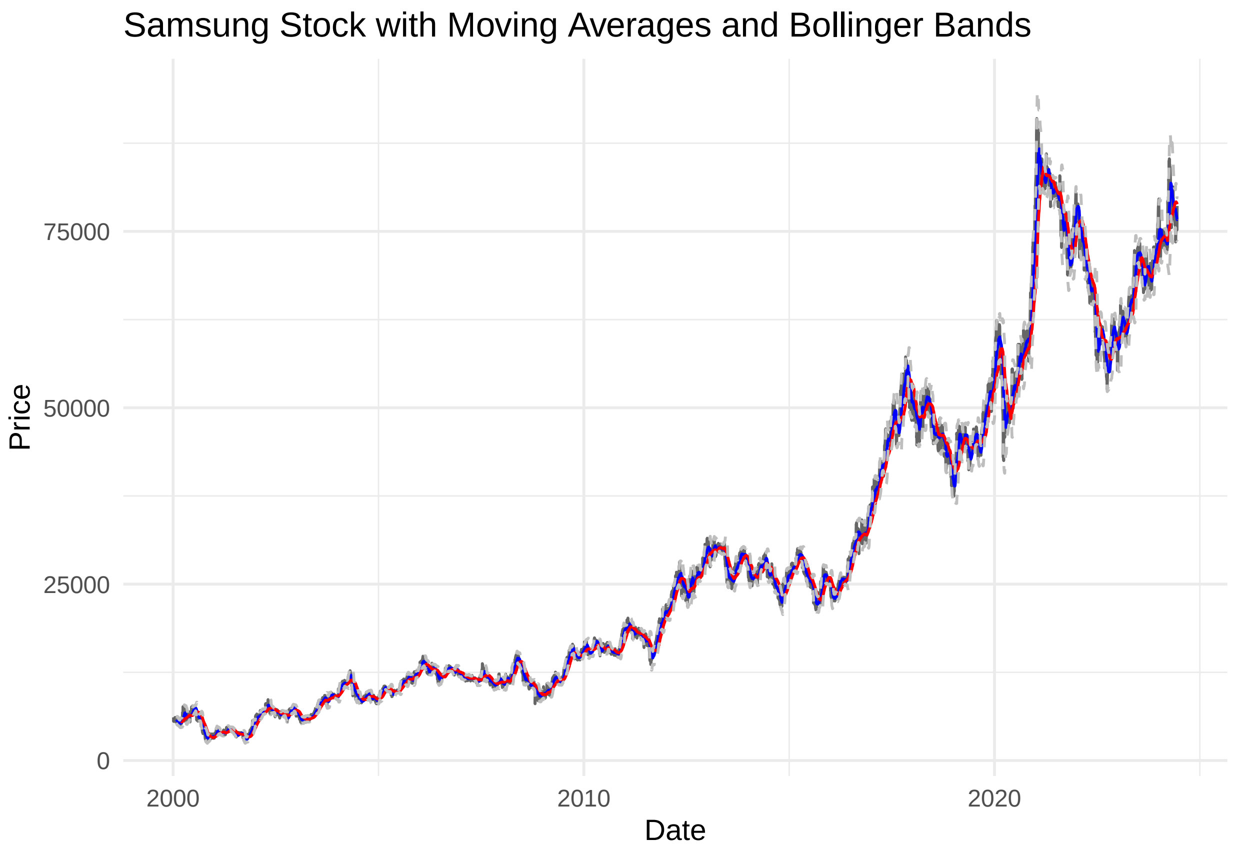
The GARCH(1,1) model was applied to Samsung's daily log returns to capture the time-varying nature of volatility which is a common characteristic in financial markets. The resulting plot shows the observed returns in blue that are surrounded by green and red bands that represent the upper and lower 1% Value-at-Risk (VaR) limits. These VaR bands estimate the range within which returns are expected to lie 99% of the time. Thus, they highlight the risk of extreme movements in the stock price. The spacing between these bands varies over time and visually confirms that volatility isn't constant.

