Trading Strategy: TMA in Metal Commodities Markets

CA01 - Investment Fundamentals, Group Project

Group 8 - Benedikt Jaletzke, Stanislas Markov, Mark Negodyuk,
Kateryna Tarasova, Olivia Zhang, Tom Tian

12/10/2020

Abstract

Triangular Moving Averages in Metal Commodities Markets for Consistent Long Short Returns

Libraries

TO DO:

- Formulas in LaTeX Format
- Buy/Sell
- Propa Write Up, Innit
- Footnotes
- Figure out how to encode this thing…
  - http://eng-web1.eng.famu.fsu.edu/~dommelen/l2h/errors.html#gen

Official Checklist:

- Introduction
- Economic Rationale
- Data (Sources)
- Method Used
  - Technical Indicator
  - Market
- Main Findings
- Future Potential Analyses
  - Metal Cmpanies in Combination with SUE Earnings
  - Mixing in Other Asset Classes (Agriculture, Energy etc.)
- Appendix

NOTES

See how the portfolio would do with S&P instead of Industrial Metals

Or Agriculture

Do that once ALL the calculations are finished. That way it should be a copy/paste job with fresh data.

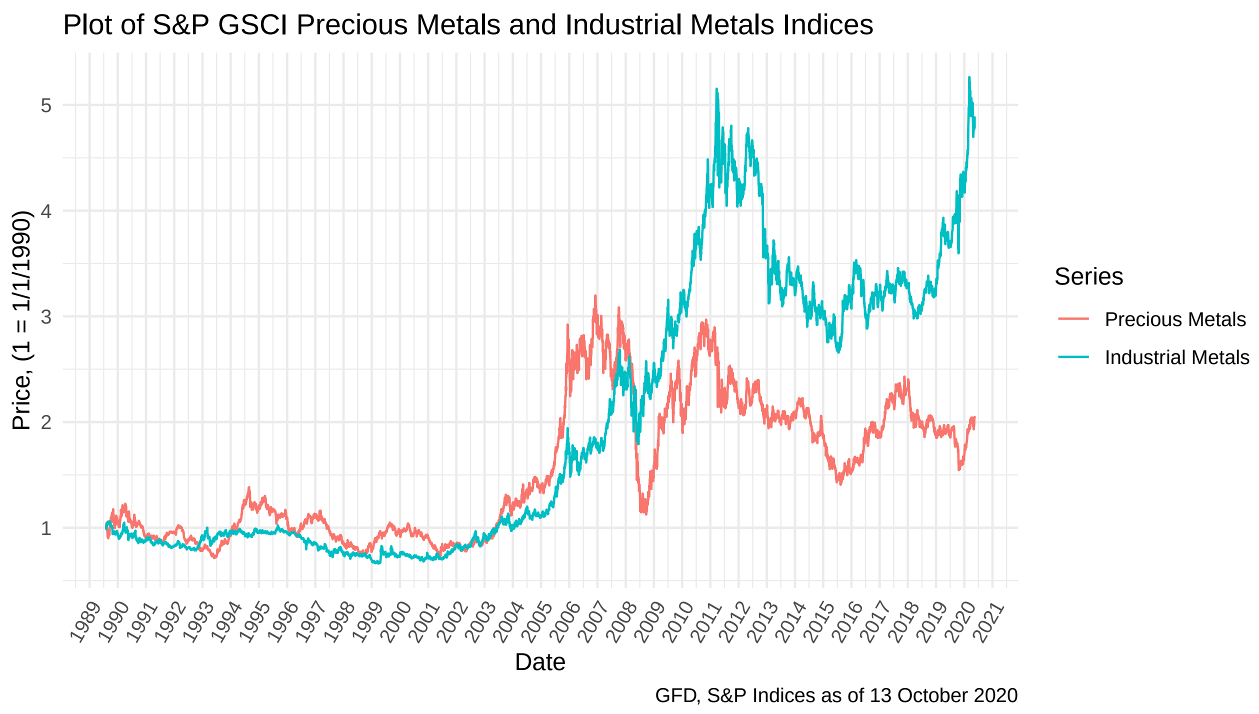
Data Import

```r
# Data Import and Cleaning
import_data <-  function(source){
  x <- read.csv(source, skip = 2)
  x %>%
  mutate(date = as.Date(x$Date, format = "%m/%d/%Y")) %>%
  filter(date > as.Date("1989-12-31")) %>%
  subset(data = x, select = -c(Date, Open, High, Low)) %>%
  rename(close = Close,
         ticker = Ticker) %>%
  mutate(index = close/close[1])
}

# I would love to implement the GFD API in order to automatically pull this data in from their site. Ma
precious_1 <- import_data("data/sp_precmet.csv")
industrial_1 <- import_data("data/sp_indmet.csv")

# Combining both indices into one frame, cleaning it up a little
combined_indices <- full_join(industrial_1, precious_1, by = "date") %>%
  rename(precious_metals = index.y,
         pm_close = close.y,
         ind_close = close.x,
         industrial_metals = index.x) %>%
  subset(select = -c(ticker.x, ticker.y)) %>%
  head(-2)
```

```r
combined_one <-  combined_indices %>%
  subset(select = -c(ind_close, pm_close)) %>%
  gather(index, values, precious_metals:industrial_metals)

ggplot(combined_one) +
  geom_line(aes(x=date, y=values, color = index)) +
  labs(title = "Plot of S&P GSCI Precious Metals and Industrial Metals Indices",
    y = "Price, (1 = 1/1/1990)",
    x= "Date",
    caption = "GFD, S&P Indices as of 13 October 2020") +
  scale_x_date(date_breaks = "12 months", date_labels =  "%Y") +
  theme_minimal() +
  theme(axis.text.x=element_text(angle=60, hjust=1)) +
  scale_color_brewer(palette = "Viridis") +
  scale_color_discrete(name = "Series", labels= c("Precious Metals", "Industrial Metals"))
```

```
## Warning in pal_name(palette, type): Unknown palette Viridis
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



Introduction

The idea for this group project was to come up with a strategy that either exploited an arbitrage opportunity in the market, or represents another form of hedge fund strategy. Our group decided to focus on using technical indicators. Within this area, we specifically chose moving averages, and as a suitable market for their application settled on metal commodities.

Economic Rationale

Data

Papers

- S&P GSCI Precious Metals Index
- S&P GSCI Industrial Metals Index

Method Used

Portfolio Allocation

In order to ideally work within the metals markets, the portfolio will consist ocmpletely of two already existing indices that are calculated on a daily basis, the S&P Precious Metals Index, consisting of [TBC] and the S&P Industrial Metals, containing [TBC]. By using these two indices as our baseline instead of attempting to construct our own, we can create a two-security portfolio with different weights to find the best possible results. As for the choice of allocations, we tried different combination weights of the indices.

To construct the efficient portfolio, we had to calculate the return and variance of the two separate indices.

