# Estimation and application of best ARIMA model for forecasting the uranium price.

Medeu Amangeldi

May 13, 2018

Capstone Project
Superviser: Dongming Wei
Second reader: Zhenisbek Assylbekov

**Abstract**

This paper presents the application of an iterative approach for prediction of uranium price by model identification, parameter estimation and diagnostic checking which are designed by Box and Jenkins. In particular, the autoregressive integrated moving average model is used to predict the future values of monthly uranium price. As the analysis of structural dependence in observations is one of the key features of time series analysis, the past values, which were taken as monthly values from January 2000 to June 2017, are used for forecasting. As a result, ARIMA(2,1,0) became one that met all the criteria and predicted the increase of uranium price over time within 95% confidence.

## 1 Introduction

The autoregressive integrated moving average model, ARIMA, is one of the statistical and forecasting models in time series analysis[5].

The data [3] of monthly uranium price from January 2000 to June 2017 of 210 data points is used as the object of the statistical investigation.

In order to build an appropriate model for forecast, mainly, the iterative approach suggested by Box and Jenkins[1] was used as follows

1. Model Identification Stage:
   - Apply logarithmic transformation to uranium price in order to stabilize its variance
   - Assess the stationarity of logarithmic price and use differencing in case of detecting the non-stationarity.
   - Identify possible ARMA models of order (p,q) by using two criteria, AIC and BIC, for $w_t = (1-B)^d y_t$
     such that,
     $$\phi(B)w_t = \theta(B)\varepsilon_t$$
     where,
     $B$ - backward shift operator, (i.e. B$y_t$=$y_{t-1}$)
     $d$ - order of differencing
     $y_t$ - uranium price at time t
     $c$ - constant term
     $\varepsilon_{t-l}$ - residual term for $l = 1, 2, \ldots, q$ , $(y_t - y_{t-1})$
     $\varepsilon_t$ - random shock $\sim N(0, \sigma^2)$
     $\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$ (polynomial operator)
     $\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q$ (polynomial operator)
2. Parameter Estimation Stage:

- Estimate parameters $\phi$ and $\theta$ by using the conditional sum of squares and maximizing the log-likelihood function.

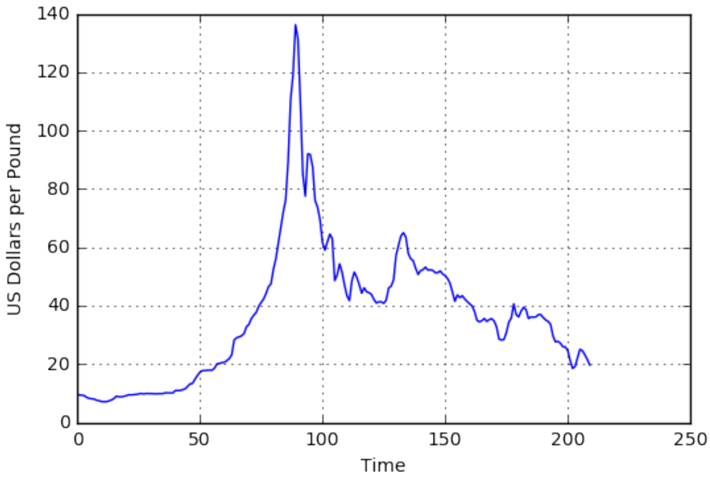
3. Diagnostic Checking Stage:

   - Analyze the residuals of each model, which were selected from previous stage, for no autocorrelation and that they are white noise with Ljung-Box and Box Pierce tests, respectively.

4. Use the best ARIMA(p,d,q) model that passes the diagnostic checking to forecast future values of uranium price within 95% confidence interval.

All listed procedures were implemented in Python (in Appendix)[4, 6].

