Mueller Industries Analysis

Bupe Chanda

Abstract

I will be conducting some predictive quantitative analysis on the precious metals and industrial metals commodities markets. The main purpose of the analysis is to provide a bullish forecast for the American Industrials company Mueller Industries (NYSE: MLI).

To do this, I will start by proving a long term relationship between Aluminium Futures, Copper Futures, and MLI revenue over the last 9 years (2016 Q1 - 2024 Q4) using a Johansen Co-integration Test.

From there, I will test for both seasonality and cyclicality in the CBOE Volatility Index. Industrial metals countercyclical with the VIX because they rely on economic growth and perform best with healthy manufacturing, infrastructure and industrial trade sectors.

1. Libraries

```r
library(tidyquant)  # For financial data retrieval and analysis
library(quantmod) # For Yahoo Finance
library(urca)     # For the Johansen test & Phillips-Perron test
library(tseries)  # For other unit root tests
library(tseries)  # For the Augmented Dickey-Fuller test

library(readxl)     # For reading Excel files
library(ggplot2)    # For data visualization
library(dplyr)      # For data manipulation
library(tidyr)      # For reshaping data
library(forecast)   # For time series decomposition
library(zoo)        # For missing value imputation
```

2. Gather Data

I start by gathering the Mueller quarterly revenue Data from Q1 of 2016 and creating a data frame for it.

```r
file_path <- "Mueller Quarterly Revenue.xlsx"
rev_data  <- read_excel(file_path)

# Convert and sort the Date column
rev_data$Date <- as.Date(rev_data$Date)
rev_data      <- rev_data[order(rev_data$Date), ]

print(head(rev_data,5))
```

```
# A tibble: 5 x 2
  Date       Price
  <date>     <chr>
1 2015-12-31 472
2 2016-03-31 533
3 2016-06-30 544
4 2016-09-30 507
5 2016-12-31 472
```

I intend to use a single data frame to store all the data. Therefore, I created an extension of the revenue data frame to also include columns for aluminium futures and copper futures

```r
# Initialize empty columns for Aluminum and Copper
final_data <- data.frame(
  Date     = rev_data$Date,
  Revenue  = rev_data$Price,
  Aluminum = NA_real_,
  Copper   = NA_real_
)

print(head(final_data,5))
```

```
        Date Revenue Aluminum Copper
1 2015-12-31     472       NA     NA
2 2016-03-31     533       NA     NA
3 2016-06-30     544       NA     NA
4 2016-09-30     507       NA     NA
5 2016-12-31     472       NA     NA
```

The commodity futures price data was added using the Yahoo Finance API. It is important to note that not all 365 days in the calendar year are trading days. Therefore, a small date window was created so that if current_date is a weekend or holiday, the code would still get the last trading day's price. A range of 10 days was used in the code, but a smaller window could also work.

```r
for (i in seq_len(nrow(final_data))) {

  current_date <- final_data$Date[i]

# Create a window to to check for available dates
  from_date <- current_date - 10
  to_date   <- current_date

  # --- Aluminum Futures (ALI=F) ---
  # Use tryCatch to avoid errors if no data is returned
  ali_temp <- tryCatch(
    getSymbols("ALI=F", src = "yahoo", from = from_date, to = to_date,
               auto.assign = FALSE),
    error = function(e) NULL
  )

  if (!is.null(ali_temp) && nrow(ali_temp) > 0) {
    # Tail of the closing price is "on or before" current_date
    final_data$Aluminum[i] <- as.numeric(last(Cl(ali_temp)))
  }

  # --- Copper Futures (HG=F) ---
  hg_temp <- tryCatch(
    getSymbols("HG=F", src = "yahoo", from = from_date, to = to_date,
               auto.assign = FALSE),
    error = function(e) NULL
  )

  if (!is.null(hg_temp) && nrow(hg_temp) > 0) {
    final_data$Copper[i] <- as.numeric(last(Cl(hg_temp)))
  }
}
```

```r
print(head(final_data,5))
```