```
# Return - Variance Calculation for the Separate Metals Indices
```

```r
ind_price <- na.approx(zoo(combined_indices$ind_close, combined_indices$date))
prec_price <- na.approx(zoo(combined_indices$pm_close, combined_indices$date))

rets_industry <- Return.calculate(xts(ind_price),  method = "discrete")
names(rets_industry) <- "return_ind"

rets_prec <- Return.calculate(xts(prec_price),  method = "discrete")
names(rets_prec) <- "return_prec"

combined_indices <- cbind(combined_indices, rets_industry, rets_prec)

stats_solo <- combined_indices %>%
  summarise(mean_ind = mean(return_ind[-1]), mean_prec = mean(return_prec[-1]),
            sd_ind = sd(return_ind[-1]), sd_prec = sd(return_prec[-1]))
```

```r
# Efficient Portfolio

# Statistics for the Portfolio: Mean, StDev, Correlation
calc_efficient <- combined_indices %>%
  summarise(mean_ind = mean(return_ind[-1]), mean_prec = mean(return_prec[-1]),
            sd_ind = sd(return_ind[-1]), sd_prec = sd(return_prec[-1]),
            corr_ind = cor(return_ind[-1],return_prec[-1]))

# Function to calculate the Weighting of Industrial Metals in the Final Portfolio
sh_r_max <- function(x){
  (x*calc_efficient$mean_ind+(1-x)*calc_efficient$mean_prec)/sqrt(x^2*calc_efficient$sd_ind^2+(1-x)^2*ca
}

# Calculating optimal weights for industrial index
opt_port <- optimize(f = sh_r_max, lower = -1,upper = 1, maximum = TRUE)

