FALL 2019

STAD57H3F

Project Final Version

Group 10
12.02.2019
Siyi Wei siyi.wei@mail.utoronto.ca
Ruotong Zheng sunny.zheng@mail.utoronto.ca
Zijie Wang zijie.wang@mail.utoronto.ca

Table of Contents

Overview

The purpose of this project is to develop a Time Series model to predict GDP deflators.

Different Time Series Modelling techniques were tried in search for the best model, and multiple tests were conducted to assess the performance of the models. A detailed methodology description and walkthrough of the analysis can be found in Section 3 and Section 4.

Below is a result summary on Mining, Oil and Gas Extraction (MOGE) Sector. [1]

MOGE Sector*	AIC	BIC	Residual	tsCV MSPE	tsCV MAPE
ARIMA Only	417.36	427.3	328.4649	132.9285	0.1282342
ARIMA With All External Regressors	333.29	359.39	164.1758	96.3871	0.1176507
Model A	338.09	358.16	188.1708	43.5178	0.0937107
Model B	339.46	359.53	190.245	49.5747	0.0942829
Model C	-29.57	-25.59	323.7368	140.0203	0.1281508

*The AIC and BIC of Model C is of little value in model selection, as it is essentially fitting a processed series, the last 3 metrics, however, is still useful.

The first 4 models are similar, we observe that Model A outperforms other models in almost all metrics.
Model C uses a different approach to fit the series, so it is hardly comparable to other models. However, this model is simple and can be used when external regressors are unattainable.

Further into discussion we show that this conclusion is universal: test on other industry sectors and subsectors produces similar result.

All data comes from Statistics Canada.

[1] Definitions of this model is specified in Section 2

Introduction

The purpose of this project is to make predictions on GDP deflators for different industries.

From an economic perspective, several external factors can contribute to the value of the deflator. Examples include Labour productivity (LP), Capital input (CI), Working hours, etc. The use of these external regressors can potentially help with predictions.

Another important factor, perhaps the most important factor, is the time. One would expect the productivity of our society in 2020 to be greater than that of 1920 – therefore we assume a drift term would contribute to the deflator; to eliminate such effect, we fit the differenced series rather than the series itself.

In the report to follow, we seek to answer the following question:

Will the introduction of external regressors help with the prediction? If so, which external regressors to include?

With the questions in mind, we start by investigating the structure of the Mining, Oil and Gas Extraction (MOGE) Sector.

Initial analysis suggests the deflator series is a random-walk-like series; which supports the existence of the drift. (I(1) series)

Starting from there, we examined the effectiveness of 5 models:

1. ARIMA Only: Use auto.arima to fit the deflator series;
2. ARIMA with All External: Use auto.arima with all external regressors;
3. Model A: do a multilinear regression on the differenced series first (StepAIC is used for model selection), then fit the deflator with selected external resources.
4. Model B: do auto.arima on the series, then apply multilinear regression on the residual (Again, StepAIC is used for model selection). Generate the model by fitting the series with selected external regressors.
5. Model C: take log of the deflator series, and fit the log series with auto.arima

Five metrics are used for model selection and performance assessment. In-Sample Testing metrics are AIC, BIC and absolute sum of residuals; Out-of-Sample Testing metrics are MSPE and MSAE. Generally, Model A outperforms other models.

Analysis on other sectors and subsectors yields similar conclusion.

A list of external regressors are considered[2]: MP, LP, CP, CSI, LI, CI, LCO, CC, LCE, and Ccost.

