

An Analysis of S&P 500 and Commodity Price Correlations

Reginal Supoint

2024-11-26

Introduction

The purpose of this analysis is to explore the correlation between the S&P 500 index and various commodities such as oil, gold, silver, platinum and palladium. Understanding these correlations can provide insights into market dynamics and investment strategies.

This dataset includes daily closing prices for the S&P 500, NASDAQ, and several commodities. This analysis will include correlation analysis, time series forecasting and scenario analysis to understand potential future trends.

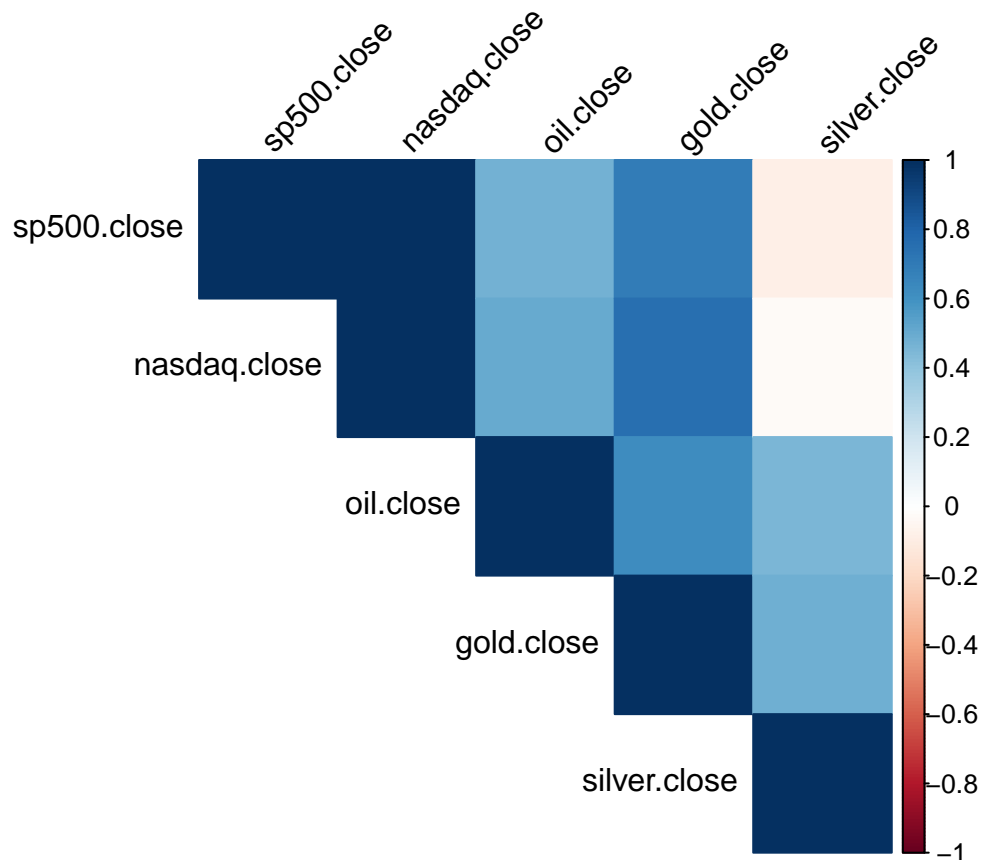
```
# Select relevant columns for correlation analysis
cor_data <- data_clean %>% select(sp500.close, nasdaq.close, oil.close, gold.close, silver.close)

# Calculate the correlation matrix
cor_matrix <- cor(cor_data)

# Print the correlation matrix
print(cor_matrix)

##           sp500.close nasdaq.close oil.close gold.close silver.close
## sp500.close  1.00000000  0.99050033 0.4765798  0.6996215 -0.08042028
## nasdaq.close 0.99050033  1.00000000 0.5017432  0.7523519 -0.02991301
## oil.close    0.47657977  0.50174317 1.0000000  0.6290720  0.45652019
## gold.close   0.69962150  0.75235187 0.6290720  1.0000000  0.48543082
## silver.close -0.08042028 -0.02991301 0.4565202  0.4854308  1.00000000

# Visualize the correlation matrix
corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", tl.srt = 45)
```



“ This correlation map provides a solid snapshot of how different assets are related to each other. We are particularly looking at how the commodities are related to the S&P 500.