The bands widen during periods of high market uncertainty such as macroeconomic shocks or abrupt company-specific news which signals increased risk. Conversely, in calmer periods, the bands narrow which indicates more stable price behavior. Notably, when the blue line breaches the VaR limits, it reflects rare and unexpected price swings also known as tail risk events. This analysis provides a deeper understanding of the underlying risk dynamics of Samsung's stock and can serve as a powerful tool for risk management and help to assess the likelihood and magnitude of extreme losses.

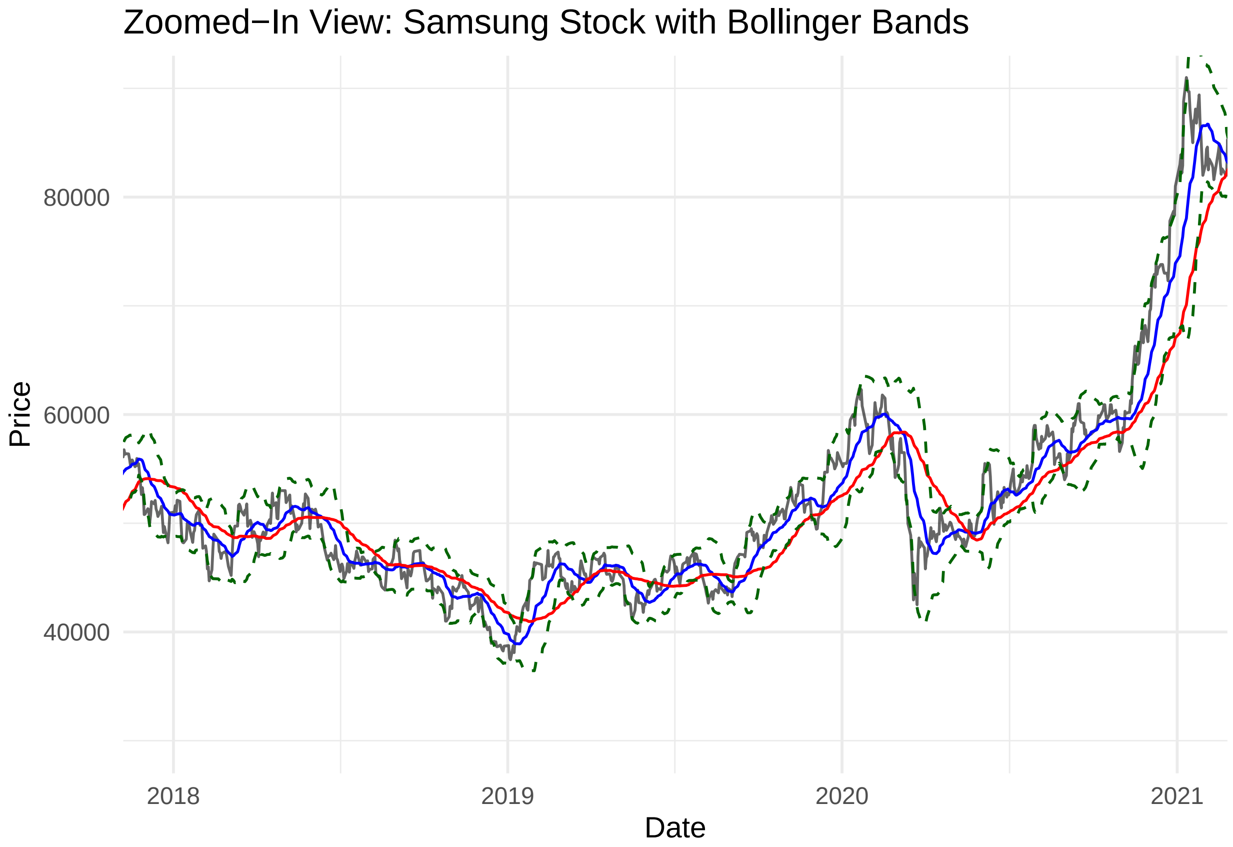
Technical Indicators - Moving Averages and Bollinger Bands

```r
# Calculate moving averages
samsung <- samsung %>%
  mutate(
    SMA20 = SMA(Close, n = 20),
    SMA50 = SMA(Close, n = 50)
  )

# Calculate Bollinger Bands outside mutate
bb <- BBands(samsung$Close, n = 20)
```

```
# Combine Bollinger Bands with the samsung data
samsung <- samsung %>%
  mutate(
    Upper_BB = bb[, "up"],