# assigning weights to the two indices
weight_industrial <- opt_port$objective
weight_precious <- 1-weight_industrial

# creating (theoretically) efficient combination
efficient_portfolio <- combined_indices %>%
  mutate(efficient = weight_industrial * industrial_metals + weight_precious * precious_metals) %>%
  subset(select = c(date, efficient))
```

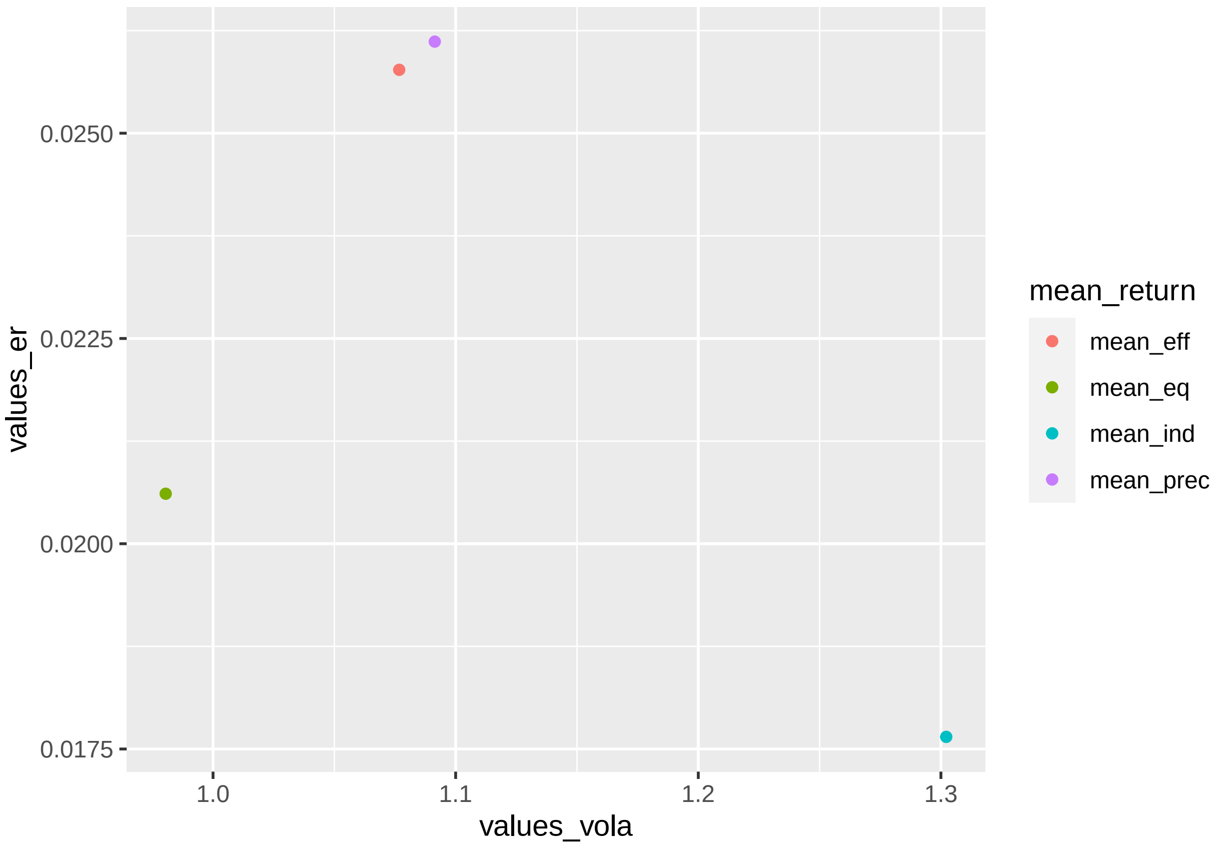
```r
# 50/50 Portfolio
# Weights
eq_w_pm <- 0.5
eq_w_ind <- 1-eq_w_pm

# calculate index
eq_w_ind <- combined_indices %>%
  mutate(eq_weight = eq_w_pm * precious_metals + eq_w_ind * industrial_metals) %>%
  subset(select = c(date, eq_weight))
```

```r
# Joining the Portfolios
# Add Equal Weights to Individual Indices
indices_equal <- right_join(eq_w_ind, combined_indices, by = "date") %>%
 subset(select = -c(ind_close, pm_close )) %>%
  subset(select = -c(return_ind, return_prec))

# Add Efficient Portfolio
portfolios <- left_join(indices_equal, efficient_portfolio, by = "date")
```

```r
# Ask Stas what this does
equal_price <- na.approx(zoo(portfolios$eq_weight, portfolios$date))
rets_eq <- Return.calculate(xts(equal_price),  method = "discrete")
names(rets_eq) <- "return_eq"

efficient_price <- na.approx(zoo(portfolios$efficient, portfolios$date))
rets_eff <- Return.calculate(xts(efficient_price),  method = "discrete")
names(rets_eff) <- "return_eff"