Correlations Scores range from -1 to 1, With 1 being the strongest correlation and -1 being the weakest correlation.

S&P 500 and NASDAQ

With a score of 0.99 we can see that the S&P 500 and NASDAQ are almost perfectly positively correlated. Therefore, when the S&P 500 moves in a certain direction the NASDAQ is likely to move in the same direction.

Correlation Summary:

High Correlation: S&P 500 and NASDAQ, NASDAQ and Gold, S&P 500 and GOLD

Moderate Correlation: S&P 500 and Oil, NASDAQ and Oil, Oil and Gold, Oil and Silver.

Weak or No Correlation: S&P 500 and Silver, NASDAQ and Silver. Due to the Market Dynamics Silver does not move in tandem with the Stock Market. Silver is more influenced by supply and demand dynamics specific to industrial and investment usages versus the stock market. Silver also tend to be highly volatile meaning weak correlation with the stock market.

```
# Convert the date column to Date type
data_clean$date <- as.Date(data_clean$date, format = "%m/%d/%Y")

# Arrange data by date
data_clean <- data_clean %>% arrange(date)
```

```

horizon <- 5 *252

# Fit an ARIMA model to the oil closing prices
model <- auto.arima(data_clean$oil.close, seasonal = FALSE)

# Print the model summary
summary(model)

## Series: data_clean$oil.close
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1      mean
##      0.9394  47.2146
## s.e.  0.0553  18.4010
##
## sigma^2 = 71.02: log likelihood = -106.55
## AIC=219.09   AICc=220.02   BIC=223.3
##
## Training set error measures:
##              ME      RMSE      MAE      MPE      MAPE      MASE      ACF1
## Training set 0.3159122 8.141474 5.737392 -13.00707 27.23056 1.004424 0.1433156

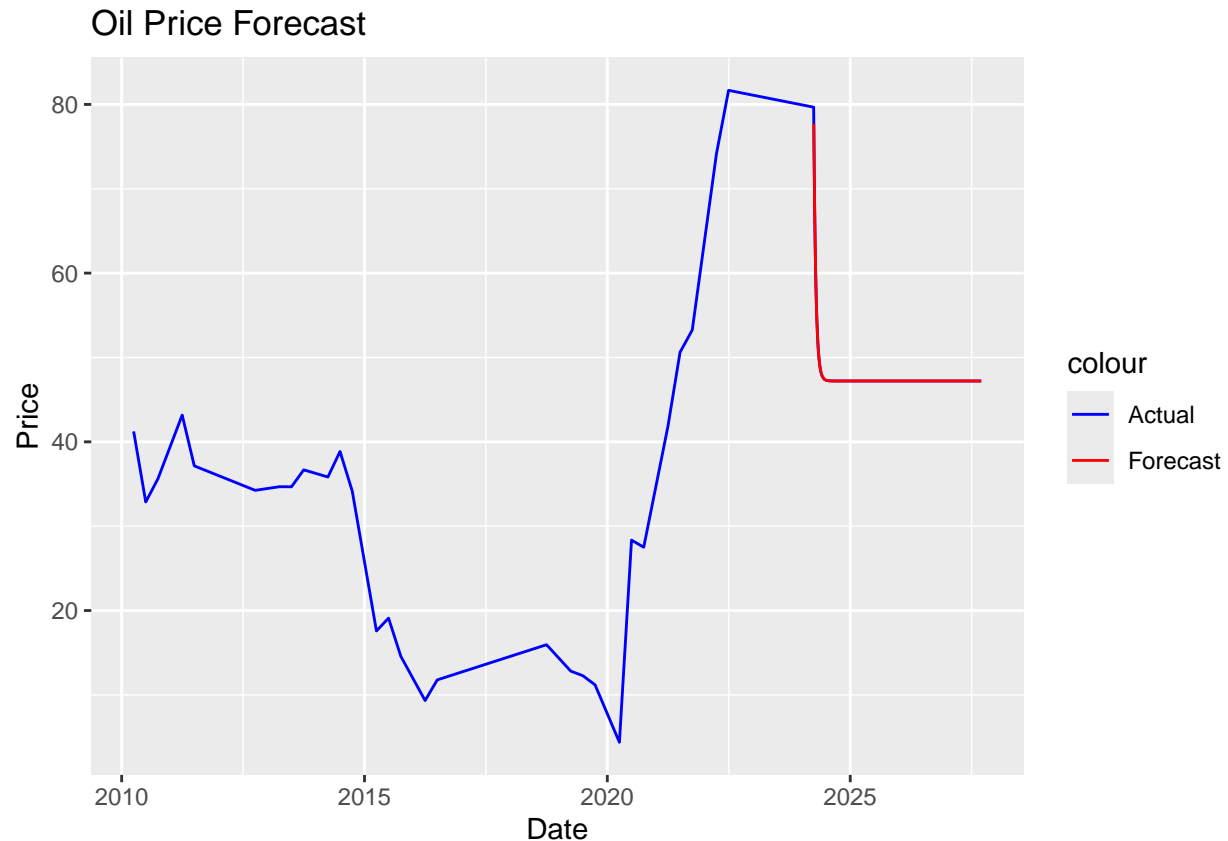
# Forecast future values
forecast_values <- forecast(model, h = horizon)

# Create a data frame with the forecast values and dates
forecast_df <- data.frame(
  date = seq.Date(from = max(data_clean$date) + 1, by = "day", length.out = horizon),
  forecast = as.numeric(forecast_values$mean)
)

# Combine the actual and forecast data for plotting
plot_data <- data_clean %>%
  select(date, oil.close) %>%
  rename(actual = oil.close) %>%
  bind_rows(forecast_df %>% rename(actual = forecast))

# Plot the actual and forecasted values
ggplot(plot_data, aes(x = date)) +
  geom_line(aes(y = actual, color = "Actual")) +
  geom_line(data = forecast_df, aes(y = forecast, color = "Forecast")) +
  labs(title = "Oil Price Forecast", x = "Date", y = "Price") +
  scale_color_manual(values = c("Actual" = "blue", "Forecast" = "red"))