Figure 1: Monthly uranium price from Jan, 2000 to Jun, 2017




# 2 Model

## 2.1 Autoregressive model

Autoregressive model is the model in which the equation represents the linear dependence of the value from its $p$ past values; written as AR(p) and defined as follows [5]

$$y_t = c + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$

where,
$c$ - constant term
$\phi_1, \ldots \phi_{t-p}$ - coefficients
$y_t$ - observation at time t
$\varepsilon_t$ - random shock $\sim$ N(0, $\sigma^2$)

## 2.2 Moving average model

Moving average model is the model in which the equation represents the linear dependence of the value from its $q$ past residuals; written as MA(q) and defined as follows

$$y_t = c + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

where,
$c$ - constant term

$\theta_1, \ldots \theta_{t-q}$ - coefficients
$\varepsilon_{t-l}$ - residual term for $l = 1, 2, \ldots, 1$, $(y_t - y_{t-1})$
$\varepsilon_t$ - random shock $\sim$ N(0, $\sigma^2$)

### 2.3 Mixed ARMA(p,q) and ARIMA(p,d,q) models

ARMA model is the model in which both autoregressive and moving average models are considered and used for stationary time series; written as ARMA(p,q) and defined as follows

$$\phi(B)y_t = c + \theta(B)\varepsilon_t$$

While Autoregressive Integrated Moving Average (ARIMA) of order (p,d,q) stands for the model used for non-stationary time series and use difference in order to produce stationary condition; rewriting difference form $w_t = \nabla^d y_t = (1-B)^d y_t$ gives the ARMA form:

$$\phi(B)w_t = c + \theta(B)\varepsilon_t$$

$$\phi(B)(1-B)^d y_t = c + \theta(B)\varepsilon_t$$

where,
$B$ - backward shift operator, (i.e. B$y_t$=$y_{t-1}$)
$y_t$ - uranium price at time t
$c$ - constant term
$\varepsilon_{t-l}$ - residual term for $l = 1, 2, \ldots, q$, $(y_t - y_{t-1})$
$\varepsilon_t$ - random shock $\sim N(0, \sigma^2)$
$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$
$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q$

## 3 Order Identification and Parameter Estimation

Stationarity assumption states that the time series process, i.e. $(y_t)$, and the value from that process at time $t$ has the same mean and variance as at time $t+n$ or $t-n$, for n=$1, 2, \ldots$. It is important for data to be stationary as the Box and Jenkins methodology, especially ARMA model is applicable to fit such time series.

This is because under stationarity assumption, statistical equilibrium is held, in other words, the process $y_t$ has constant mean and variance throughout the time and its future values can be predicted taking into account these properties.

Monthly uranium price from January 2000 to June 2017 is shown in Figure 1.

The way Box and Jenkins suggests to check the stationarity is to plot the autocorrelation function and see if the function dies-out quickly.

Important note: Generally, a time series has non-stationary behavior if it consists a unit root. Augmented Dickey Fuller test examines if a time series has a unit root. For instance, imagine AR(1) process and we assume that $\varepsilon_t$ is zero-mean stationary.

$$y_t = c + \phi_1 y_{t-1} + \varepsilon_t$$

Assuming that this time series has a unit root, $\phi_1 = 1$:

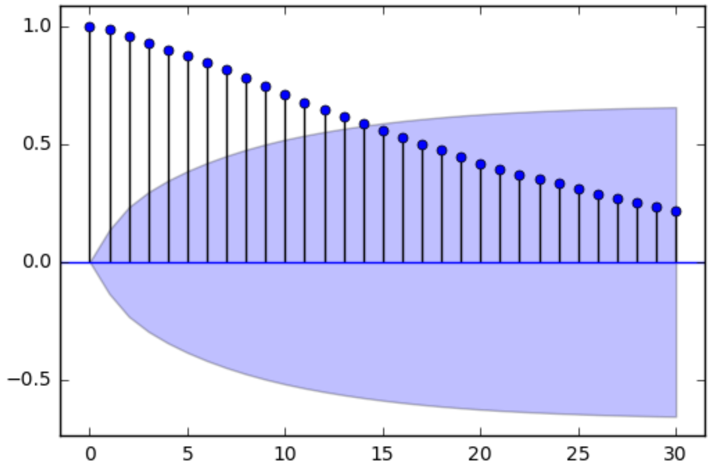
$$y_t - y_{t-1} = c + \varepsilon_t$$

$$(1-B)y_t = c + \varepsilon_t$$

$$y_t = \frac{c + \varepsilon_t}{1-B} = \sum_{j=0}^{t} 1^j c + \sum_{j=0}^{t} 1^j \varepsilon_{t-j} = ct + \varepsilon_t + \varepsilon_{t-1} + \ldots$$

So, above the time series has a mean that changes over time $(ct)$ and it contradicts the stationarity assumption. Generally, unit root series refer to the fact that the autocorrelation decays very slowly to zero (nearly linearly). So slow decay of ACF is a signal for non-stationary behavior.