# Statistics for the Equal Weight Portfolio
portfolios <- cbind(portfolios, rets_eq)
stats_eq <- portfolios %>%
  summarise(mean_eq = mean(return_eq[-1]),
            sd_eq = sd(return_eq[-1]))

# Statistics for the Efficient Portfolio
portfolios <- cbind(portfolios, rets_eff)
stats_eff <- portfolios %>%
  summarise(mean_eff = mean(return_eff[-1]),
            sd_eff = sd(return_eff[-1]))
```

```r
# Sharpe Ratios
# Assumed Risk Free Rate, based on current 10-Year US Treasuries
risk_free <- 0.0076

# Sharpe Ratio Function ????????
sharpe <- function(dataset, mean, sd){
  (dataset$mean - (risk_free / 365)) / dataset$sd
}

# Sharpe Ratios for the Individual Indices
sharpe_ind <- sharpe(stats_solo, mean_ind, sd_ind)
sharpe_prec <- sharpe(dataset = stats_solo, mean = mean_prec, sd = sd_prec)
# FOR SOME F***ING REASON THE FORMULA ABOVE DOES NOT WORK FOR THE INDIVIDUAL SHARPE RATIOS. I COULD THR
sharpe_efficient <- sharpe(stats_eff, mean_eff, sd_eff)
sharpe_equal <-sharpe(stats_eq, mean_eq, sd_eq)
# But it does work for these two clowns...

sharpe_ind <- (stats_solo$mean_ind - risk_free / 365)/stats_solo$sd_ind
sharpe_prec <- (stats_solo$mean_prec - (risk_free / 365)) / stats_solo$sd_prec

er_sd <- cbind(stats_solo, stats_eq, stats_eff) %>%
  gather(mean_return, values_er, c(mean_ind, mean_prec, mean_eff, mean_eq)) %>%
  gather(volatility, values_vola, c(sd_ind, sd_prec, sd_eff, sd_eq)) %>%
  rownames_to_column("return") %>%
  slice(1, 6, 11, 16) %>%
  subset(select = -return) %>%
  mutate(values_vola = values_vola * 100) %>%
  mutate(values_er = values_er * 100)
```

```r
ggplot(er_sd) + geom_point(aes(x = values_vola, y = values_er, color = mean_return))
```

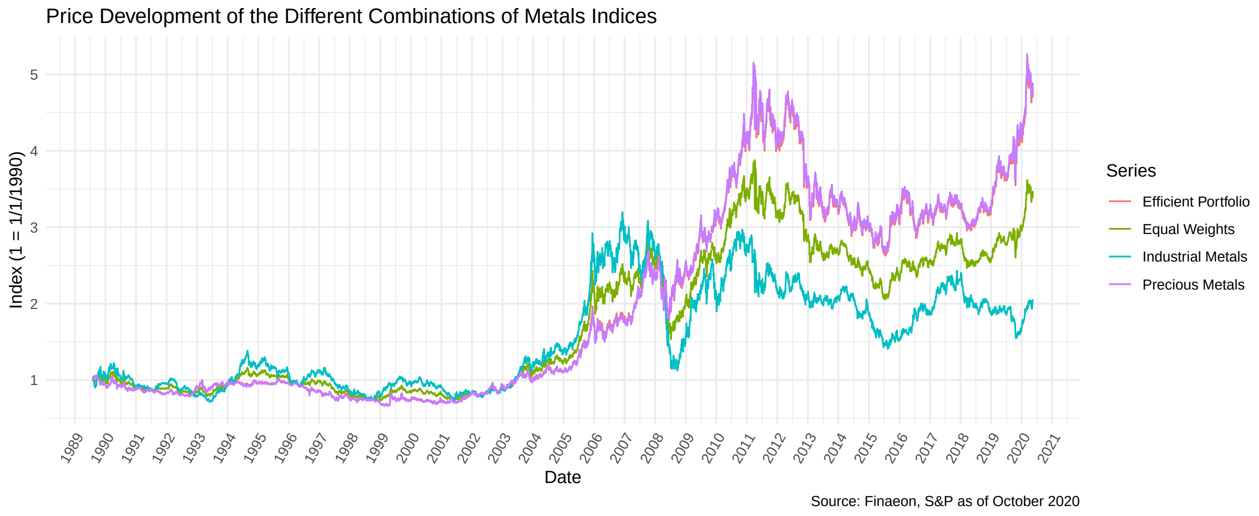