```



In order to forecast the movement of stocks for the future I thought the ARIMA model would be the appropriate method. The model uses previous values of the time series in order to predict future outcomes. Which I thought would be an appropriate way to predict the value of Oil in a vacuum; before I go ahead and graph the future values of the other commodities.

The graph itself is forecasting for the price of oil to drop in 2025.

```
forecast_commodity <- function(data, column_name, horizon) {
  # Fit ARIMA model
  model <- auto.arima(data[[column_name]], seasonal = FALSE)
  # Forecast future values
  forecast_values <- forecast(model, h = horizon)
  # Create a data frame with the forecast values and dates
  forecast_df <- data.frame(
    date = seq.Date(from = max(data$date) + 1, by = "day", length.out = horizon),
    forecast = as.numeric(forecast_values$mean)
  )
  return(forecast_df)
}

# Forecast for each commodity
oil_forecast <- forecast_commodity(data_clean, "oil.close", horizon)
gold_forecast <- forecast_commodity(data_clean, "gold.close", horizon)
silver_forecast <- forecast_commodity(data_clean, "silver.close", horizon)
platinum_forecast <- forecast_commodity(data_clean, "platinum.close", horizon)
palladium_forecast <- forecast_commodity(data_clean, "palladium.close", horizon)
```

For the models I noticed that all of the prices for the Commodities stayed exactly on a straight line for the