[2] Appendix A

Methodology

We want to demonstrate two ways for improving the impact of xreg in details, namely, Approach A and Approach B. Aside from using external regressors, we also seek to improve the model by doing a log transform on the series. (Approach C)

The original method in Auto.Arima for evaluating the weights of xreg was based on the following formula.

$$y_t = \beta X_t + \epsilon_t$$
$$\epsilon_t = \varphi_1 \epsilon_{t-1} + \ldots + \varphi_p \epsilon_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots \theta_q \varepsilon_{t-q}$$

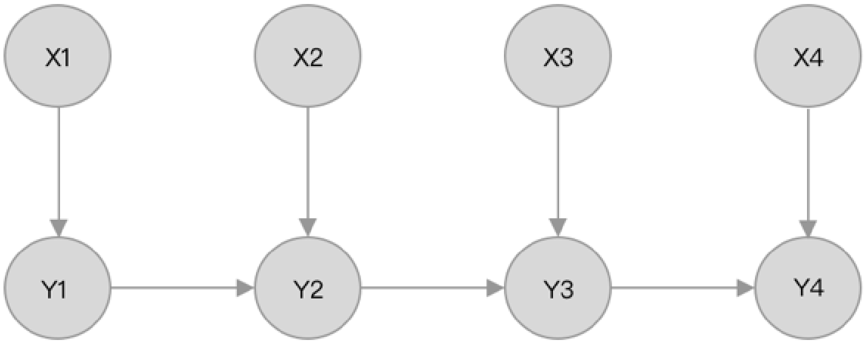
Where y is the original data and x is xreg. This method could have a potential problem; there is no selection process for xreg. This problem could cause overfitting which will further decrease the predicting accuracy and will be extreme time consuming for large xreg datasets.

Approach A:

Our first approach uses Auto.Arima to predict the original time series data initially. We would expect the residual to contain all the information linearly related to xreg. That information could be extracted using StepAIC linear fit. StepAIC is a generally accepted method for model selection, which outputs the model with minimum AIC in all combinations of xreg. The process could be formalized in the following notations.

$$y_t = \varphi_1 y_{t-1} + \ldots + \varphi_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots \theta_q \varepsilon_{t-q} + \epsilon_t$$
$$\epsilon_t = StepAIC(\beta X_t + \omega_t)$$

The Bayesian Network representation is shown below.



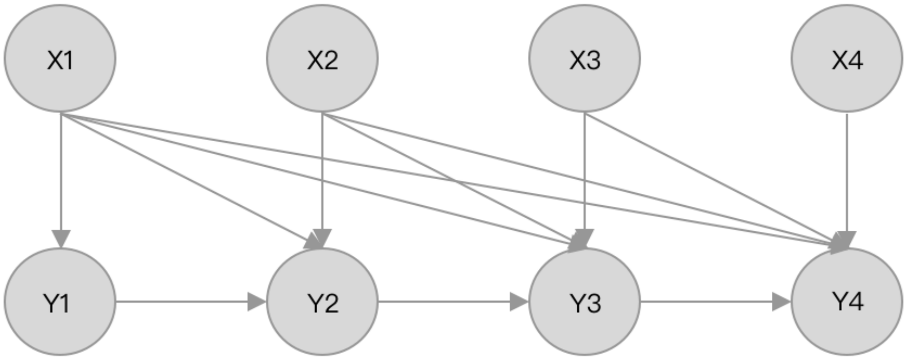
However, this method will not take the impact of the past values of the xreg. We would introduce an alternative approach which will include the past value of xreg.

Approach B:

The second approach will first fit the original time series data StepAIC linear fit. Then use Auto.Arima to fit the residual, the math notation could show that this approach will take consider of the past value of the external regressors.

$$y_t = StepAIC(\beta X_t + \epsilon_t)$$
$$\epsilon_t = \varphi_1 \epsilon_{t-1} + \ldots + \varphi_p \epsilon_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots \theta_q \varepsilon_{t-q}$$

Our second approach is almost the same as the Auto.arima, except we will do a StepAIC for parameters selection.