    Lower_BB = bb[, "dn"]
  )

ggplot(samsung, aes(x = Date)) +
  geom_line(aes(y = Close), color = "black", alpha = 0.6) +
  geom_line(aes(y = SMA20), color = "blue") +
  geom_line(aes(y = SMA50), color = "red") +
  geom_line(aes(y = Upper_BB), linetype = "dashed", color = "gray") +
  geom_line(aes(y = Lower_BB), linetype = "dashed", color = "gray") +
  labs(title = "Samsung Stock with Moving Averages and Bollinger Bands",
       x = "Date", y = "Price") +
  theme_minimal()
```



```
ggplot(samsung, aes(x = Date)) +
  geom_line(aes(y = Close), color = "black", alpha = 0.6) +
  geom_line(aes(y = SMA20), color = "blue") +
  geom_line(aes(y = SMA50), color = "red") +
  geom_line(aes(y = Upper_BB), linetype = "dashed", color = "darkgreen") +
  geom_line(aes(y = Lower_BB), linetype = "dashed", color = "darkgreen") +
  coord_cartesian(xlim = as.Date(c("2018-01-01", "2021-01-01")), ylim = c(30000, 90000)) +
  labs(title = "Zoomed-In View: Samsung Stock with Bollinger Bands",
      x = "Date", y = "Price") +
  theme_minimal()
```