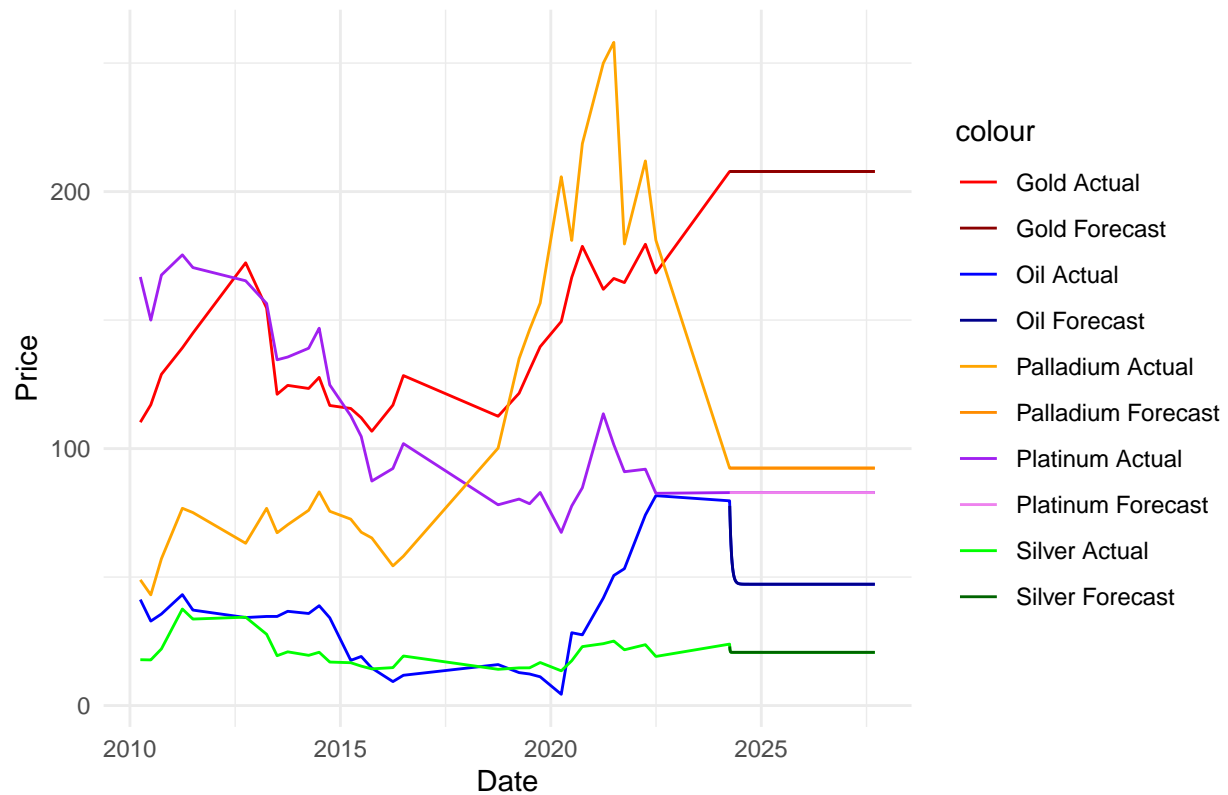
prediction that was trying to be made. Which made me consider if I was using the correct model to forecast the price.

```
# Combine forecast data with historical data
combine_data <- function(data, forecast_df, column_name) {
  combined <- data %>%
    select(date, !!sym(column_name)) %>%
    rename(actual = !!sym(column_name)) %>%
    bind_rows(forecast_df %>% rename(actual = forecast))
  return(combined)
}

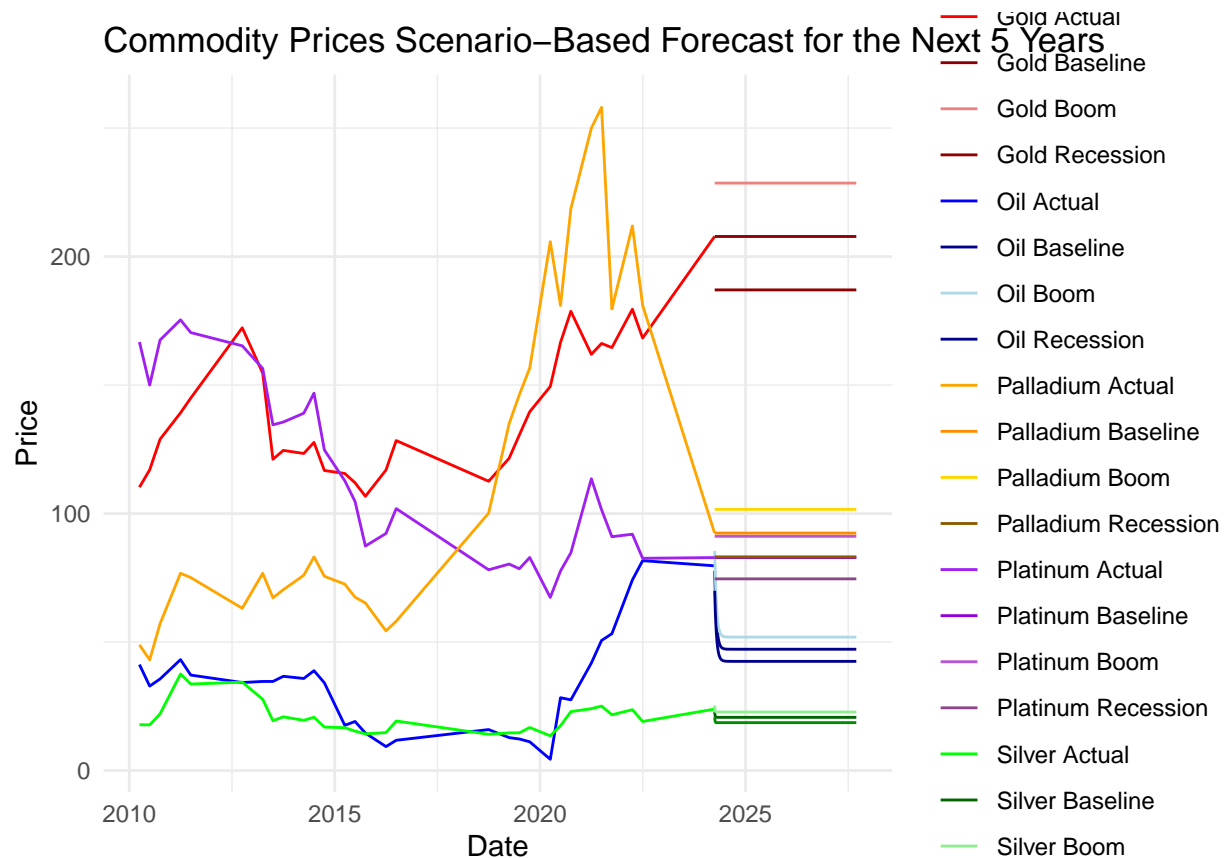
# Combine data for each commodity
oil_combined <- combine_data(data_clean, oil_forecast, "oil.close")
gold_combined <- combine_data(data_clean, gold_forecast, "gold.close")
silver_combined <- combine_data(data_clean, silver_forecast, "silver.close")
platinum_combined <- combine_data(data_clean, platinum_forecast, "platinum.close")
palladium_combined <- combine_data(data_clean, palladium_forecast, "palladium.close")

# Plot the actual and forecasted values for all commodities and S&P 500
ggplot() +
  geom_line(data = oil_combined, aes(x = date, y = actual, color = "Oil Actual")) +
  geom_line(data = oil_forecast, aes(x = date, y = forecast, color = "Oil Forecast")) +