Approach C:

Approach C handles the drift differently. Rather than taking the difference, we take the log of the series and then fit the log series with ARIMA. Simplicity is the greatest advantage of this approach, and since external regressors are not needed for this approach, we get to test this approach on Subsectors. (The information of external regressors for subsectors is not available on StatsCan)

Analysis Walkthrough

Initial Analysis

Figure 1: Plot of the MOGE Sector deflator[3]

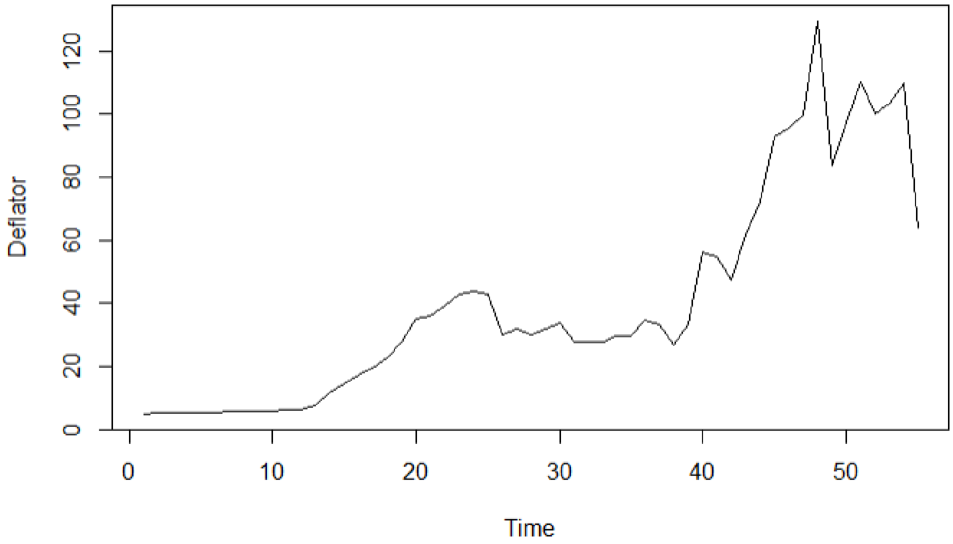
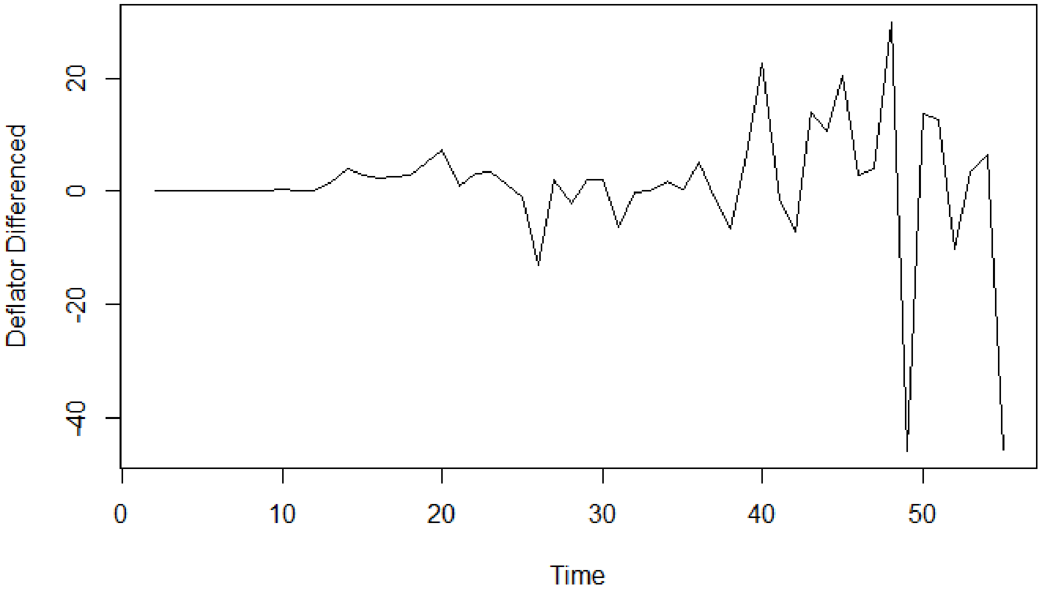


Figure 2: Plot of the differenced deflator (at lag 1)[4]