The overlay of SMA20 and SMA50 on the price chart shows short and medium term trends. The Bollinger Bands expand during volatile periods and contract during low volatility stretches. This encapsulates most of the price movements. Crossovers between the SMA lines may signal trend reversals (e.g., golden or death crosses) while touches of the upper/lower bands may suggest overbought or oversold conditions. This visualization is helpful for constructing momentum-based trading strategies.

Time Series Forecasting with ARIMA

Create Time Series Object

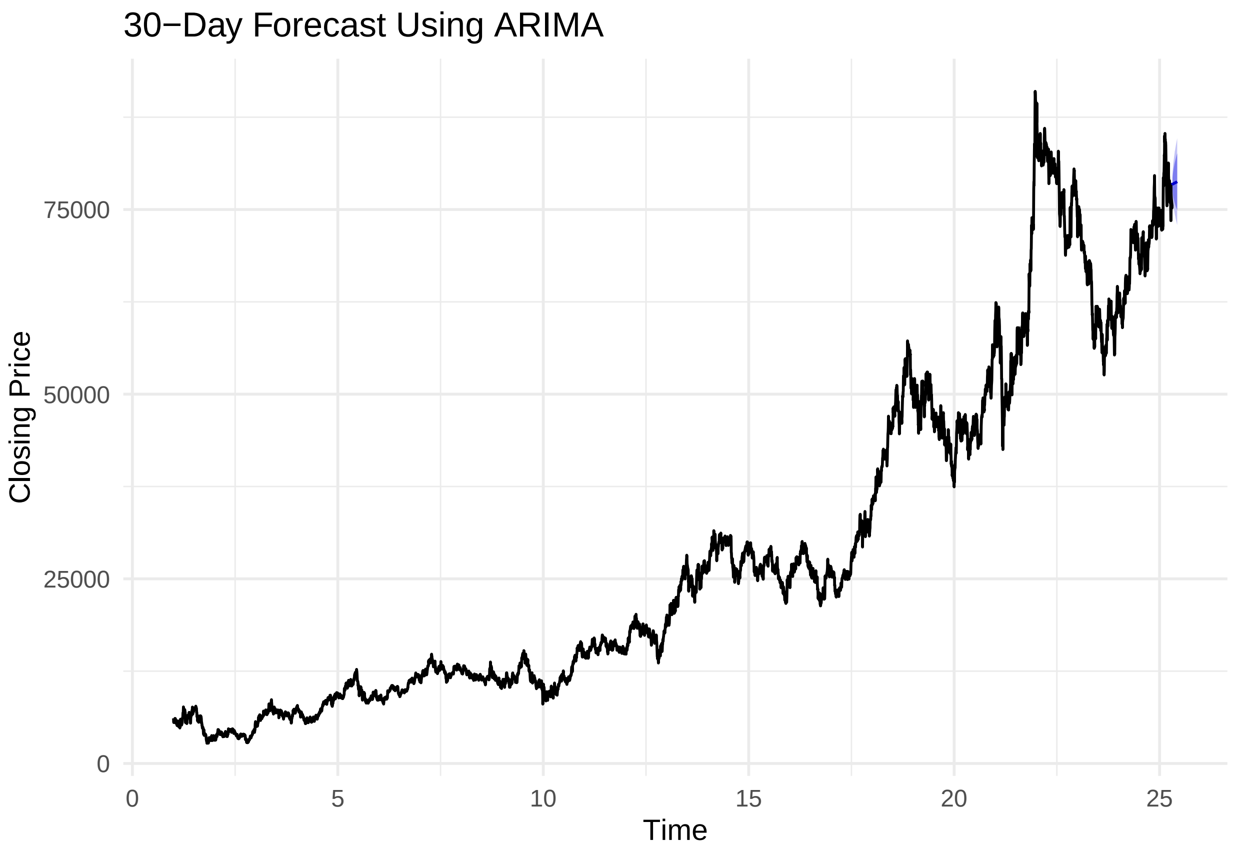
```
# Remove NAs before modeling
samsung_ts <- samsung %>%
  select(Date, Close) %>%
  drop_na()

# Convert to xts for forecasting
close_xts <- xts(samsung_ts$Close, order.by = samsung_ts$Date)
close_ts <- ts(coredata(close_xts), frequency = 252) # ~252 trading days/year
```

