

Alternative Timeline: Estimating non-COVID Gold Prices

Erick Guevara, Greane Ramos, Richard Collie

12-07-2023

1. Introduction

The unfortunate event of the COVID-19 Pandemic affected the entire world. Alongside the great number of deaths and suffering that occurred, the global economies also found themselves in poor market conditions. Due to quarantine and stay-at-home orders, the manufacturing of goods slowed down tremendously, travel and shipping decreased significantly, and overall returns in stocks was low.

Specifically in March 2020, the entire market crashed. The DOW (DJIA) had fallen roughly 26% over the course of four trading days [Mieszko]. In parallel with this index, virtually every other stock had taken a hard hit. This was due to many factors, the primary source being the high volatility and low market sentiment as a result of the actions taken by the government in order to mitigate the infections of the virus.

The question that this report intends to answer is “How should an asset have performed based on historical ‘pre-COVID’ data observations if COVID never occurred” . The asset chosen for this study is gold. Gold was chosen due to its uniqueness in being a historical hedge against inflation. Typically, gold is less volatile when compared to other stocks which makes it efficient in regards to the interest of time it will take to aggregate and analyze data during the COVID period.

In order to perform this analysis, Vector Auto Regressive (VAR) will be modeled to estimate prices. Once completed, out-of-sample-simulations will be conducted in order to simulate the most likely scenario that may have been based on historical “pre-COVID” data. Interest and inflation rate data will be used as control variables to allow more accurate measurement when comparing the actual observed COVID gold prices.

1.1 Researching and Aggregating Data

It took us a while to look for the needed data, but we came across files with a lot of nulls, incorrect time frames, download limits, and pay walls. Eventually, we just used yahoo to extract the historical gold prices but at an initial time range constraint from 2004. The `quantmod` package was used to aggregate inflation and interest rates data by calling it from DGS10 and CPIAUCSL databases.

```
#packages needed
```

```
library(dplyr)
library(ggplot2)
library(tseries)
library(xts)
library(quantmod)
library(tidyquant)
library(vars)
library(forecast)
library(zoo)
```

```
library(lubridate)

#Calling datasets

interest10_rate <- tq_get("DGS10", get = "economic.data", from = "2004-09-01", to="2020-03-30")
inflation10_rate <- tq_get("CPIAUCSL", get= "economic.data", from = "2004-08-11", to="2020-03-30")

gold <- read.csv("C:/Users/justd/Desktop/Gold2004_2023.csv", header=T)
gold$Date <- as.Date.character(gold$Date, format= "%m/%d/%Y")
```

1.2 Data Cleaning

From the gold data set we selected only date and price. Afterwards we removed any special characters that would mess with transforming since R isn't too friendly with special characters.

```
gold_data <- gold %>%
  dplyr::select(Date, Price)

gold_data$Price <- as.numeric(gsub(',', '', gold_data$Price))

interest10_rate <- interest10_rate %>%
  dplyr::select(-symbol)
inflation10_rate <- inflation10_rate %>%
  dplyr::select(-symbol)
```

1.3 Linear Interpolation and Timeframe Change

In order to deal with null values in the data sets we would first determine what method worked best for producing or estimating values to minimize distortion and over-fitting with noise in the forecasting model. We found that linear interpolation works best for filling missing values. Since, NA are small gaps between observations in different dates this method would allow for proper estimation and avoid smoothing when filling in the gaps. This works by linear regressing past and post observations to fit in the linear trend.

```
interest10_rate$price <- na.approx(interest10_rate$price)
```

Once, short gaps were filled the pre-COVID period was selected to be transformed into monthly data to match the inflation rate structure and length for time series analysis. This was done by first creating a sequence of dates that we will use to estimate then average the price of every month and saving it under the first day of each month.

```
start_date <- as.Date("2004-12-01")
end_date <- as.Date("2020-03-01")

date_seq <- seq(from = start_date, to = end_date, by = "day")

gold_ts <- data.frame(Date = date_seq) %>%
  left_join(gold_data, by = 'Date') %>%
  mutate(Price = na.approx(Price, na.rm = F))
```

```

gold_ts <- gold_ts %>%
  mutate(date = floor_date(Date, "month")) %>%
  group_by(date) %>%
  summarize(Price = mean(Price, na.rm = T)) %>%
  ungroup()

interest10_rate <- data.frame(date = date_seq) %>%
  left_join(interest10_rate, by = 'date') %>%
  mutate(price = na.approx(price, na.rm = F))

interest10_rate <- interest10_rate %>%
  mutate(date = floor_date(date, "month")) %>%
  group_by(date) %>%
  summarize(price = mean(price, na.rm = T)) %>%
  ungroup()

```

2. Time Series Analysis

We needed a model that factors in relationships of exploratory variables that influence gold prices in order to create an accurate estimate of prices that achieves an impact towards the prediction. We chose vector auto regressive model which would model in the factors of control variables to estimate gold prices. Auto regressive only works well with data that has a stochastic behavior, a constant mean and variance across the time period. Using auto correlation function and augmented Dickey-Fuller test we can determine if it's stationary. The observations were transformed in order to achieve a stationary process that would allow for capturing most of the residual patterns to achieve accuracy. To properly highlight the optimal lag intervals for each set, we applied Schwert formula to decide the maximum number of lag values and used alkaline information criterion to select best model for each time interval applied.