Figure 2: ACF of monthly uranium price

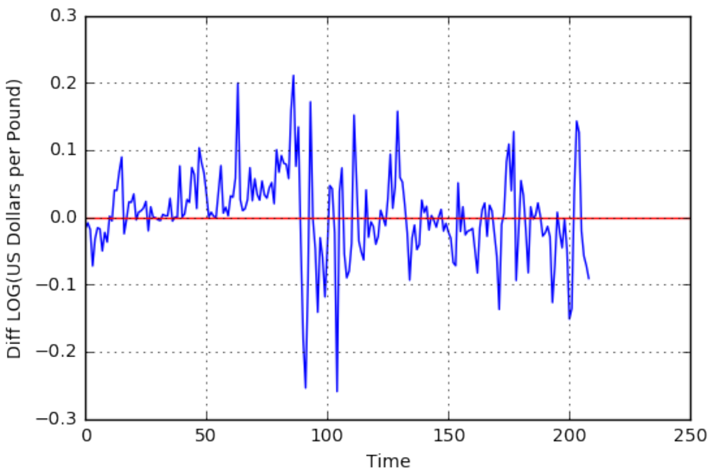



It can be seen that the autocorrelation function is nearly linear and dies-out very slowly, so the monthly uranium price is non-stationary, Figure 2

Moreover, the Augmented Dickey Fuller test gives the p-value more than 0.05, we fail to reject the null hypothesis of non-stationarity of data.

Applying the Box and Jenkins approach of taking logarithm and difference the observations (Figure 3), $lny_t - lny_{t-1}$, helps to produce stationarity.

Figure 3: d=1 difference



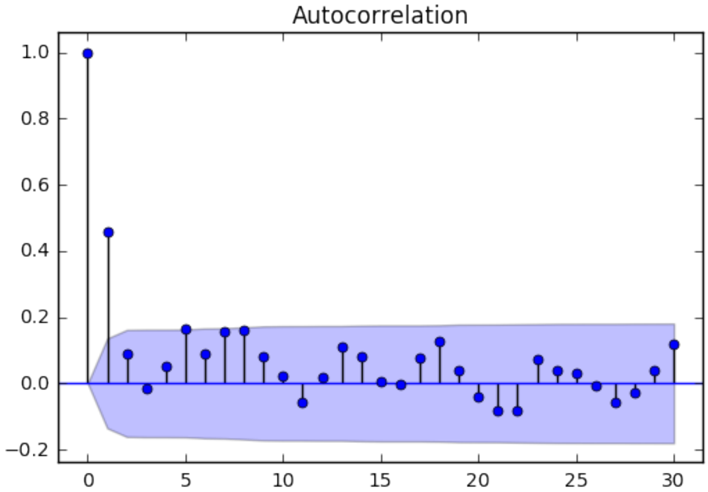

The autocorrelation function of log differenced data is shown in Figure 4.

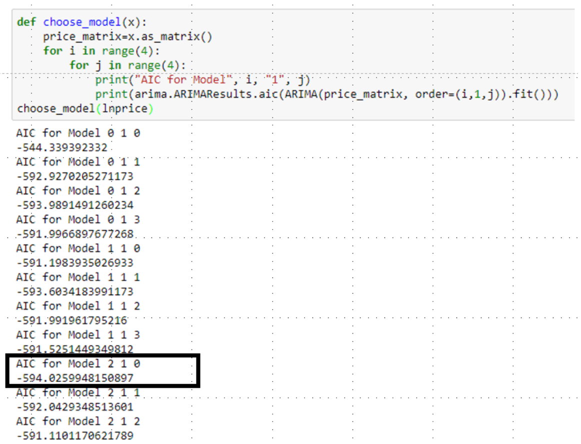
It can be seen that the ACF dies-out very quickly, suggesting that the autocorrelation after the first lag is insignificant at 95% confidence interval[2]. Moreover, Augmented Dickey Fuller test gives the p-value less than 0.05 and we reject the null hypothesis that the data is non-stationary.

As the stationarity condition was achieved after taking 1st difference, the order d is known, and can be labeled as I(1).