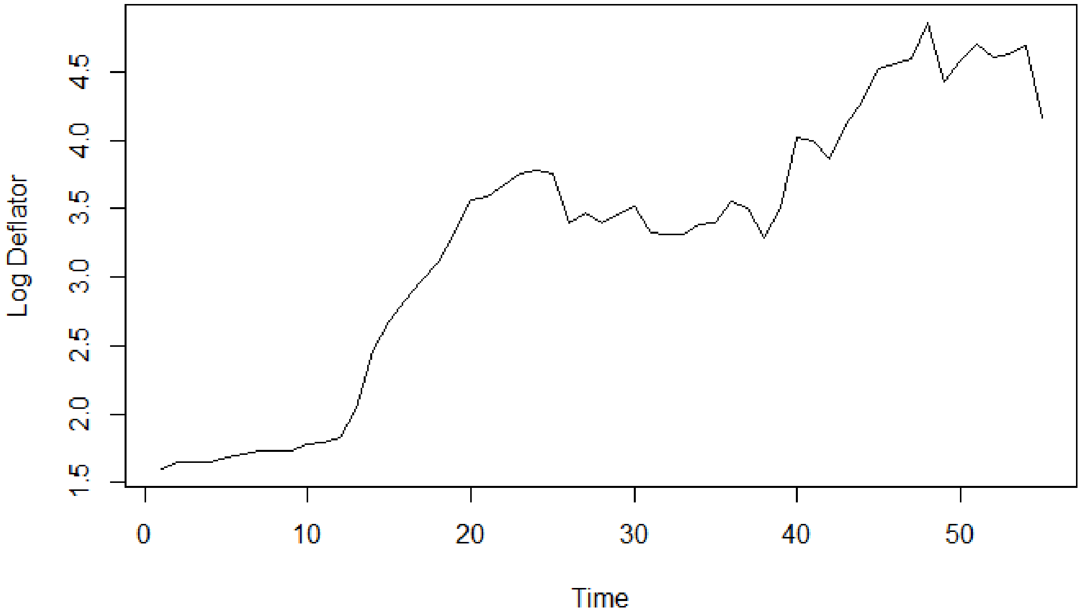
The main goal in initial analysis is to test whether data is a stationary Time Series. We observe that differencing smoothens the fluctuation of the deflator series. ACF and PACF plots show that

[3] Figure 1 see Appendix C - Code 2 Figure 1

[4] Figure 2 see Appendix C – Code 4 Figure 2

the differenced series is almost a White Noise Series. One caveat here is that the first few values are small (the deflator of the first 10 years), which could harm the integrity of the model. To resolve this issue, we would consider excluding these values when fitting models; or alternatively, taking a log transform.

Figure 3: Plot of the log deflator[5]



As we can see from Figure 3, taking log makes the differences in values less drastic; which produces a random-walk-like behavior (with a drift).

Model Selection

Five models are developed in this subsection as candidates for the best model.

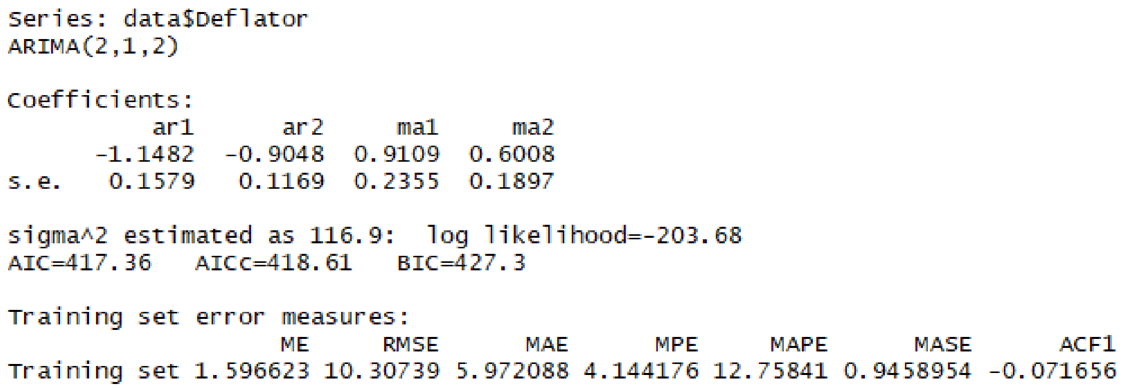
ARIMA Only:

Use auto.arima to fit the deflator series.
Direct and simple; nothing too interesting.