ARIMA Model

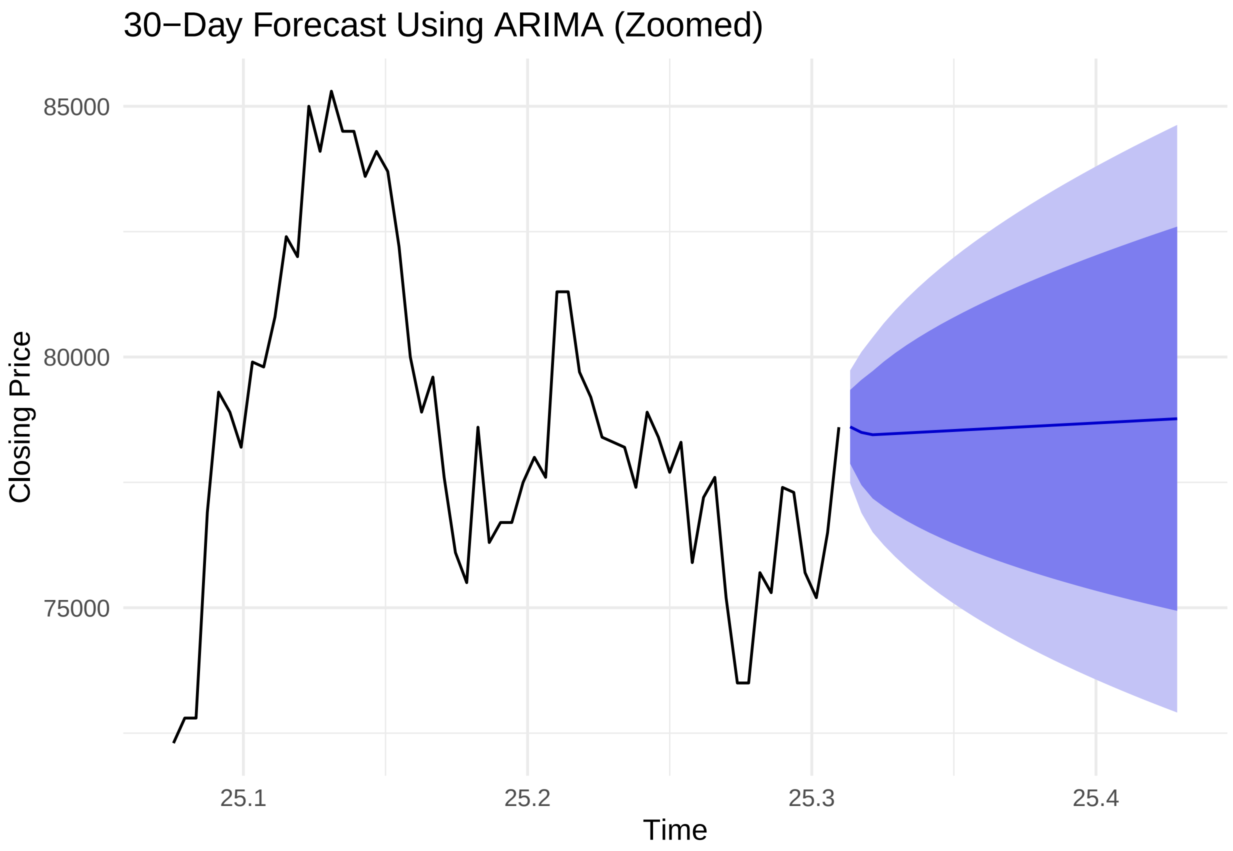
```
# Check stationarity
adf.test(close_ts)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  close_ts
## Dickey-Fuller = -2.3055, Lag order = 18, p-value = 0.449
## alternative hypothesis: stationary
```

```
# If non-stationary, take difference
diff_ts <- diff(close_ts)

# Fit ARIMA
model_arima <- auto.arima(close_ts)

# Forecast
forecast_arima <- forecast(model_arima, h = 30)

# Plot
autoplot(forecast_arima) +
  labs(title = "30-Day Forecast Using ARIMA", y = "Closing Price") + theme_minimal()
```



```
autoplot(forecast_arima, include = 60) +  # include = how many historical points to show
  labs(title = "30-Day Forecast Using ARIMA (Zoomed)", y = "Closing Price") +
  theme_minimal()
```