```r
# combining the four portfolios into one
portfolios_overview <- left_join(indices_equal, efficient_portfolio, by = "date") %>%
  gather(index, values, eq_weight:efficient)
```

Portfolio Graph

```
## Warning in pal_name(palette, type): Unknown palette Viridis
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



The graph shows clearly that the most efficient portfolio is one in which the heaviest weighting is placed on the precious metals component of the portfolio, which is not surprising given it's outperformance of the industrial metals class, especially over the last ten years. For a further analysis, this raises the obvious question of whether a more efficient portfolio could be constructed using different indices/components that would diversify the returns away from this sector.

Technical Indicator

From https://uc-r.github.io/ts_moving_averages

The concept of simple moving averages can be extended to taking moving averages of moving averages. This technique is often employed with an even number of data points so that the final product is symmetric around each point. An even-numbered moving average is unbalanced, and for our purposes, the unbalancing will be in favor of more recent observations. For example, to calculate a 4-MA, the equation is as follows:

```
# LaTeX formula for TMA (wont be inside R Block)
# ^yt=(yt-1+yt+yt+1+yt+2)/4
```

To make the moving average symmetric (and therefore more accurate), we then take a 2-MA of the 4-MA to create a 2 x 4-MA. For the 2-MA step, we average the current and previous moving averages, thus resulting in an overall estimate of:

Finding appropriate Modifiers for the TMA

The main challenge lies in finding the appropriate time periods for the TMA. As it is an average of averages, we have to consider which of these will give us the best return. The best TMA will be one that fits the trend well enough, but can still react to changes in the underlying assets quickly enough to be useful.

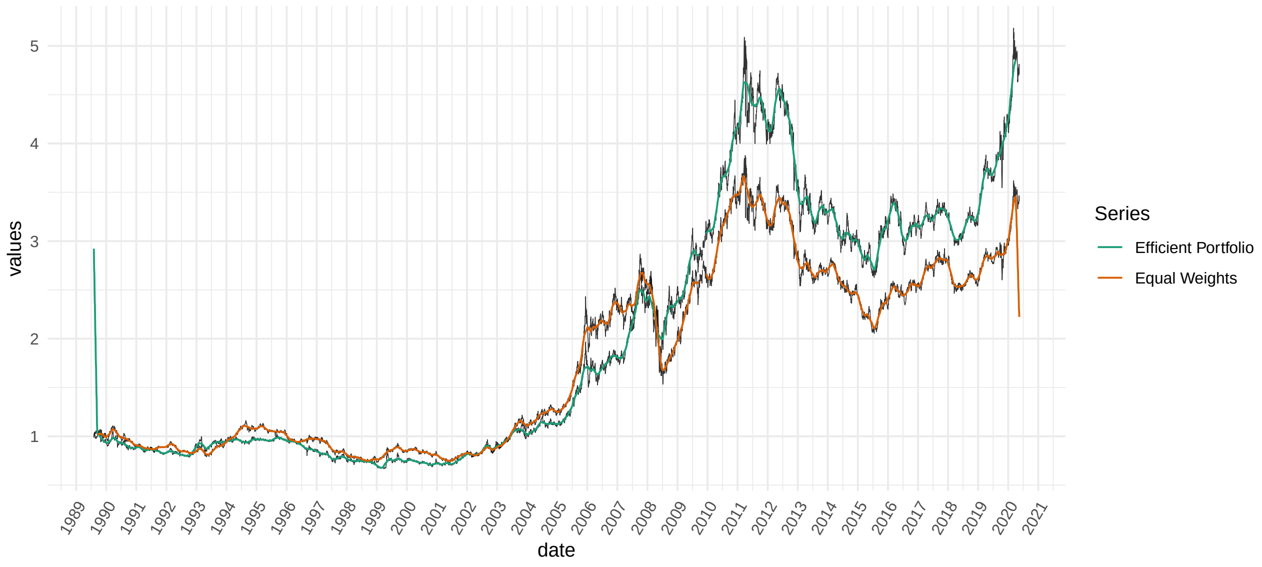
To compute a basic TMA, we can use the ma() function and take a look at the output, in this case with a 60 day compound.

```
# rolling averages. Useful for later calculations
# k = number of days used in average

portfolios_tma <- portfolios_overview %>%
  mutate(ma4 = ma(values, order = 60, centre = TRUE))

pf_subset <- portfolios_tma %>%
  #filter(index == "eq_weight" || index == "efficient")
  filter(index %in% c("eq_weight", "efficient"))
```

```r
# Have to find a way to remove the extreme values at the beginning and end of the series

# 60 Day TMAs over the daily data for the two composite portfolios
ggplot(pf_subset) +
  geom_line(aes(x = date, y=values, group = index),
            alpha = 0.8, fill = "black", size = 0.2) +
  geom_line(aes(x = date, y=ma4, color = index)) +
  scale_color_brewer(palette="Dark2", name = "Series", labels = c("Efficient Portfolio", "Equal Weights
  scale_x_date(date_breaks = "12 months", date_labels =  "%Y") +
  theme_minimal() +
  theme(axis.text.x=element_text(angle=60, hjust=1))
```

```
## Warning: Ignoring unknown parameters: fill
```

```
## Warning: Removed 60 row(s) containing missing values (geom_path).
```