Table 1: [6]

[5] See Appendix C – Code 5 Figure 5

[6] See Appendix C – Code 10 Solution 2



```
Series: data$Deflator
ARIMA(2,1,2)

Coefficients:
          ar1      ar2     ma1     ma2
      -1.1482  -0.9048  0.9109  0.6008
s.e.   0.1579   0.1169  0.2355  0.1897

sigma^2 estimated as 116.9:  log likelihood=-203.68
AIC=417.36   AICc=418.61   BIC=427.3

Training set error measures:
                   ME     RMSE      MAE      MPE     MAPE      MASE      ACF1
Training set 1.596623 10.30739 5.972088 4.144176 12.75841 0.9458954 -0.071656
```

For MOGE Sector, ARIMA (2, 1, 2) is returned.

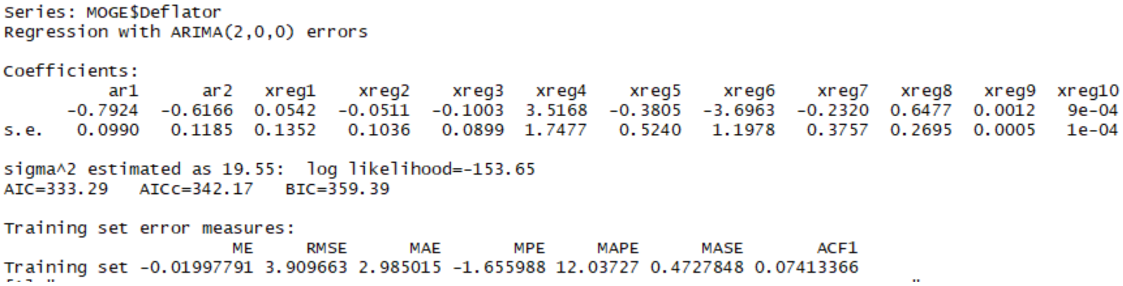
ARIMA with All External:

Use auto.arima with all external regressors.

A list of external regressors are considered: Multifactor productivity (MP), Labour productivity (LP), Capital productivity (CP), Combined labour and capital inputs (CSI), Labour input (LI), Capital input (CI), Labour composition (LCO), Capital composition (CC), Labour compensation (LCE) and Capital Cost(Ccost).

In this model, all external regressors are used without selection.

Table 2: [7]



```
Series: MOGE$Deflator
Regression with ARIMA(2,0,0) errors

Coefficients:
          ar1      ar2   xreg1    xreg2    xreg3   xreg4    xreg5    xreg6    xreg7   xreg8   xreg9  xreg10
      -0.7924  -0.6166  0.0542  -0.0511  -0.1003  3.5168  -0.3805  -3.6963  -0.2320  0.6477  0.0012   9e-04
s.e.   0.0990   0.1185  0.1352   0.1036   0.0899  1.7477   0.5240   1.1978   0.3757  0.2695  0.0005   1e-04

sigma^2 estimated as 19.55:  log likelihood=-153.65
AIC=333.29   AICc=342.17   BIC=359.39

Training set error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set -0.01997791 3.909663 2.985015 -1.655988 12.03727 0.4727848 0.07413366
```

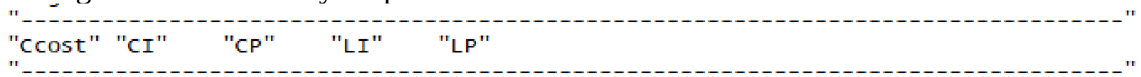
For MOGE Sector, the result is shown above.

Model A:

Do a multilinear regression on the differenced series first (StepAIC is used for model selection), then fit the residual with auto.arima.

This model is developed from Approach A described in Section 3.