2.1. Time Series Approach

The sets was first turned into an extensible time series frame which would allow for viewing seasonality, autocorrelation, structural breaks, and noticeable trends that would be needed to get removed. We first can see that gold, interest, and inflation rates are non-stationary showing increasing trends. We confirmed for gold auto correlation by conducting an Augmented Dickey-Fuller Test which comes back with a p-value of .6444 suggesting we fail to reject null hypothesis. The p-value for inflation rate is .1993 and interest rate is .3993 which also implies both process are non-stationary since p-value is higher than the significant level of 0.05 failing to reject the null hypothesis.

```

#Gold Prices
gold_xts <- xts(gold_ts$Price, order.by=gold_ts$date)
gold_xts <- na.omit(gold_xts)

adf_goldxts <- adf.test(gold_xts)

#Inflation Rate
inflation10_xts <- xts(inflation10_rate$price, order.by=inflation10_rate$date)
adf_inflationxts <- adf.test(inflation10_xts)

```

```

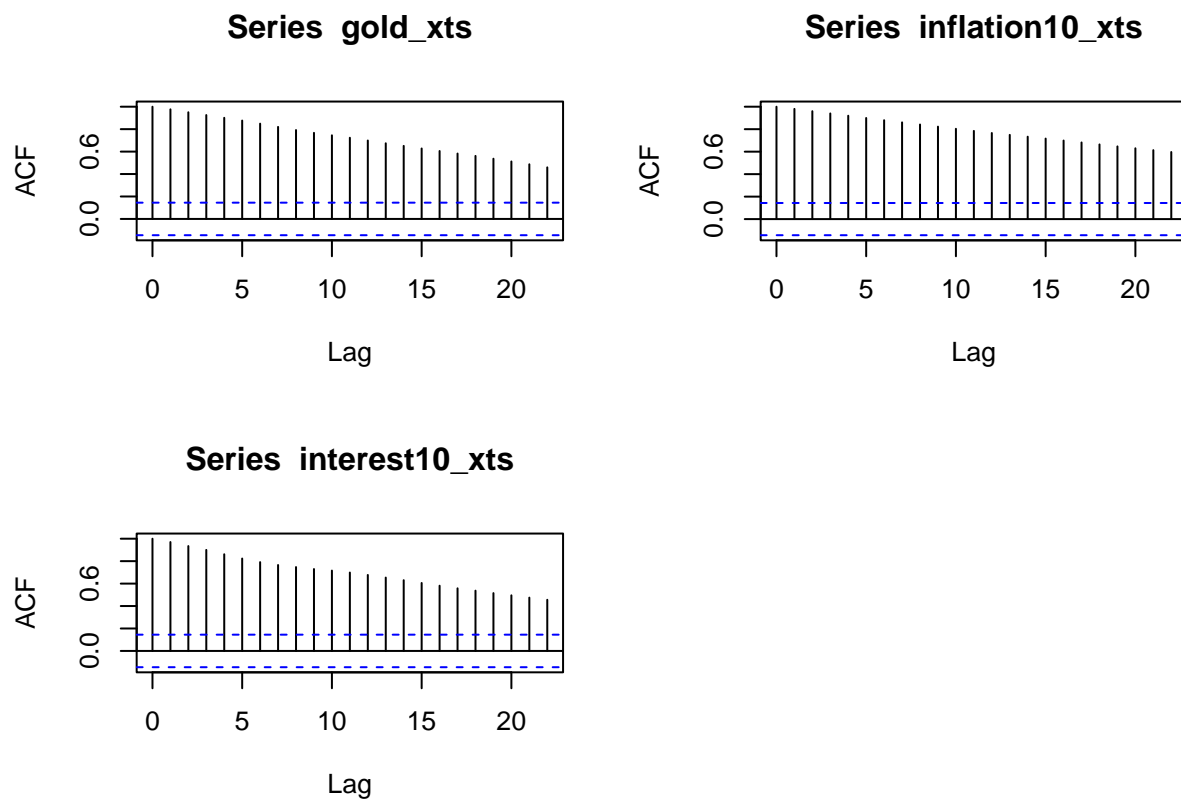
#Interest Rate
interest10_xts <- xts(interest10_rate$price, order.by=interest10_rate$date)
interest10_xts <- na.omit(interest10_xts)

adf_interestxts <- adf.test(interest10_xts)

#Visualizations
#Visualizations
par(mfrow=c(2,2))
acf(gold_xts)
acf(inflation10_xts)
acf(interest10_xts)

par(mfrow=c(2,2))

```



```

plot(gold_xts)
plot(inflation10_xts)
plot(interest10_xts)

#Augmented Dickey Fuller Tests
adf_goldxts

##
## Augmented Dickey-Fuller Test

```

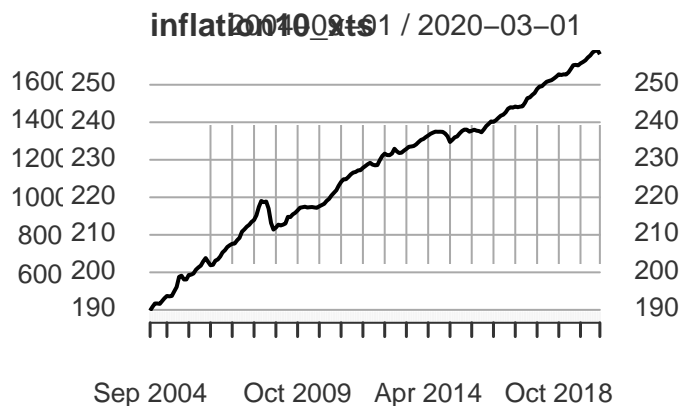
```
##
## data: gold_xts
## Dickey-Fuller = -1.8369, Lag order = 5, p-value = 0.6444
## alternative hypothesis: stationary
```

```
adf_inflationxts
```

```
##
## Augmented Dickey-Fuller Test
##
## data: inflation10_xts
## Dickey-Fuller = -2.9011, Lag order = 5, p-value = 0.1993
## alternative hypothesis: stationary
```

```
adf_interestxts
```

```
##
## Augmented Dickey-Fuller Test
##
## data: interest10_xts
## Dickey-Fuller = -2.4231, Lag order = 5, p-value = 0.3993
## alternative hypothesis: stationary
```