Next, I set upper bounds, $P$ and $Q$, for AR and MA order, respectively. Then, I fitted all possible models for $p \leq P$ and $q \leq Q$: (0,1,0), (1,1,0), (2,1,0), (2,1,1), (2,1,2), (0,1,1), (0,1,2), (1,1,2), (1,1,1)

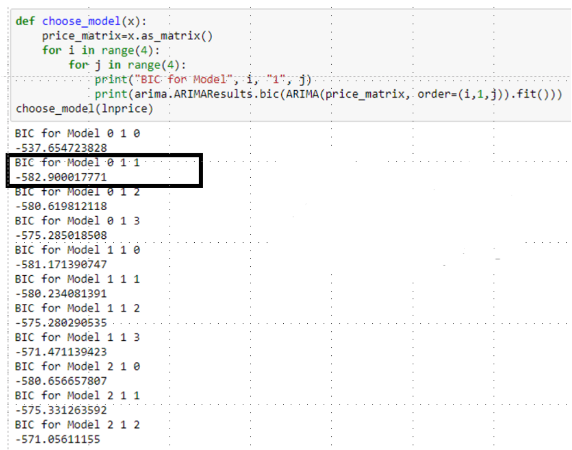
Figure 4: ACF log differenced monthly uranium price



```
def choose_model(x):
    price_matrix=x.as_matrix()
    for i in range(4):
        for j in range(4):
            print("AIC for Model", i, "1", j)
            print(arima.ARIMAResults.aic(ARIMA(price_matrix, order=(i,1,j)).fit()))
choose_model(lnprice)
AIC for Model 0 1 0
-544.339392332
AIC for Model 0 1 1
-592.9270205271173
AIC for Model 0 1 2
-593.9891491260234
AIC for Model 0 1 3
-591.9966897677268
AIC for Model 1 1 0
-591.1983935026933
AIC for Model 1 1 1
-593.6034183991173
AIC for Model 1 1 2
-591.991961795216
AIC for Model 1 1 3
-591.5251449349812
AIC for Model 2 1 0
-594.0259948150897
AIC for Model 2 1 1
-592.0429348513601
AIC for Model 2 1 2
-591.1101170621789
```

Figure 5: AIC values for different possible models and the model with the least criterion



```
def choose_model(x):
    price_matrix=x.as_matrix()
    for i in range(4):
        for j in range(4):
            print("BIC for Model", i, "1", j)
            print(arima.ARIMAResults.bic(ARIMA(price_matrix, order=(i,1,j)).fit()))
choose_model(lnprice)
BIC for Model 0 1 0
-537.654723828
BIC for Model 0 1 1
-582.900017771
BIC for Model 0 1 2
-580.619812118
BIC for Model 0 1 3
-575.285018508
BIC for Model 1 1 0
-581.171390747
BIC for Model 1 1 1
-580.234081391
BIC for Model 1 1 2
-575.280290535
BIC for Model 1 1 3
-571.471139423
BIC for Model 2 1 0
-580.656657807
BIC for Model 2 1 1
-575.331263592
BIC for Model 2 1 2
-571.05611155
```

Figure 6: BIC values for different possible models and the model with the least criterion

In addition, by Box and Jenkins approach [2], the information criterion values may be used to choose possible appropriate models, which are Akaike Information Criterion (AIC) and Bayesian Information Criterion(BIC); the least value is better[2]. I used BIC to assess the quality of the model and it is defined as follows:

$$AIC = ln(\sigma_a^2) + 2r/n$$
$$BIC = ln(\sigma_a^2) + rln(n)/n$$

where,
$\sigma_a^2$ - estimated variance of $a_t$
$r$ - number of parameters with constant term, $p + q + 1$
$n$ - number of observations.

By implemented code (in Appendix) in Python, I estimated values of AIC and BIC for possible models in Figure 5 and 6.

By above, I considered models of orders ARIMA(0,1,1) and ARIMA(2,1,0), as they have the least criterion.

In order to check the goodness-of-fit of the models, diagnostic checking was performed, particularly, Ljung-Box test for no autocorrelation and Box-Pierce test for white noise of residuals.

# 4 Model Diagnostics

Ljung-Box test for no autocorrelation performed by Python code (in Appendix) showed interesting results for ARIMA(2,1,0),(Figure 4). No p-values are less than 0.05 with which we fail to reject the null hypothesis of no autocorrelation. It means that there is no autocorrelation between series up to 40 lags.