  geom_line(data = gold_combined, aes(x = date, y = actual, color = "Gold Actual")) +
  geom_line(data = gold_forecast, aes(x = date, y = forecast, color = "Gold Forecast")) +
  geom_line(data = silver_combined, aes(x = date, y = actual, color = "Silver Actual")) +
  geom_line(data = silver_forecast, aes(x = date, y = forecast, color = "Silver Forecast")) +
  geom_line(data = platinum_combined, aes(x = date, y = actual, color = "Platinum Actual")) +
  geom_line(data = platinum_forecast, aes(x = date, y = forecast, color = "Platinum Forecast")) +
  geom_line(data = palladium_combined, aes(x = date, y = actual, color = "Palladium Actual")) +
  geom_line(data = palladium_forecast, aes(x = date, y = forecast, color = "Palladium Forecast")) +
  labs(title = "Commodity Prices Forecast for the Next 5 Years",
       x = "Date", y = "Price") +
  scale_color_manual(values = c(
    "Oil Actual" = "blue", "Oil Forecast" = "darkblue",
    "Gold Actual" = "red", "Gold Forecast" = "darkred",
    "Silver Actual" = "green", "Silver Forecast" = "darkgreen",
    "Platinum Actual" = "purple", "Platinum Forecast" = "violet",
    "Palladium Actual" = "orange", "Palladium Forecast" = "darkorange"
  )) +
  theme_minimal()
```