2.2. Stationary Process

To remove the systematic structure from the time series sets a first order difference transformation was introduced to achieve the stationary process. This was done by subtracting the previous observation from each current value in the series.

```
#Gold First Order Difference Transformation
gold_xts_diff <- diff(gold_xts)
gold_xts_diff <- na.omit(gold_xts_diff)

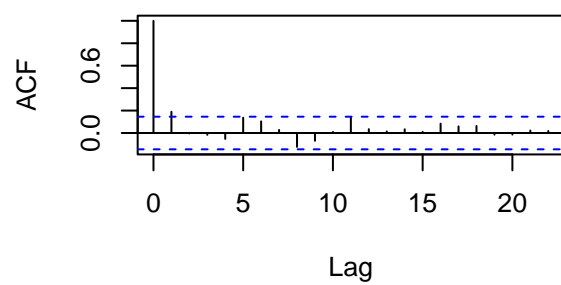
#Interest Rate First Order Difference Transformation
interest_diff <- diff(interest10_xts)
interest_diff <- na.omit(interest_diff)

#Inflation Rate First Order Difference Transformation
inflation_diff <- diff(inflation10_xts)
inflation_diff <- na.omit(inflation_diff)

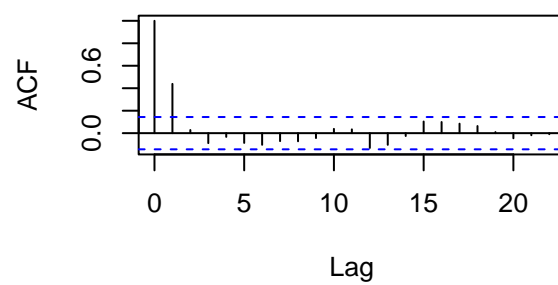
#Visualizations
par(mfrow=c(2,2))
acf(gold_xts_diff)
acf(inflation_diff)
acf(interest_diff)

par(mfrow=c(2,2))
```

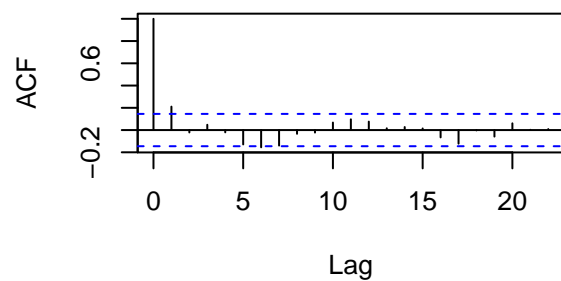
Series gold_xts_diff



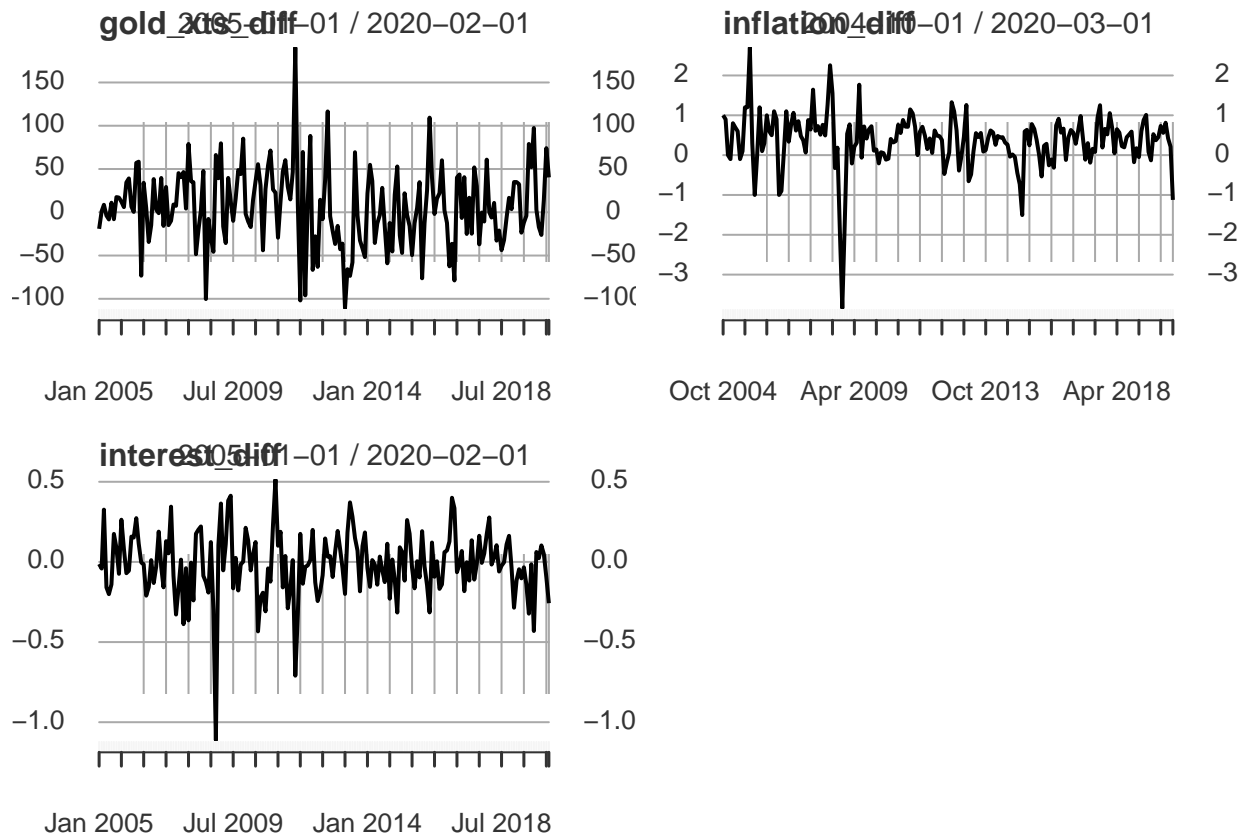
Series inflation_diff



Series interest_diff



```
plot(gold_xts_diff)
plot(inflation_diff)
plot(interest_diff)
```