After performing the Box-Pierce test for the white noise of residuals, no p-values were less than 0.05 with which we fail to reject the null hypothesis that the residuals are white noise.

Figure 7: LB and Box Pierce tests results (statistic and p-values for each test, respectively)

```
ssd.acorr_ljungbox(arima.ARIMAResults.resid(model_fit1),boxpierce=True)
```

```
(array([  3.94788243e-03,   1.61383306e-02,   5.31290882e-02,
          7.70624654e-02,   7.45277623e+00,   8.38850516e+00,
          1.11525070e+01,   1.35425910e+01,   1.36413146e+01,
          1.41611486e+01,   1.58671153e+01,   1.58976031e+01,
          1.77908072e+01,   1.81770729e+01,   1.82098694e+01,
          1.82802133e+01,   1.85418053e+01,   2.11458916e+01,
          2.11521764e+01,   2.12401756e+01,   2.13531341e+01,
          2.45524354e+01,   2.87594539e+01,   2.89979615e+01,
          2.91236015e+01,   2.91955787e+01,   2.97899149e+01,
          2.98189353e+01,   2.99146628e+01,   3.51380350e+01,
          3.63937730e+01,   3.64501515e+01,   3.71082564e+01,
          3.71257583e+01,   3.72060768e+01,   3.72245922e+01,
          3.72250176e+01,   4.07778792e+01,   4.10334805e+01,
          4.10340087e+01]),
 array([ 0.94990014,  0.9919633 ,  0.99679443,  0.99927647,  0.18908484,
         0.21099929,  0.1321037 ,  0.0944946 ,  0.13566862,  0.16576721,
         0.14613632,  0.19597118,  0.16562513,  0.19883463,  0.25179292,
         0.30774552,  0.35545665,  0.27211665,  0.3284536 ,  0.38311921,
         0.43756749,  0.31895236,  0.18841774,  0.22020759,  0.25878215,
         0.30226348,  0.3236756 ,  0.37192511,  0.41827312,  0.23765251,
         0.23192616,  0.26930172,  0.28519667,  0.32693893,  0.36778996,
         0.41248176,  0.45871854,  0.34926387,  0.38147197,  0.4250115 ]),
 array([  3.89175140e-03,   1.58511010e-02,   5.19653004e-02,
          7.52181076e-02,   7.20623995e+00,   8.10649100e+00,
          1.07525970e+01,   1.30294069e+01,   1.31229838e+01,
          1.36132538e+01,   1.52141135e+01,   1.52425784e+01,
          1.70011945e+01,   1.73581700e+01,   1.73883241e+01,
          1.74526671e+01,   1.76907034e+01,   2.00479569e+01,
          2.00536162e+01,   2.01324401e+01,   2.02330856e+01,
          2.30684853e+01,   2.67770419e+01,   2.69861599e+01,
          2.70957228e+01,   2.71581485e+01,   2.76707987e+01,
          2.76956930e+01,   2.77773563e+01,   3.22085582e+01,
          3.32679012e+01,   3.33151950e+01,   3.38641356e+01,
          3.38786514e+01,   3.39448856e+01,   3.39600665e+01,
          3.39604133e+01,   3.68397466e+01,   3.70456813e+01,
          3.70461044e+01]),
 array([ 0.95025711,  0.99210577,  0.99689809,  0.99931026,  0.20574828,
         0.23040489,  0.1497806 ,  0.11084181,  0.15711989,  0.19137422,
         0.17290307,  0.22843787,  0.19924947,  0.23759291,  0.29618515,
         0.35690564,  0.40859991,  0.33012622,  0.39135837,  0.44967283,
         0.50654333,  0.39787109,  0.26557342,  0.3050987 ,  0.3511194 ,
         0.40107676,  0.42809275,  0.48065795,  0.52983972,  0.35789158,
         0.3573399 ,  0.40308315,  0.42565232,  0.47359053,  0.51890718,
         0.56594539,  0.61232148,  0.52300929,  0.55930083,  0.6039807 ]))
```