In order to see the efficiency of any portfolio, we have to implement buy and sell signals. In our case, as described earlier, the idea would be to buy when the TMA is passed upwards, and to sell & short when the line crosses below. To do this, we can use the code suggested on the QuantInst Blog

This would give us a return of [TBA]

Using the optimize function we employed to get the portfolio weights earlier, we can attempt to maximise the output of our TMA line on both portfolios.

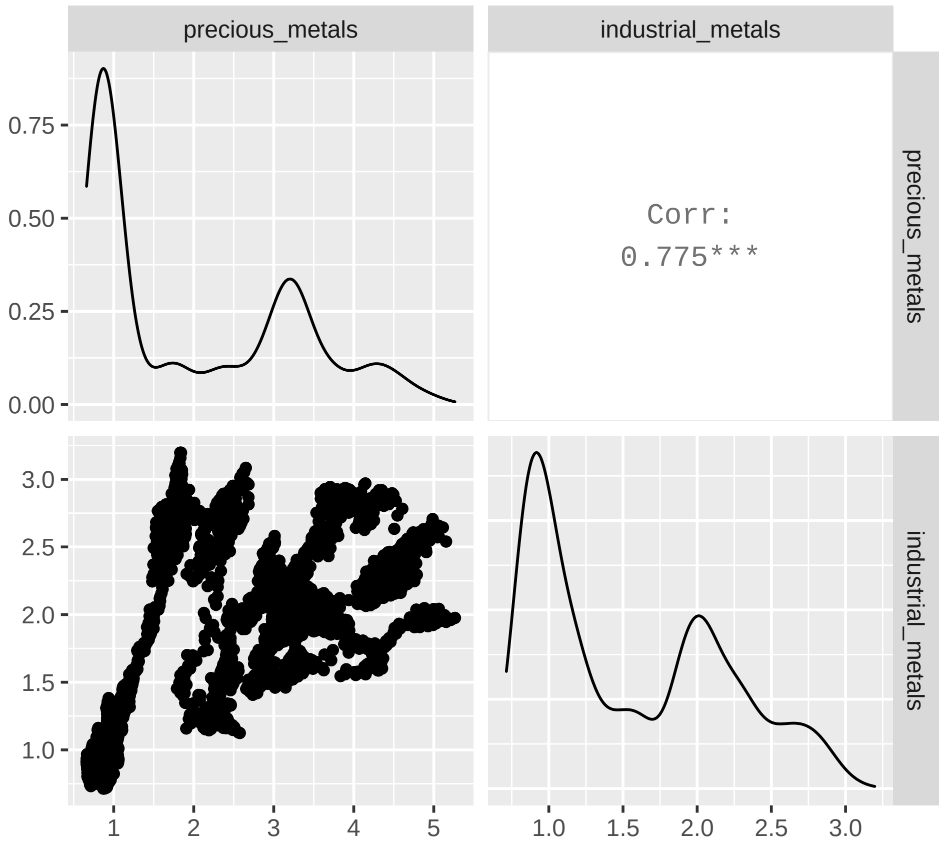
Market Choice

```r
combined_indices %>%
  subset(select = c(precious_metals, industrial_metals)) %>%
  ggpairs(combined_indices)
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



As the above corelation graph shows, and intuition would confirm, there is an extreme correlation between the different classes of metals. This is despite hte fact that gold, which makes up a large part of the precious metals index, is traded less for its value sas a metal from an utilitarian perspective and more because of it's economic value as a crisis/hedging asset.

Results / Main Findings

Appendix

Importing and Filtering Equities Indices for Comparison Purposes