All p-values were 0.01 which is the cap from R-studio, since its below 0.05 it suggests that the process is now stationary by rejecting the null hypothesis. This is also illustrated in the time series plot as the trend seems stable across the time frame and the autocorrelation plot depicts the intervals to be within the bounds across each lag values. The problem with the non-stationary process has been removed as shown in the figure above.

2.3. Schwert Formula and Optimal Lag Slection

To select the optimal lag values, we first calculate the max lag value to be able to define the ceiling and choose the number of efficient lag values by using AIC model selection. This is to make sure not to over fit the model with unnecessary amount of noise. This checks one by one until max lag and then outputs the number of efficient time intervals that highlight most of the residuals.

```
#Gold Prices
ts_length <- length(gold_xts) #gets the length of observation of ts data
max_lags = floor(12 * (ts_length/100) ^ 0.25)
aic_values = rep(NA, max_lags + 1) #The aic values is the the return of max_lags

for (k in 0:max_lags) {
  tryCatch({
    fit_gold = Arima(gold_xts_diff, order = c(k, 0, 0), include.constant = F)
    aic_values[k + 1] = fit_gold$aic,
    error = function(e) {
      cat("Error: ", e$message, "\n")
      aic_values[k + 1] = NA
    }
  })
}
```



```

}

valid_aic <- !is.na(aic_values)
if (any(valid_aic)) {
  optimal_lags = which.min(aic_values[valid_aic]) - 1
  optimal_lags
} else {
  cat("No valid AIC values were found.")
}

## [1] 1

adf_gold <- adf.test(gold_xts_diff, k = optimal_lags)
cat("\nLags: ", k, "ADF Test Statistics: ", adf_gold$statistic, "P-value: ", adf_gold$p.value, "\n")

##
## Lags: 13 ADF Test Statistics: -8.776544 P-value: 0.01

adf_gold

##
## Augmented Dickey-Fuller Test
##
## data: gold_xts_diff
## Dickey-Fuller = -8.7765, Lag order = 1, p-value = 0.01
## alternative hypothesis: stationary

#Inflation Rate

#Based on the best score for AIC values it looks for the optimal lag
ts_length_inflation <- length(inflation10_xts)
max_lags_infl = floor(12 * (ts_length_inflation/100) ^ 0.25)
aic_values_infl = rep(NA, max_lags_infl + 1)

for (p in 0:max_lags_infl) {
  tryCatch({
    fit_inflation = Arima(inflation_diff, order = c(p, 0, 0), include.constant = F)
    aic_values_infl[p + 1] = fit_inflation$aic,

    error_infl = function(e) {
      aic_values_infl[p + 1] = NA
    })
  })
}

valid_aic_infl <- !is.na(aic_values_infl)
if (any(valid_aic_infl)) {
  optimal_lags_infl = which.min(aic_values_infl[valid_aic_infl]) - 1
  optimal_lags_infl
} else {
  cat("No valid AIC values were found.")
}

```

```
## [1] 4
```

```
inflation_adf <- adf.test(inflation_diff, k= optimal_lags_infl)
inflation_adf
```

```
##
## Augmented Dickey-Fuller Test
##
## data: inflation_diff
## Dickey-Fuller = -6.3995, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

```
#Interest Rate
```

```
ts_length_interest <- length(interest10_xts)
max_lags_inter = floor(12 * (ts_length_interest/100) ^ 0.25)
aic_values_inter = rep(NA, max_lags_inter + 1)

for (p in 0:max_lags_inter) {
  tryCatch({
    fit_interest = Arima(interest_diff, order = c(p, 0, 0), include.constant = F)
    aic_values_inter[p + 1] = fit_interest$aic,

    error_int = function(e) {
      aic_values_inter[p + 1] = NA
    })
  })
}

#Based on the best score for AIC values it looks for the optimal lag
valid_aic_inter <- !is.na(aic_values_inter)
if (any(valid_aic_inter)) {
  optimal_lags_inter = which.min(aic_values_inter[valid_aic_inter]) - 1
  optimal_lags_inter
} else {
  cat("No valid AIC values were found.")
}
```

```
## [1] 1
```

```
interest_adf <- adf.test(interest_diff, k = optimal_lags_inter)
```

It took sometime for the AIC model selection algorithm to run due to being limited, computationally. A way to overcome this slow calculation is to use a parallel computing package which would split my system's cores to independently work on the selections which would be much faster. Another method can be used is cloud services specifically Data bricks to run much faster on virtual machines. Ultimately, it did finish running and the chose optimal lag numbers as illustrated above. 1 being that the whole structure itself starting from the first lag value is optimal enough.

3. Vector Autoregressive Model (VAR)

Once the extensible time series data sets became stationary with optimal lag values. A merged time series was created by all having the same time frame from 2005 to 2020 and converted to being a time series.

The vector auto regressive model will incorporate variables that will help influence the estimation of the y dependent variable. Since, we read that inflation and interest rates does have an influence towards the behavior of gold prices. It was decided to use those variables to measure in, the association towards gold prices and be able to have a prediction during the COVID timeline.

After the sets were merged and tested for a combined stationary process then the optimal lag values for the merged set was found by using the same Schwert's formula and AIC model selection methods. However, we used the tseries package VARselect() function to choose the optimal values, a feature not available for extensible time series data.