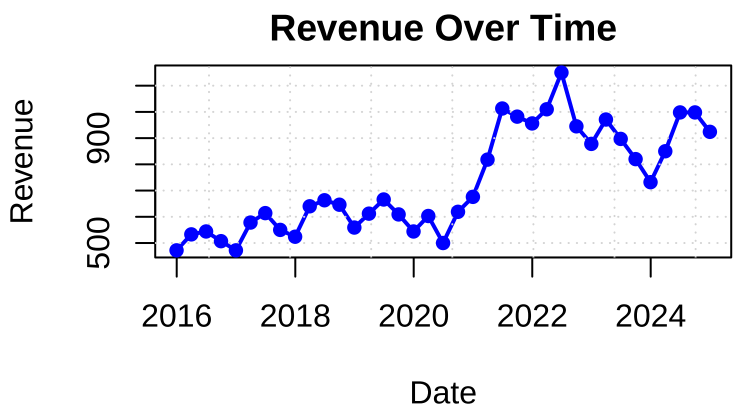
```
        Date Revenue Aluminum Copper
1 2015-12-31     472  1661.25 2.1370
2 2016-03-31     533  1590.00 2.1870
3 2016-06-30     544  1678.75 2.1835
4 2016-09-30     507  1726.25 2.1810
5 2016-12-31     472  1794.00 2.4975
```

3. Plot Data

While a formal test will be used for co-integration, I have still included the plots for visual inspection purposes. It is very clear even from here that the revenue and the futures do follow very similar long term trends.

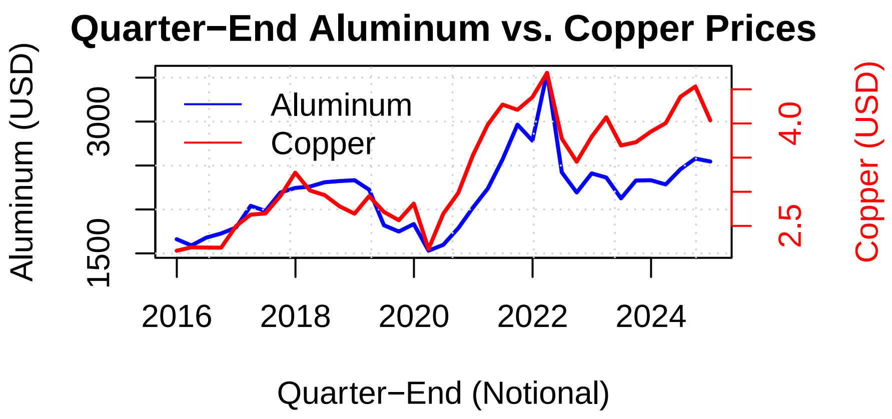
Revenue:

```r
par(mfrow=c(1,1), mar=c(2,3,2,3), pin=c(3,1))  # Adjust pin (plot dimensions)
plot(final_data$Date,
     final_data$Revenue,
     type="l",
     col="blue", lwd=2,
     xlab="Date", ylab="Revenue",
     main="Revenue Over Time")
grid()
points(final_data$Date, final_data$Revenue, col="blue", pch=16)
```



Industrial Metals Futures:

```r
par(mfrow=c(1,1), mar=c(2,3,2,3), pin=c(3,1))  # Adjust the figure size
plot(
  final_data$Date,
  final_data$Aluminum,
  type = "l",
  col  = "blue",
  lwd  = 2,
  xlab = "Quarter-End (Notional)",
  ylab = "Aluminum (USD)",
  main = "Quarter-End Aluminum vs. Copper Prices"
)
grid()

# Overlay Copper on the same chart, with axis on the right
par(new = TRUE)
plot(
  final_data$Date,
  final_data$Copper,
  type = "l",
  col  = "red",
  lwd  = 2,
  axes = FALSE,
  xlab = "",
  ylab = ""
)
axis(side = 4, col.axis = "red", col = "red")
mtext("Copper (USD)", side = 4, line = 3, col = "red")

legend(
  "topleft",
  legend = c("Aluminum", "Copper"),
  col    = c("blue", "red"),
  lty    = 1,
  bty    = "n"
)
```



4. Check for Stationarity

To formally test for stationarity, I used an Augmented-Dickey Fuller (ADF) test and a Phillips-Perron (PP) test. The PP test was included as well because it is more resistant to autocorrelation and heteroscedasticity. However, as will be seen with differencing, I am preferably looking for both tests to indicate the same result.

```r
# Convert data columns to time series objects
revenue_ts <- ts(final_data$Revenue, start = c(2016, 1), frequency = 4)
aluminum_ts <- ts(final_data$Aluminum, start = c(2016, 1), frequency = 4)
copper_ts <- ts(final_data$Copper, start = c(2016, 1), frequency = 4)
```

