**Final Project: Solution Design**

**1. Discuss the results provided by the regression (alpha, beta, R-squared). Would you use the current model in predicting gold prices?**

The regression statistics for Gold show a p-value of 1.47954842648e-16, which represents the probability of finding the observed coefficient estimate or something more extreme under the null hypothesis. And a typical null hypothesis to be a tested is, is that coefficient equal to zero. Therefore, based on the p-value that is less than 0.01, we reject the null hypothesis and say at the 99% confidence interval, the coefficients alpha and beta not likely to be equal to zero. This is evident from its beta value of 0.922226946193.

(a) Beta is the Slope of the regression or the Trend of the series. The Gold's beta shows the yearly trends of gold from 1978 to 2014. Using a different time period for the regression or different return intervals (weekly or daily) for the same period can result in a different beta. In other words, “Beta” refers to the degree to which the forecasted gold price more or less volatile than the previous period gold price. A beta coefficient of 1 implies that the current prices will move with the previous period gold prices, thus the best predictor of any future value is the current value. However, this is not usually the case in real time. While, a beta of less than 1 means it will be less volatile than the previous period gold price. Therefore, a beta of 0.9222, means that every time the previous period gold price increases (or decreases) by a certain amount, the current gold prices increases (or decreases) by 0.9222 multiplied by that amount.

(b) Alpha is the Intercept of the regression = 0.428123903567; This is a measure of forecasted Gold price's performance, when it is compared with the previous period gold price. The intercept, alpha shows how the previous period gold prices perform when the market has a flat year, where the beta is equal to zero. An Alpha of 0.428 indicates that the current gold prices have positively drifted from the previous period gold prices by exp (0.428) (USD 1.53).

Since, the regression results show that both the Beta (p-value <0.0001) and Alpha (P-value <0.0001) are statically significant at the 99% level (strong). Then, there is statistical confidence at the 99% confidence interval, to support that current gold prices are affected by the previous period gold prices. Furthermore, the current gold prices have a strong positive relationship (r = 0.932) with the previous period gold prices. Therefore, the current gold prices track the previous period gold prices.

(c) The coefficient of determination or R squared of the regression = 0.868737862694, suggests that 86.87% of the variability in the Gold price comes from the historical or previous period prices. While, only 13.13% comes from out of sample sources such interest rates risk, government policies risk, inflation risk and so on. Therefore, R-square tells us what fraction of the future gold’s volatility is attributable to historical movements. As a result, the current model or similar auto regressive model can be used in predicting gold prices, considering that only 13.13% of variability is attributed to diversifiable risk, not explained by the model.

**2. Implement an econometric model for predicting gold price evolution (use the latest available data). The econometric model can be ARMA, VAR, VEC, GARCH, Multiple Regression, Bayesian regression, Machine Learning methods or any other method that you consider suitable. Explain your model choice for this exercise. Provide the Python code and the final forecasts.**

An autoregressive (AR) process operates under the premise that past values have an effect on current values and based on the results of the OLS prediction, which explained 87% of the variability in the model. Two models will be used. The first one, will be the OLS and the second one is an ARIMA model. AR models will be ideal in providing the best fit model, since an OLS is a form of AR.

ARIMA is a time series forecasting technique for short run, which is widely used in today’s world since the evolution of sophisticated statistical software package. The model chosen is an Autoregressive Integrated Moving-Average (ARIMA) using a Kalman Filtering Algorithm. ARIMA has four major steps in model building- Identification, Estimation, Diagnostics & Forecast. Within these four steps, the tentative model parameters can be identified through graphs ACF and PACF, then the coefficient is determined and find out the likely model, next steps involves validating the model and at the end, we use simple statistics and confidence intervals to determine the validity of the forecast and track model performance. A Python version of R Programming Auto ARIMA will be use to estimate the parameters for step 1. ARIMA model uses the historic data and decomposes it into Autoregressive (AR) Indicates weighted moving average over past observations, Integrated (I) Indicates linear trends or polynomial trend and Moving Average (MA) Indicates weighted moving average over past errors. The Kalman Filter will be used to improve the accuracy of the forecast.