```
gold_diff_sub <- gold_xts_diff['2005-01-01/2020-02-01']
inflation_diff_sub <- inflation_diff['2005-01-01/2020-02-01']
interest_diff_sub <- interest_diff['2005-01-01/2020-02-01']

merged.xts.data <- merge(gold_diff_sub, inflation_diff_sub, interest_diff_sub)
summary(merged.xts.data)
```

##	Index	gold_diff_sub	inflation_diff_sub	interest_diff_sub
##	Min. :2005-01-01	Min. :-110.814	Min. :-3.84200	Min. :-1.11193
##	1st Qu.:2008-10-08	1st Qu.: -18.648	1st Qu.: 0.07925	1st Qu.: -0.13024
##	Median :2012-07-16	Median : 4.549	Median : 0.44950	Median : -0.01443
##	Mean :2012-07-16	Mean : 6.358	Mean : 0.37114	Mean : -0.01494
##	3rd Qu.:2016-04-23	3rd Qu.: 35.849	3rd Qu.: 0.69250	3rd Qu.: 0.11019
##	Max. :2020-02-01	Max. : 190.313	Max. : 2.70000	Max. : 0.51369

```
merged.ts.data <- as.ts(merged.xts.data) #changed xts to ts data type for VAR

#Choosing optimal lag order
merged.ts.maxlag<- length(merged.ts.data) #gets the length of observations of ts data
merged.lag.max <- floor(12 * (merged.ts.maxlag/100) ^ 0.25)

lag_selection <- VARselect(merged.ts.data, lag.max = merged.lag.max, type= 'both')
lag_selection$selection

## AIC(n) HQ(n) SC(n) FPE(n)
##      2      1      1      2

optimal_lag.merged <- lag_selection$selection["AIC(n)"]

VAR.model <- VAR(merged.ts.data, p = optimal_lag.merged, type= 'both')
VAR.model$varresult

## $gold_diff_sub
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##      gold_diff_sub.l1 inflation_diff_sub.l1 interest_diff_sub.l1
##              0.20851             -0.50496                11.13141
##      gold_diff_sub.l2 inflation_diff_sub.l2 interest_diff_sub.l2
##             -0.07543              -4.57689               -29.98584
```

```
##          const          trend
##      11.53704         -0.04584
##
##
## $inflation_diff_sub
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##      gold_diff_sub.l1  inflation_diff_sub.l1  interest_diff_sub.l1
##           0.0006996           0.5101140           0.2866337
##      gold_diff_sub.l2  inflation_diff_sub.l2  interest_diff_sub.l2
##           0.0009597           -0.2245372           -0.0594967
##           const          trend
##           0.2875238          -0.0003313
##
##
## $interest_diff_sub
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
##      gold_diff_sub.l1  inflation_diff_sub.l1  interest_diff_sub.l1
##          -0.0002718           0.1091838           0.1468650
##      gold_diff_sub.l2  inflation_diff_sub.l2  interest_diff_sub.l2
##          -0.0003318           -0.0522281           -0.0831953
##           const          trend
##          -0.0286307          -0.0000337
```

4. VAR Model Diagnostics

Before moving on with utilizing the VAR model to perform out-of-sample estimations. Diagnostics were needed to make sure of statistically significant results, if most of the pattern was captured, no white noise, variable comparison on impacts from shocks, and seeing the amount of variation it affects gold price. Common methods and tests were applied to produce and verify these diagnostics.

4.1 Residual Testing

First, we checked to see if the variance remain constant over time. This will tell us if it is homeostatic a characteristic to have for the model. The results below states that the p-value is lower than 0.05 which means to reject null hypothesis and suggests the residuals are homeostatic across time.

```
arch.model <- arch.test(VAR.model)
arch.model

##
## ARCH (multivariate)
##
## data:  Residuals of VAR object VAR.model
## Chi-squared = 263.29, df = 180, p-value = 5.098e-05
```

4.2. Portmanteau (Ljung-Box Test) - White Noise Testing

Portmanteau or Ljung-Box test will check for white noise in the residual basically confirming for random walk, seeing if there was a series of correlations (auto correlations), basically making sure residuals don't correlate with each other, to have a stochastic pattern behavior. Ultimately, to determine if the model captured the time series dynamics adequately for accurate estimations.

```
serial.test(VAR.model)