Commodity Prices Forecast for the Next 5 Years



```
## Warning: Removed 3780 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 3780 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 3780 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 3780 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 3780 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



To Test if my model was working the way I wanted it to I decided to graph the prices of the Commodities based on two scenarios. One which predicted a 10% increase in all the prices represented as a “boom” and another that represented a 10% decrease represented as a “recession”. This confirmed to me the the graphs were indeed working, but I decided to test out a different statistical model instead.

```
## Warning: package 'rugarch' is in use and will not be installed
## Installing package into 'C:/Users/regg0/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
## package 'xts' successfully unpacked and MD5 sums checked
## Warning: cannot remove prior installation of package 'xts'
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\regg0\AppData\Local\R\win-library\4.4\00LOCK\xts\libs\x64\xts.dll to
## C:\Users\regg0\AppData\Local\R\win-library\4.4\xts\libs\x64\xts.dll: Permission
## denied
## Warning: restored 'xts'
##
## The downloaded binary packages are in
## C:\Users\regg0\AppData\Local\Temp\RtmpsjLTzY\downloaded_packages
## Warning: package 'xts' was built under R version 4.4.2
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 4.4.2
##
```

```

## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
##
## ##### Warning from 'xts' package #####
## #
## # The dplyr lag() function breaks how base R's lag() function is supposed to #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or #
## # source() into this session won't work correctly. #
## #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop #
## # dplyr from breaking base R's lag() function. #
## #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #
## #
## #####
##
## Attaching package: 'xts'

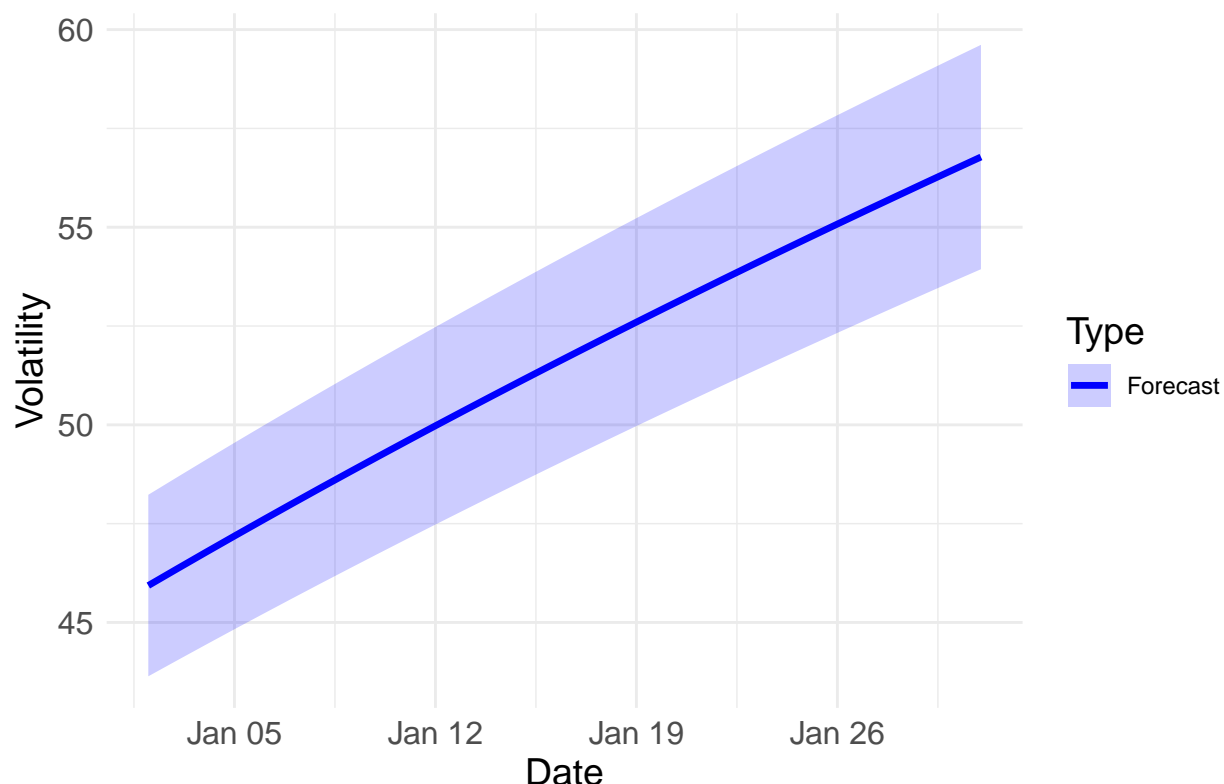
## The following objects are masked from 'package:dplyr':
##
##      first, last

## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :
## ugarchfit-->waring: using less than 100 data
## points for estimation

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```


Oil GARCH Volatility Forecast with Historical Context



Another model I used is called the GARCH model, which is used to estimate the volatility of financial returns. Typically, it is used in markets to model and forecast the volatility of different assets. Looking at the trend line it is clear that volatility is expected to steady increase over time. You can also see that the interval increase over time, generally as the interval becomes wider it represents greater uncertainty in the forecast. Therefore, this suggests that there is uncertainty in the future. This could represent greater risks in the market for traders and investors in oil related assets.

```
## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :  
## ugarchfit-->waring: using less than 100 data  
## points for estimation  
## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :  
## ugarchfit-->waring: using less than 100 data  
## points for estimation  
## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :  
## ugarchfit-->waring: using less than 100 data  
## points for estimation  
## Warning in .sgarchfit(spec = spec, data = data, out.sample = out.sample, :  
## ugarchfit-->waring: using less than 100 data  
## points for estimation
```

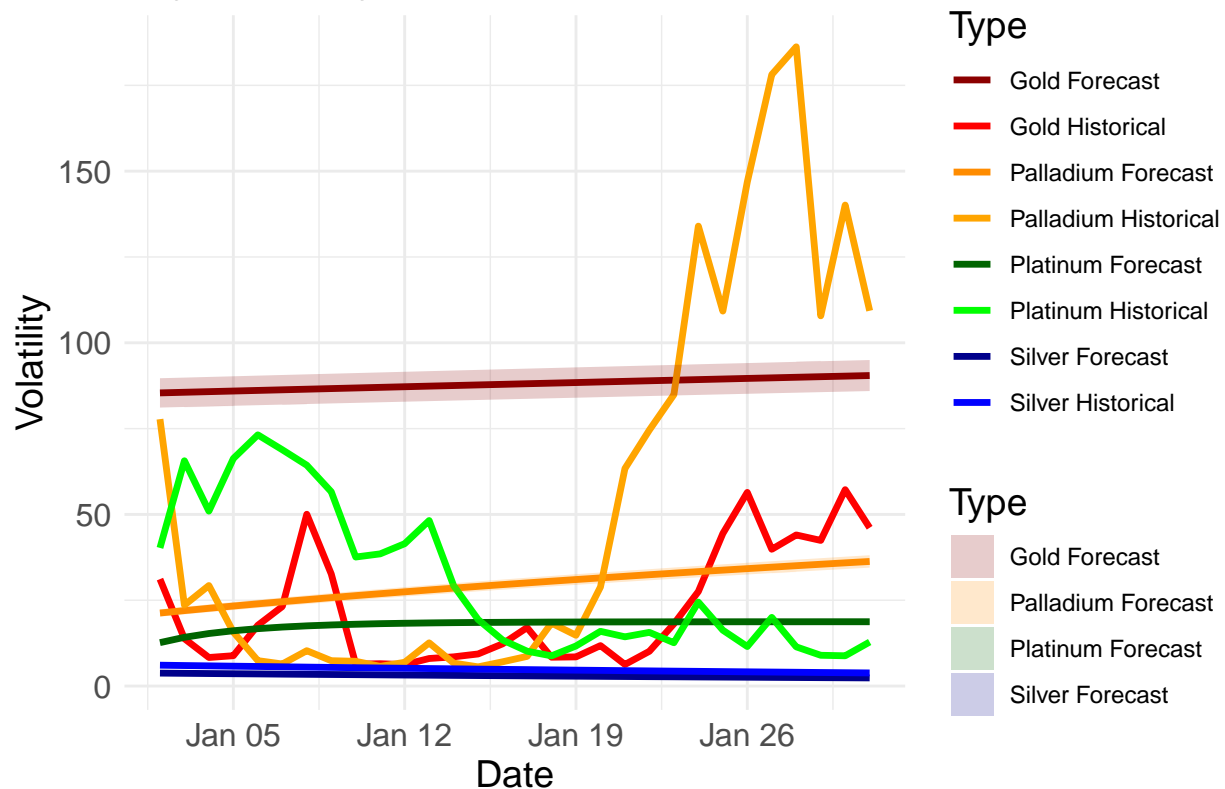
```
# Plot the combined data with confidence intervals  
ggplot(combined_data, aes(x = Date, y = Volatility, color = Type)) +  
  geom_line(size = 1.2) +  
  geom_ribbon(  
    data = combined_data %>% filter(str_detect(Type, "Forecast")),  
    aes(x = Date, ymin = Lower_CI, ymax = Upper_CI, fill = Type),  
    alpha = 0.2,
```

```

    inherit.aes = FALSE
  ) +
  labs(
    title = "Commodity Volatility Forecasts with Historical Context",
    x = "Date",
    y = "Volatility",
    color = "Type",
    fill = "Type"
  ) +
  scale_color_manual(values = c(
    "Gold Historical" = "red", "Gold Forecast" = "darkred",