Augmented-Dickey Fuller:

```r
adf_revenue <- adf.test(na.omit(revenue_ts))
adf_aluminum <- adf.test(na.omit(aluminum_ts))
adf_copper <- adf.test(na.omit(copper_ts))

print(adf_revenue)
```

```
    Augmented Dickey-Fuller Test

data:  na.omit(revenue_ts)
Dickey-Fuller = -2.0433, Lag order = 3, p-value = 0.5564
alternative hypothesis: stationary
```

```
print(adf_aluminum)
```

```

	Augmented Dickey-Fuller Test

data:  na.omit(aluminum_ts)
Dickey-Fuller = -3.2233, Lag order = 3, p-value = 0.09858
alternative hypothesis: stationary
```

```
print(adf_copper)
```

```

	Augmented Dickey-Fuller Test

data:  na.omit(copper_ts)
Dickey-Fuller = -2.7533, Lag order = 3, p-value = 0.2795
alternative hypothesis: stationary
```

Phillips-Perron:

```
pp_revenue <- pp.test(na.omit(revenue_ts))
pp_aluminum <- pp.test(na.omit(aluminum_ts))
pp_copper <- pp.test(na.omit(copper_ts))

print(pp_revenue)
```

```

	Phillips-Perron Unit Root Test

data:  na.omit(revenue_ts)
Dickey-Fuller Z(alpha) = -11.302, Truncation lag parameter = 3, p-value
= 0.4102
alternative hypothesis: stationary
```

```
print(pp_aluminum)
```

```

	Phillips-Perron Unit Root Test
```

```
data:  na.omit(aluminum_ts)
Dickey-Fuller Z(alpha) = -13.825, Truncation lag parameter = 3, p-value
= 0.248
alternative hypothesis: stationary
```

```r
print(pp_copper)
```

```

    Phillips-Perron Unit Root Test

data:  na.omit(copper_ts)
Dickey-Fuller Z(alpha) = -12.503, Truncation lag parameter = 3, p-value
= 0.333
alternative hypothesis: stationary
```

5. Co-Integration Order

All the tests had p-values above 0.05, indicating that all the datasets were non-stationary. However, the Johansen test is valid when all variable are integrated of the same order (i.e. it becomes stationary after differencing the same number of times). While first-differencing was initially used, the resulting ADF and PP tests provided different results. Therefore, this was followed up with a second differencing for a final integrated order of I(2).

p-value $< 0.05 \rightarrow$ Stationary

p-value $> 0.05 \rightarrow$ Non-stationary.

Prepare the data:

```r
# Remove the NAs and make the data numeric
revenue_clean <- as.numeric(na.omit(final_data$Revenue))
aluminum_clean <- as.numeric(na.omit(final_data$Aluminum))
copper_clean <- as.numeric(na.omit(final_data$Copper))

# Convert cleaned data to time series
revenue_ts <- ts(revenue_clean, start = c(2016, 1), frequency = 4)
aluminum_ts <- ts(aluminum_clean, start = c(2016, 1), frequency = 4)
copper_ts <- ts(copper_clean, start = c(2016, 1), frequency = 4)

# Second differencing to make data stationary
```

```r
revenue_diff <- diff(revenue_ts, differences = 2)
aluminum_diff <- diff(aluminum_ts, differences = 2)
copper_diff <- diff(copper_ts, differences = 2)
```

Augmented-Dickey Fuller:

```r
adf_revenue_diff <- adf.test(revenue_diff)
adf_aluminum_diff <- adf.test(aluminum_diff)
adf_copper_diff <- adf.test(copper_diff)

print(adf_revenue_diff)
```

```
	Augmented Dickey-Fuller Test

data:  revenue_diff
Dickey-Fuller = -4.7369, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

```r
print(adf_aluminum_diff)
```

```
	Augmented Dickey-Fuller Test

data:  aluminum_diff
Dickey-Fuller = -3.4233, Lag order = 3, p-value = 0.07035
alternative hypothesis: stationary
```