##
## Portmanteau Test (asymptotic)
##
## data: Residuals of VAR object VAR.model
## Chi-squared = 99.194, df = 126, p-value = 0.9627
```

Based on the results above, from running the test. The p-value shows to be higher than 0.05, indicating that we should fail to reject the null hypothesis which suggests the model has auto correlations, In this case since we failed to reject it implies the model sufficiently captured the pattern of the time series.

4.3. Impulse Response Function

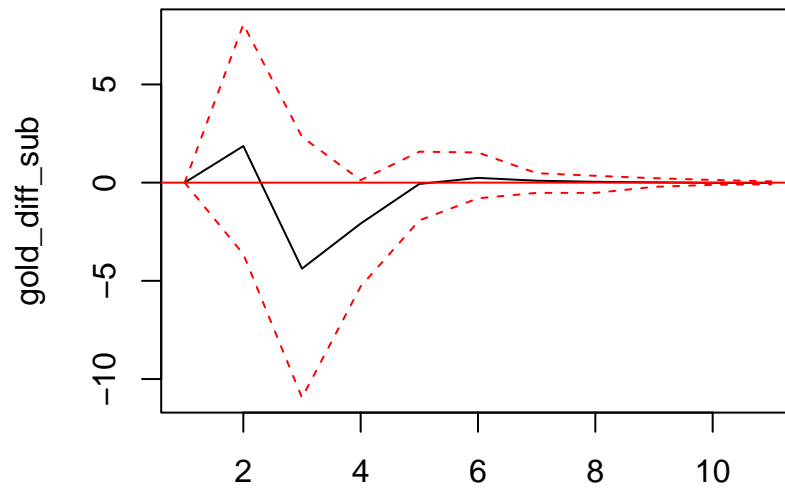
Impulse response function analysis measures the change in affect that the control variables imposes on gold prices when they're affected themselves. This will tell us the importance of shock influence that interest and inflation variables has on gold prices. The figures below illustrate the response magnitude of the impact on each time lags representing months. The significance depends on the confidence intervals where it's movement depicts the importance of the black line's trajectory across the maximum lag.

```
irf_var.inflation <- irf(VAR.model, impulse = "inflation_diff_sub", response = "gold_diff_sub", n.ahead = 10)

irf_var.interest <- irf(VAR.model, impulse = "interest_diff_sub", response = "gold_diff_sub", n.ahead = 10)

plot(irf_var.interest)
```

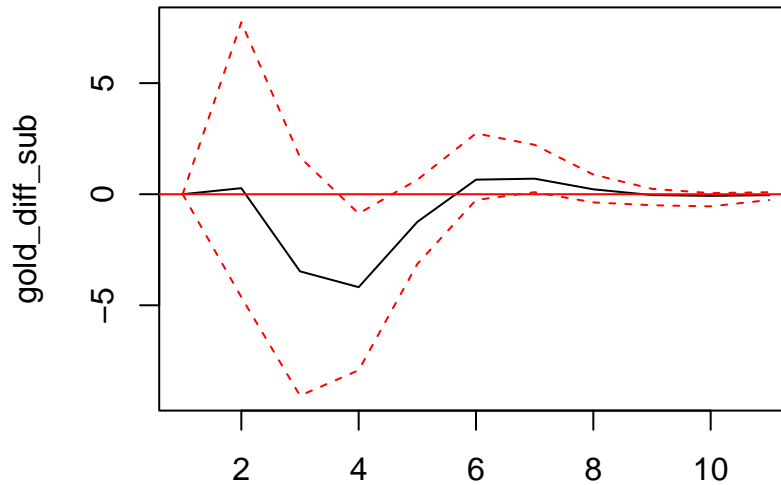
Orthogonal Impulse Response from interest_diff_sub



95 % Bootstrap CI, 100 runs

```
plot(irf_var.inflation)
```

Orthogonal Impulse Response from inflation_diff_sub



95 % Bootstrap CI, 100 runs

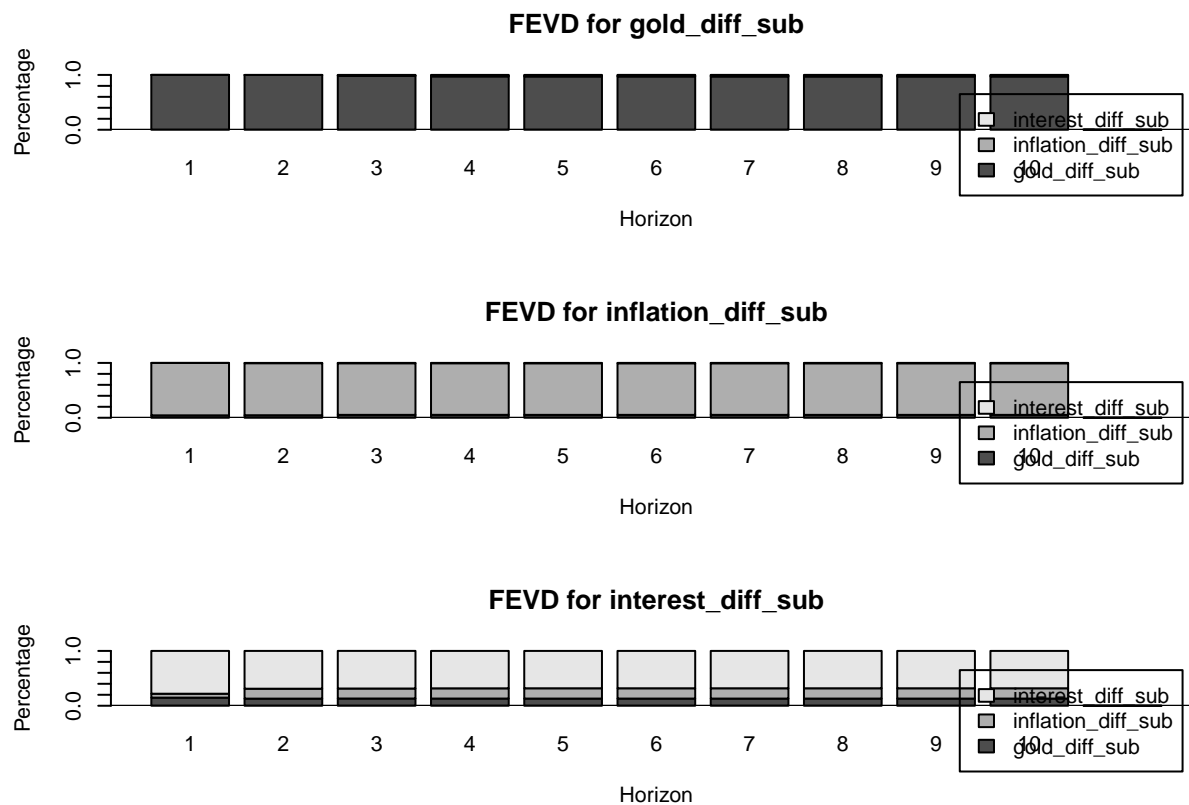
The interest rate IRF plot reveals the traditional relationship between gold prices and interest rates, which validates the method used. The initial negative response showcase during that time interval, interest rates rose which caused a dampen in gold prices. This is witnessed in the real world when interest rates increase the non-yielding assets like gold will likely decrease. However, the shocks didn't last very long as the structure begins to stabilize. Shown by the converging confidence intervals that includes zero after lag value 5 meaning it's not statistically significant anymore.

The inflation rate IRF plot reveal an same initial trend from interest rates. This suggests the magnitude of change from the impact of inflation has negatively affected gold prices. Again, highlighting the relationship between inflation and gold prices. However, due to the CIs including zero, it shows that it's not statistically significant and the magnitude begins to level off.

4.4. Forecast Error Variance Decomposition (FEVD)

Last analysis conducted was to measure the shock from the variation of both variable to gold prices is the forecast error variance decomposition. This should tell us how much of the variation affected the shock of gold. The figure below looks into 10 lag values meaning 10 months ahead. The plots show that gold prices for the most part is affected by its own shock. Showing us that most of the shock is explained by most of the variation of gold. This is the same for inflation and interest revealing that both variables wouldn't increase the prediction value since gold prices reacts the same from both variables. Uncovering on just how simple the predictions will turn out since it only has itself and a second variable that merely impacts gold.