The regressors returned by StepAIC is as below:[8]



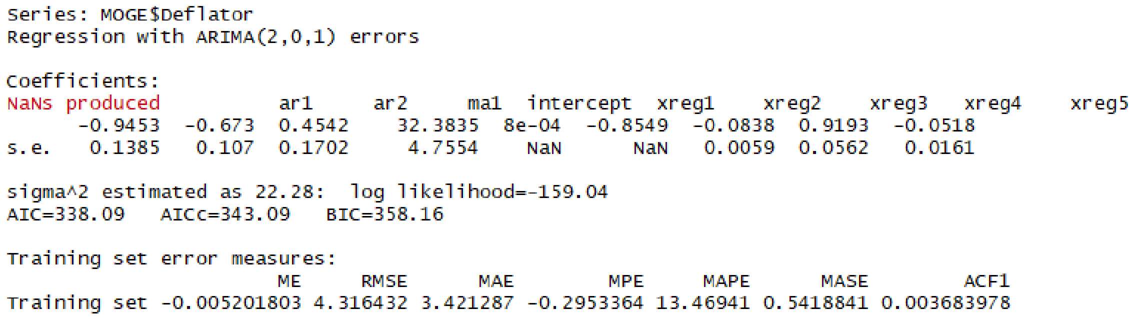
```
"-----------------------------------------------------------------------------"
"Ccost" "CI"    "CP"    "LI"    "LP"
"-----------------------------------------------------------------------------"
```

The model produced with these 5 regressors is as below:

[7] See Appendix C – Code 14 Solution 5

[8] See Appendix C – Code 10 Solution 2

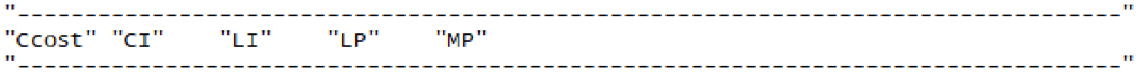
Table 3: [9]



```
Series: MOGE$Deflator
Regression with ARIMA(2,0,1) errors

Coefficients:
NaNs produced          ar1     ar2     ma1  intercept  xreg1    xreg2    xreg3   xreg4    xreg5
      -0.9453  -0.673  0.4542    32.3835  8e-04  -0.8549  -0.0838  0.9193  -0.0518
s.e.   0.1385   0.107  0.1702     4.7554    NaN      NaN   0.0059  0.0562   0.0161

sigma^2 estimated as 22.28:  log likelihood=-159.04
AIC=338.09   AICc=343.09   BIC=358.16

Training set error measures:
                       ME     RMSE      MAE        MPE     MAPE      MASE        ACF1
Training set -0.005201803 4.316432 3.421287 -0.2953364 13.46941 0.5418841 0.003683978
```

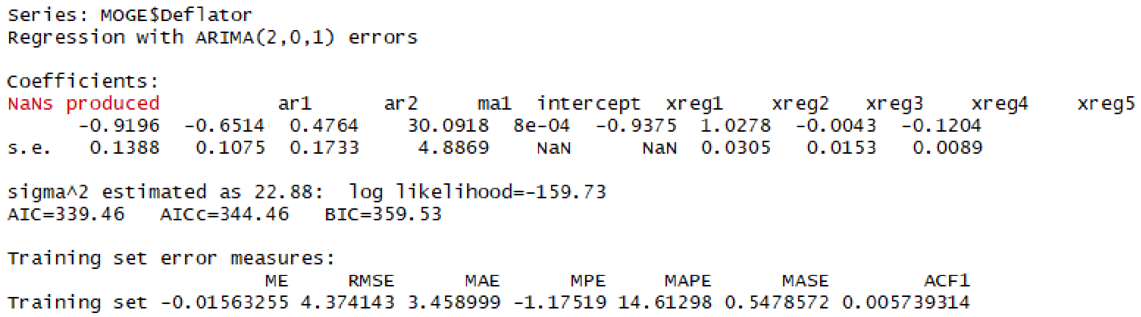
<u>Model B:</u> [10]

Do auto.arima on the differenced series, then apply multilinear regression on the residual (Again, StepAIC is used for model selection)
This model is developed from Approach B described in Section 3.



```
"---------------------------------------------------------------------------------------------"
"Ccost" "CI"    "LI"    "LP"    "MP"
"---------------------------------------------------------------------------------------------"
```