# 5 Forecasting ability

As ARIMA(2,1,0) successfully passes the diagnostics stage, it can be used to predict the uranium price. However, the model must be checked for forecasting ability and was used to predict the uranium prices in last five months by calculating the margin of error of predicted prices which is defined as follows:

$$ME_t = \frac{FP_t - AP_t}{AP_t} \times 100$$

where,
$ME_t$ - margin of error for forecasted uranium price at time t
$FP_t$ - forecasted uranium price at time t
$AP_t$ - actual uranium price at time t

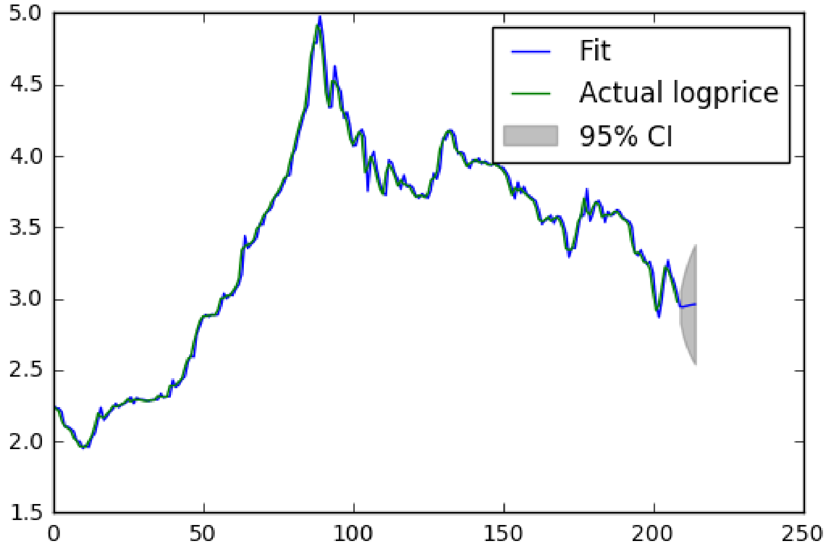
Plot of the prediction is shown below as well, Figure 8.

It can be seen that out-of-sample forecast shows the increase of uranium logprice over time within 95% confidence interval. Indeed, this suggests that the price of uranium will increase over time.

Table 1: Actual and forecasted (ARIMA(2,1,0)) uranium prices with margin of errors for the last five months data

| Month | Actual price ($) | Forecasted price ($) | Margin of Error (%) |
|---|---|---|---|
| Feb 2017 | 25.06 | 23.74 | -5.27 |
| March 2017 | 24.55 | 26.26 | 6.97 |
| April 2017 | 23.17 | 23.87 | 3.04 |
| May 2017 | 21.56 | 22.59 | 4.77 |
| June 2017 | 19.68 | 20.98 | 6.61 |

Figure 8: ARIMA(2,1,0) model prediction for logprice




# 6 Conclusion

Non-stationary time series of monthly uranium price between January 2000 and June 2017 was taken as an object of statistical investigation. Particularly, most appropriate ARIMA model was estimated by Box and Jenkins approach and used for prediction of the future values.

Finally, ARIMA(2,1,0) became relatively most preferable model to forecast the uranium price for future and showed the forecasting ability within 7% margin of error, as a result, the model predicted the rise of uranium price in the future. Further investigations can be done on modification of linear ARIMA model, particularly, using recurrent neural networks.

## References


[1] G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden Day, San Francisco CA, 5 edition, 1970.

[2] George E P Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time Series Analysis: Forecasting and Control*. Jon Wiley and Sons, Inc, Hoboken NJ, 5 edition, 2016.

[3] International Monetary Fund. Primary commodity prices: Monthly data. Available at http://www.imf.org/external/np/res/commod/index.aspx.

[4] Michael Grogan. Arima models: Stock price forecasting with python and r. Available at http://http://www.michaeljgrogan.com/arima-model-statsmodels-python/ (2017/10/5).

[5] Sungki Kim, Wonil Ko, Hyoon Nam, Chulmin Kim, Yanghon Chung, and Sungsig Bang. Statistical model for forecasting uranium prices to estimate the nuclear fuel cycle cost. *Nuclear Engineering and Technology*, 49(5):1063 – 1070, 2017.

[6] Josef Perktold, Skipper Seabold, and Jonathan Taylor. Statsmodels: Statistics in python. Available at http://http://www.statsmodels.org/dev/index.html (2009/01/01).