```
fevd.VAR <- fevd(VAR.model, n_ahead = 10)
plot(fevd.VAR)
```



5. Out-of-Sample Simulations

We can now move on to the final step to obtain the price estimations during the COVID timeline is to conduct a simulation from the measured variable relationships model. The calculations will be produced as the change in the difference between one period to the next starting at the last price value observed. Since the results is the change, we would need to un difference the predicted values to bring it back to the same scale as the actual prices by getting the cumulative sum.

```
n.ahead.forecast = 22 #About 2 years

VAR.gold_estimations <- predict(VAR.model, n.ahead = n.ahead.forecast)

#extracting the forecasting values

gold.forecast_values <- VAR.gold_estimations$fcst$gold_diff_sub[, "fcst"]
```

6. Comparative Analysis

I extracted, imputed nulls, and changed it to monthly data for the actual observed gold prices that will be plotted against the estimated values. Combined all vectors to create a full comparative analysis on estimated price, actual price during COVID, and post COVID prices.


```

start_covid = as.Date("2020-02-01")
end_covid = as.Date("2021-12-01")

date_seq_covid <- seq(from = start_covid, to = end_covid, by = "day")

gold_covid <- data.frame(Date = date_seq_covid) %>%
  left_join(gold_data, by = 'Date') %>%
  mutate(Price = na.approx(Price, na.rm = F))

gold_covid <- gold_covid %>%
  mutate(date = floor_date(Date, "month")) %>%
  group_by(date) %>%
  summarize(Price = mean(Price, na.rm = T)) %>%
  ungroup()
#Post-Covid time frame

post_covid.start = as.Date("2022-01-01")
post_covid.end = as.Date("2023-11-01")

date.seq_post.covid <- seq(from = post_covid.start, to = post_covid.end, by = "day")

gold_post.covid <- data.frame(Date = date.seq_post.covid) %>%
  left_join(gold_data, by = 'Date') %>%
  mutate(Price = na.approx(Price, na.rm = F))

gold_post.covid <- gold_post.covid %>%
  mutate(date = floor_date(Date, 'month')) %>%
  group_by(date) %>%
  summarize(Price = mean(Price, na.rm = T)) %>%
  ungroup()

gold_post.covid <- gold_post.covid %>%
  rename(Time = date)

last_obs_value <- tail(gold_covid[gold_covid$date <= start_covid, "Price"], 1)
forecasted_levels <- cumsum(c(last_obs_value, gold.forecast_values))

comparative_analysis <- data.frame(
  Time = seq(from = as.Date("2020-02-01"), by = 'month', length.out = length(forecasted_levels)),
  Forecasted = forecasted_levels,
  Actual = gold_covid$Price
)

#comparative_analysis <- comparative_analysis[-1, ]

#gold_post.covid <- gold_post.covid[-nrow(gold_post.covid), ]

full.analysis <- merge(comparative_analysis, gold_post.covid, by= "Time", all= T )

plot(full.analysis$Time, full.analysis$Forecasted, type = 'l', col = 'blue',
      ylim = range(c(full.analysis$Forecasted, full.analysis$Actual, full.analysis$Price), na.rm = T),
      ylab= "Gold Prices", xlab = 'Time', xaxt = 'n')