```r
print(adf_copper_diff)
```

```
	Augmented Dickey-Fuller Test

data:  copper_diff
Dickey-Fuller = -3.8722, Lag order = 3, p-value = 0.02679
alternative hypothesis: stationary
```

Phillips-Perron:

```r
pp_revenue_diff <- pp.test(revenue_diff)
pp_aluminum_diff <- pp.test(aluminum_diff)
pp_copper_diff <- pp.test(copper_diff)

print(pp_revenue_diff)
```

```


	Phillips-Perron Unit Root Test

data:  revenue_diff
Dickey-Fuller Z(alpha) = -33.087, Truncation lag parameter = 3, p-value
= 0.01
alternative hypothesis: stationary
```

```r
print(pp_aluminum_diff)
```

```


	Phillips-Perron Unit Root Test

data:  aluminum_diff
Dickey-Fuller Z(alpha) = -48.363, Truncation lag parameter = 3, p-value
= 0.01
alternative hypothesis: stationary
```

```r
print(pp_copper_diff)
```

```


	Phillips-Perron Unit Root Test

data:  copper_diff
Dickey-Fuller Z(alpha) = -39.074, Truncation lag parameter = 3, p-value
= 0.01
alternative hypothesis: stationary
```

Once the second differencing has been applied to the three time series, a Johansen test can be carried out to identify any long term relationships between Mueller Industries' revenue, aluminum prices, and copper prices.

6. Johansen Co-Integration Test

```r
# Combine the second-differenced data into a matrix
data_matrix <- cbind(revenue_diff, aluminum_diff, copper_diff)

# Run the Johansen test
johansen_test <- ca.jo(data_matrix, type = "trace", ecdet = "none", K = 2)
print(summary(johansen_test))
```

```
######################
# Johansen-Procedure #
######################

Test type: trace statistic , with linear trend

Eigenvalues (lambda):
[1] 0.7132175 0.6958475 0.5502170

Values of teststatistic and critical values of test:

           test 10pct  5pct  1pct
r <= 2 |  26.37  6.50  8.18 11.65
r <= 1 |  65.64 15.66 17.95 23.52
r = 0  | 106.86 28.71 31.52 37.22

Eigenvectors, normalised to first column:
(These are the cointegration relations)

                 revenue_diff.l2 aluminum_diff.l2 copper_diff.l2
revenue_diff.l2        1.0000000         1.000000       1.000000
aluminum_diff.l2       0.1875802         1.534309      -0.478058
copper_diff.l2      -373.0220328      -294.742519     611.295655

Weights W:
(This is the loading matrix)