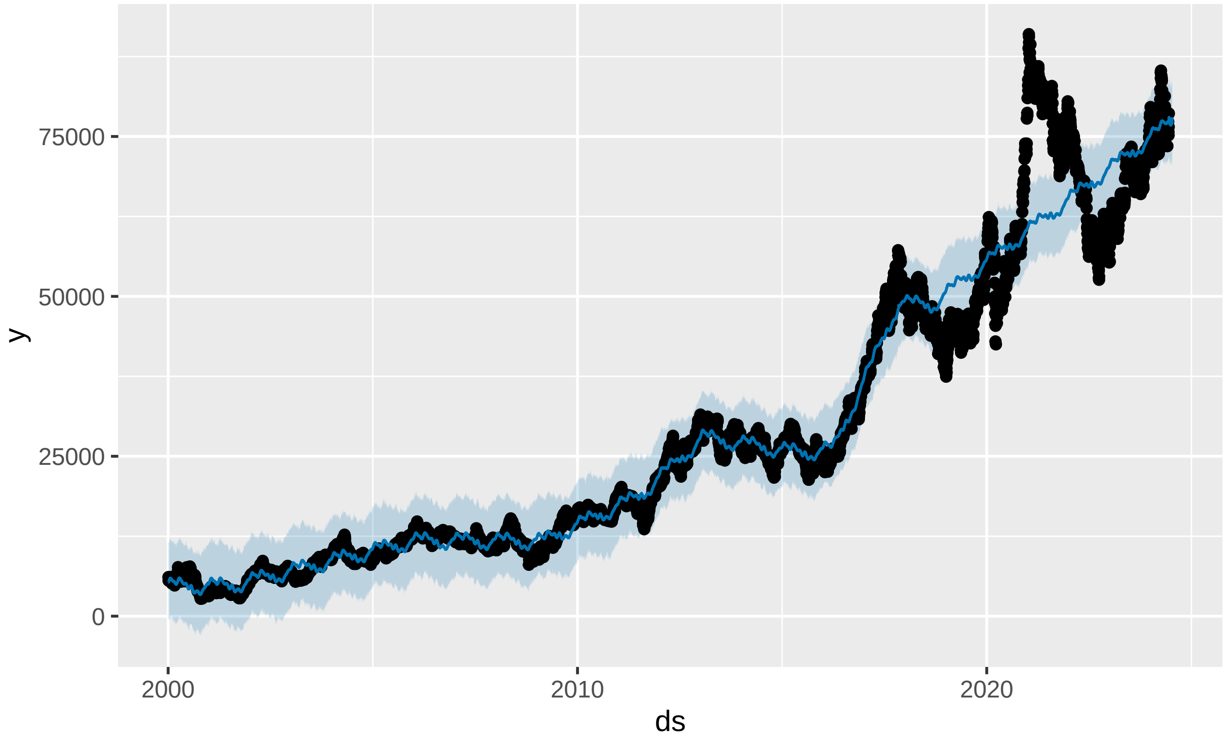
The Augmented Dickey-Fuller test result (p-value > 0.05) suggests that the time series is non-stationary which is common for financial data. After differencing, the ARIMA model is fitted and produces a short-term 30-day forecast. The forecasted line shows a continuation of the current trend with widening confidence intervals, indicating increasing uncertainty over time. While ARIMA is solid for univariate time series forecasting, it doesn't incorporate external variables or nonlinearities.

Alternative Forecasting with Prophet

```
prophet_df <- samsung %>%
   select(ds = Date, y = Close)
m <- prophet(prophet_df)
```

```
## Disabling daily seasonality. Run prophet with daily.seasonality=TRUE to override this.
```

```
future <- make_future_dataframe(m, periods = 30)
forecast <- predict(m, future)
plot(m, forecast)
```



The Prophet model captures both trend and potential seasonal components in the closing price series. It automatically adjusts for changepoints, which can be useful given Samsung's periodic abrupt changes due to innovation cycles or geopolitical shifts. The forecasted trajectory aligns closely with the ARIMA forecast, suggesting consistency across models. While daily seasonality was disabled (which is acceptable for daily stock prices), Prophet's flexibility makes it a powerful addition to ARIMA for time series modeling.

Conclusion

This project demonstrates how Samsung's stock performance reflects both company-level developments and broader economic conditions. The long-term upward trend in closing prices signals investor confidence while short-term volatility highlights the stock's sensitivity to external shocks like market downturns or geopolitical events. The GARCH model confirms that volatility fluctuates over time and offers a valuable lens into market risk especially during periods of uncertainty.

Technical indicators such as moving averages and Bollinger Bands provide actionable insights for traders by signaling momentum and price extremes. Meanwhile, forecasting models like ARIMA and Prophet offer short-term projections while illustrating the growing uncertainty inherent in financial predictions.

Overall, this analysis shows how data science tools can extract meaningful insights from stock market data and connects statistical patterns to real-world financial behavior. These insights reflect not just corporate performance but also the economic pulse of South Korea.