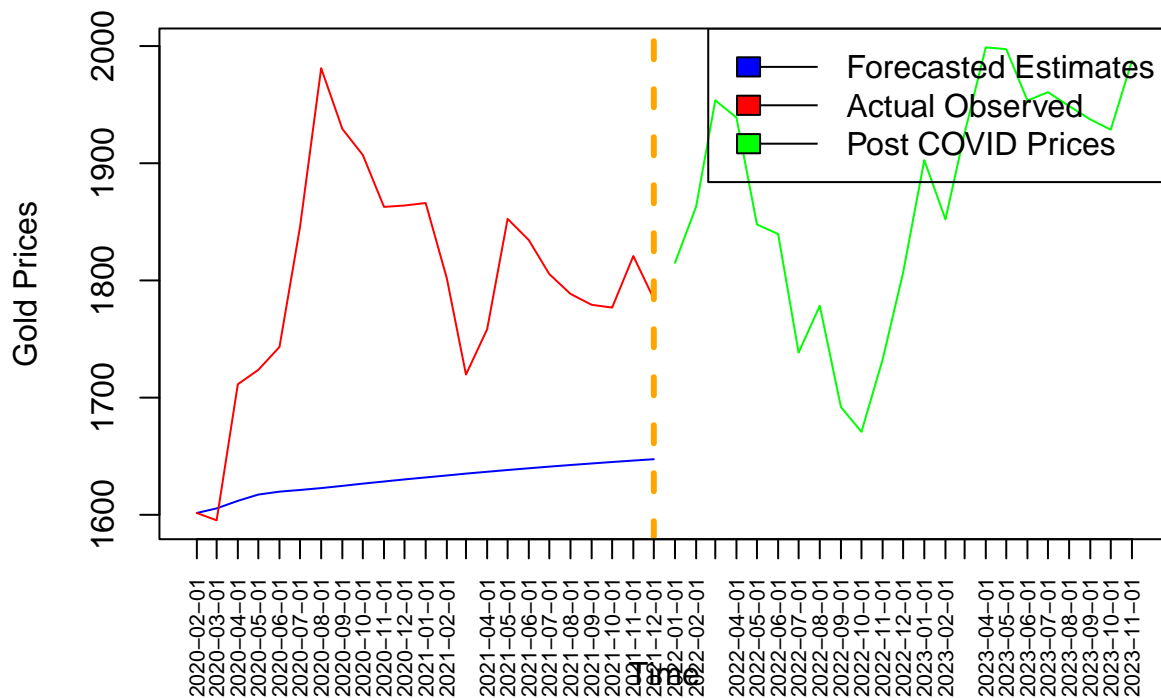
```

```

lines(full.analysis$Time, full.analysis$Actual, col = 'red')
lines(full.analysis$Time, full.analysis$Price, col = 'green')
abline(v= as.Date("2021-12-01"), col = 'orange', lty = 2, lwd = 3)

legend("topright", legend = c("Forecasted Estimates", "Actual Observed", "Post COVID Prices"), fill = c
axis(1, at = full.analysis$Time, labels = format(full.analysis$Time, "%Y-%m-%d"), las = 2, cex.axis = 0

```



```

par(srt = 45)

```

The plot above depicts the answer from the original question on how has much gold prices was affected during COVID. The blue line represents the estimated trend which appears to be linearly increasing. While the red is the actual gold prices that occurred which shows a much more stochastic pattern. To validate the results, the main idea was to see how close those estimates were from connecting to the post COVID actual observed prices. Which by the looks of it, it shows the simplicity of the model as it didn't produce a volatile behavior and it was far off from connecting to the post COVID prices.

7. Mean Absolute Percentage Error (MAPE)

To finalize validation towards our model and see if there was any wrong inputs in the application that impedes us from getting accurate estimates. We use the mean absolute percentage error to measure and tell us the average magnitude of errors in the set of forecasts to quantify our accuracy. From the calculations it shows that the error between the actual and predicted values is 9.6%. Less than 10% is considered fairly

accurate in the econometric world, but given that a simple model was produced, it shows that at the very least something went right in order to have obtained a fairly decent error score.

```
MAPE <- function(actual, forecast) {  
  mean(abs((actual - forecast) / actual)) * 100  
}
```

```
attach(comparative_analysis)  
mape.val <- MAPE(Actual, Forecasted)
```

```
cat("Mean Absolute Percentage Error:\n")
```

```
## Mean Absolute Percentage Error:
```

```
print(mape.val)
```

```
## [1] 9.173771
```

```
cat("\nMAPE: 9.6%")
```

```
##
```

```
## MAPE: 9.6%
```

8. Conclusion

Given our initial problem of answering the question, “How should an asset have performed based on historical ‘pre-COVID’ data observations if COVID never occurred”, through a simple VAR model we have achieved results. Evidence from the counter-factual model, the estimated data concludes that the prices should have been lower than what was historically observed. This calculation is performing as we expected, though admittedly it is not perfect due to having low number of influential variables towards gold prices. More control variables should have been used, and having two variables that perform and impact the same. Nonetheless, the model validation provides support that the application had no fault based on the MAPE metric, it was just really simple. Finally, given the model and validation, the calculated asset performance has been sufficiently calculated and proven against the historical COVID time period.

9. Collaboration

Erick Guevara lead for the development of the model using R-Studio. Main person in regards to developing the script analysis and conducting research on the required methods and packages. Performed the validation and error testing effort in order to ensure that all of the analysis was applied correctly. Additionally helped in interpreting the final results and any weak points that needed to be addressed in the model.

Richard Collie lead in researching the data that will be used. Fixed difficulties in not being able to export comma separated files from the internet in order to use within R Studio. Conducted research from websites that contained the needed data. Additionally, aided in interpreting the data while in the preparation stages (analyzed stationary, white noise, etc).

Grean Ramos lead for the team’s scheduling and ensured that all deadlines were met in an orderly fashion. Also was responsible for maintaining communication with the professor and documentation. Aside from the responsibilities of a project manager. He aided in interpreting the data and fact checked economic information to be 100% accurate when using specific information.

10. References

Ustariz, S. (2021, October 21). Gold: A safe asset. BBVA.CH. <https://www.bbva.ch/en/news/gold-a-safe-asset/> <https://www.investing.com/commodities/gold-historical-data>