The Kalman filter is a set of mathematical equations that provides an efficient computational (recursive) means to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter is very powerful in several aspects: it supports estimations of past, present, and even future states, and it can do so even when the precise nature of the modeled system is unknown. The Statsmodels module will be used to the implementation of the Kalman Filter and the ARIMA model through the Seasonal Autoregressive Integrated Moving-Average with eXogenous regressors (SARIMAX) class. The SARIMAX class is a fully-fledged model created using the states space backend for estimation. Moreover, SARIMAX can be used very similarly to tsa models, and works on a wider range of models by adding the estimation of additive and multiplicative seasonal effects, which includes a Kalman filter algorithm.

The module required and their installation steps are outlined in the ReadMe file. The implementation of the code and output are also attached. The Code Explanation document attached explains the Python code with in-line comments. An ARIMA model based on the SARIMAX class, which uses the maximum likelihood and Kalman Filter to predict gold price time series. The code is executed in a systematic and sequential stepwise implementation, which involves forecasting the gold prices with a OLS linear regression. The OLS model was used to predict gold prices four days in the future and the value was compare to the World Gold Council daily price of 2018-04-05.

An ARIMA model was also used to forecast the gold prices by adapting a procedures used in STATA and R programming for ARIMA forecasting. The python adaptation relied on the statsmodels state space model, especially the SARIMAX class. Then using the results wrapper and the \_get.forecast tool for SARIMAX , a 11 day/step forecast was one for a dynamic ARIMA model. The log forecast was inverse using the power of 10 to convert the forecasted logarithms to forecasted out-of-sample gold prices. The ARIMA model performed better than the OLS regression, it Mean square error(MSE).The procedure is outlines in the python code (py file) and the output is shown in the output pdf.

**Conclusion**

The SARIMAX provided a black box coding approach to using the Kalman Filter with an ARIMA model. The statsmodels documentation and github sites were also used in the implementation of the project. The results showed that the gold prices were on a downward to stationary trend in the short term, as mean reversion property or market correction similar to 1982 downward trend. The downward trend could also be seasonality effects of troughs between the April to June, which usually peaks later in the year around September and dips towards December. Peaking again February for another cycle. Lastly, the SARIMA model developed suggested that gold has a seasonality component of 11 lags between days.

# **References**

Cuesta, H. (2013) *Practical Data Analysis: Transform, model, and visualize your data through hands-on projects, developed in open source tools.* Mumbai: Packt Publishing.

Doucet, A., & Johnsen, A. M., (2008). *A Tutorial on Particle Filtering and Smoothing: Fifteen years later*. Available at: <http://www.cs.ubc.ca/~arnaud/doucet_johansen_tutorialPF.pdf>

QuantInsti (2017) *Gold Price Prediction using machine learning in Python.* Available at: <https://www.quantinsti.com/blog/gold-price-prediction-using-machine-learning-python/>

Quandl (2018) *World Gold Council Dataset in USD from January 1970 to March 2018. Available at:* [*https://www.quandl.com/data/WGC/GOLD\_DAILY\_USD-Gold-Prices-Daily-Currency-USD*](https://www.quandl.com/data/WGC/GOLD_DAILY_USD-Gold-Prices-Daily-Currency-USD)

Ohri, A. (2017) SARIMAX *for Air Passenger from R-Dataset: An Auto ARIMA example*. Available at: <https://github.com/decisionstats/pythonfordatascience/blob/master/time%2Bseries.ipynb>

Perktold, J., Seabold, S., & Taylor, J., (2017) *SARIMAX Module PyCode*. Available at: <https://github.com/statsmodels/statsmodels/blob/master/statsmodels/tsa/statespace/sarimax.py>

Saha, A., & Upadhyay, R.,(2015) <http://ucanalytics.com/blogs/python-code-time-series-forecasting-arima-models-manufacturing-case-study-example/>

STATA, *ARIMA Post estimation*. Available at: <https://www.stata.com/manuals13/tsarimapostestimation.pdf>

Smith, T. G.,(2017). *Auto ARIMA from R in Python.* Available at: <https://github.com/tgsmith61591/pyramid/blob/master/examples/quick_start_example.ipynb>

Vincent, T., (2017) A *Guide to Time Series Forecasting with ARIMA in Python 3*

Available at: <https://www.digitalocean.com/community/tutorials/a-guide-to-time-series-forecasting-with-arima-in-python-3>