                revenue_diff.l2 aluminum_diff.l2 copper_diff.l2
revenue_diff.d     -1.405226143     -0.004788084   -0.318439693
aluminum_diff.d     1.021833400     -1.831785673   -0.038808734
copper_diff.d       0.002329713     -0.001394147   -0.002547716
```

7. Co-Integration Results Analysis

Interpreting the Johansen test involves comparing the test statistic against critical values. If the test statistic exceeds the critical value at a 5% significance level, the null hypothesis of no cointegration is rejected. According to the values of the test statistics and critical values, at rank $r = 0$, $r \leq 1$ and $r \leq 2$, we reject the null hypothesis. Therefore, there exists at least two co-integrating relationships between the revenue, and the the Aluminium and Copper Futures.

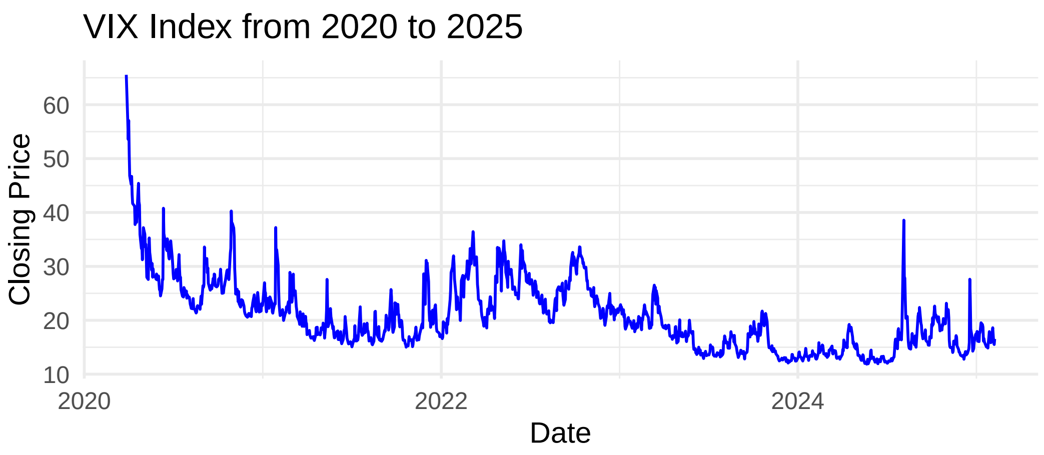
The matrix of coefficients show very different scales for the values, however, this is likely due to the revenue being in the hundreds of millions, the alumium futures being in the thousands and the copper futures consistently remaining below 10 USD.

The most important result here is that there do exist co-integrating relationships between the variables, indicating a stable long term relationship.

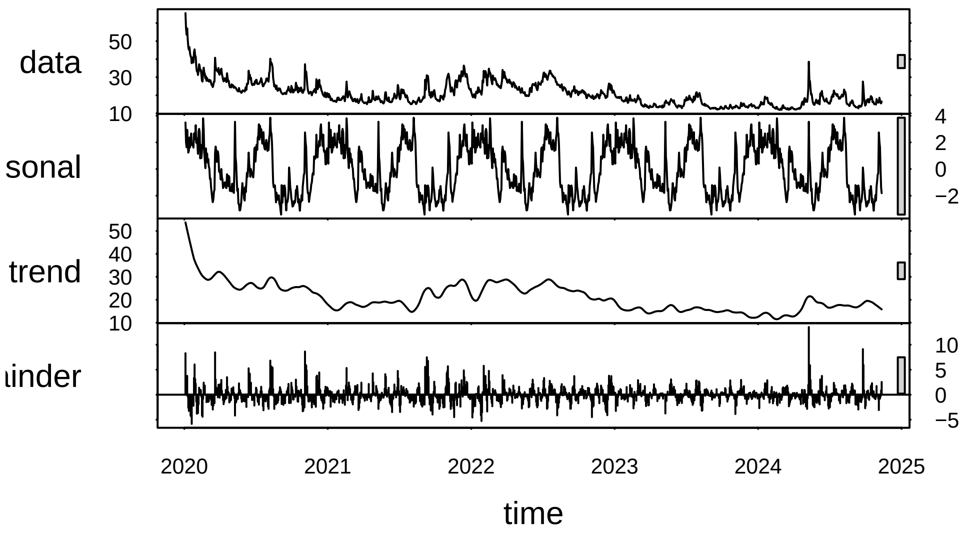
8. VIX Time Series Decomposition

From here, I used a simple time series decomposition to observe a long term down trend in the Chicago Board Options Exchange's CBOE Volatility Index (^VIX).

```r
# Download data for the required date range
vix_data <- tq_get("^VIX", from = "2020-03-27", to = "2025-02-10")

# Handle missing values by linear interpolation
vix_data$close <- na.approx(vix_data$close, na.rm = FALSE)

# Remove any remaining NA values after interpolation
vix_data <- vix_data %>% drop_na()

# Plot the VIX time series
ggplot(vix_data, aes(x = date, y = close)) +
  geom_line(color = "blue") +
  labs(title = "VIX Index from 2020 to 2025",
       x = "Date",
       y = "Closing Price") +
  theme_minimal() +
  theme(plot.margin = margin(5, 5, 5, 5),  # Adjust margins
        aspect.ratio = 1/3)  # Adjust aspect ratio
```



```
# Convert data to time series format (252 trading days per year)
vix_ts <- ts(vix_data$close, start = c(2020, 3), frequency = 252)

# Perform STL decomposition with a shorter trend window (30 trading days)
vix_decomp <- stl(vix_ts, s.window = "periodic", t.window = 30)

# Plot the decomposition
par(las = 1)
plot(vix_decomp)
```



It is evident that the time series has a downward trend over the last 5, probably contributing in some part to the increased price and performance of the company and industrial metals futures. This stable macroeconomic environment indicated by the sustained downward trajectory of the VIX leads me towards a bullish sentiment for the commodities market and by extension the Mueller Industries Company.