    "Silver Historical" = "blue", "Silver Forecast" = "darkblue",
    "Platinum Historical" = "green", "Platinum Forecast" = "darkgreen",
    "Palladium Historical" = "orange", "Palladium Forecast" = "darkorange"
  )) +
  scale_fill_manual(values = c(
    "Gold Forecast" = "darkred",
    "Silver Forecast" = "darkblue",
    "Platinum Forecast" = "darkgreen",
    "Palladium Forecast" = "darkorange"
  )) +
  theme_minimal() +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold", size = 16),
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    legend.title = element_text(size = 14)
  )

```

Commodity Volatility Forecasts with Historical Context



Conclusion: Gold shows a relatively stable potential forecast compared to its historical trends, volatility remains steady. Palladium shows some extreme levels of fluctuations, but the forecast remains high and stable. Platinum displays historical volatility but the forecast suggest stability. Silver showed that both historical and forecasted volatility are minimal compared to the other commodities.

Important Observations

Palladium's historical volatility exhibits sharp spikes that are likely influenced by disruptions in the market or other factors. The forecast predicts this volatility to stabilize but stay around the 100 - 150 range.

Gold has the most stable historical and forecasted volatility, reflecting its reputation as a safe investment.

Platinum demonstrated moderate historical fluctuations, with its forecast suggesting a reduction in volatility in the future.

Silver has minimal volatility, with historical and forecasted levels being closely aligned, which could imply silver has a stable market.

Potential Insights

As expected, gold and silver appear to be less risky investments due to their stable volatility trends, make them safer options for risk-averse investors.

Platinum shows moderate volatility offering a medium-risk option for investors

Palladium's volatility signals opportunities for high-risk, high-reward strategies but will often require careful consideration of the market.

rsconnect::rpubsUpload()