The model produced with these 5 regressors is as below:
Table 4: [11]



```
Series: MOGE$Deflator
Regression with ARIMA(2,0,1) errors

Coefficients:
NaNs produced           ar1      ar2     ma1  intercept  xreg1    xreg2   xreg3    xreg4    xreg5
      -0.9196  -0.6514  0.4764    30.0918  8e-04  -0.9375  1.0278  -0.0043  -0.1204
s.e.   0.1388   0.1075  0.1733     4.8869    NaN      NaN  0.0305   0.0153   0.0089

sigma^2 estimated as 22.88:  log likelihood=-159.73
AIC=339.46   AICc=344.46   BIC=359.53

Training set error measures:
                      ME     RMSE      MAE      MPE     MAPE      MASE        ACF1
Training set -0.01563255 4.374143 3.458999 -1.17519 14.61298 0.5478572 0.005739314
```

<u>Model C:</u>

Take log of the deflator series, and fit the log series with auto.arima
This model is developed from Approach C described in Section 3.

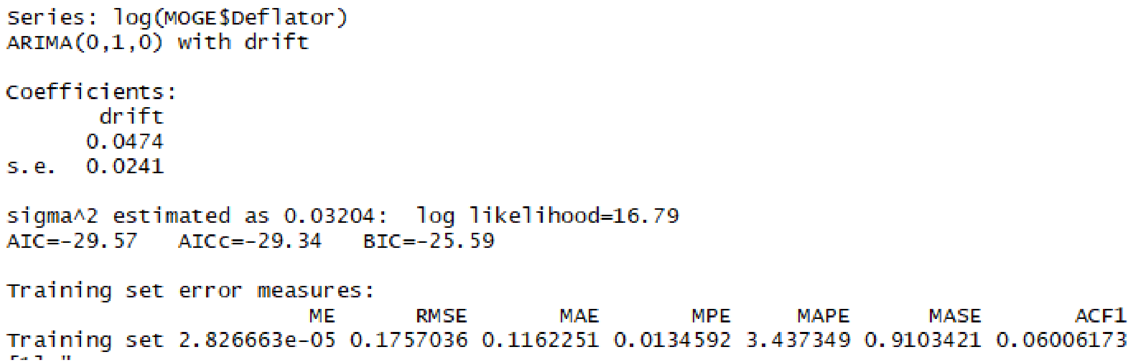
The model produced is as below:
Table 5: [12]

[9] See Appendix C – Code 15 Solution 6

[10] See Appendix C – Code 11 Solution 3

[11] See Appendix C – Code 16 Solution 7

[12] See Appendix C – Code 17 Solution 8



```
Series: log(MOGE$Deflator)
ARIMA(0,1,0) with drift

Coefficients:
       drift
      0.0474
s.e.  0.0241

sigma^2 estimated as 0.03204:  log likelihood=16.79
AIC=-29.57   AICc=-29.34   BIC=-25.59

Training set error measures:
                       ME      RMSE       MAE       MPE     MAPE      MASE       ACF1
Training set 2.826663e-05 0.1757036 0.1162251 0.0134592 3.437349 0.9103421 0.06006173
```

We further used Cross Validation[13] to provide an out-of-sample performance assessment.

All results are summarized in the table below:

Table 6:

MOGE Sector*	AIC	BIC	Residual	tsCV MSPE	tsCV MAPE
ARIMA Only	417.36	427.3	328.4649	132.9285	0.1282342
ARIMA With All External Regressors	333.29	359.39	164.1758	96.3871	0.1176507
Model A	338.09	358.16	188.1708	43.5178	0.0937107
Model B	339.46	359.53	190.245	49.5747	0.0942829
Model C	-29.57	-25.59	323.7368	140.0203	0.1281508

Clearly, Model A significantly outperforms other models. Where ARIMA shows underfit pattern and ARIMA with all regressors shows overfit pattern. Model A find the balanced trade-off between overfit and underfit.

Model Diagnostic

We move on to check the standard residuals left from Model A.

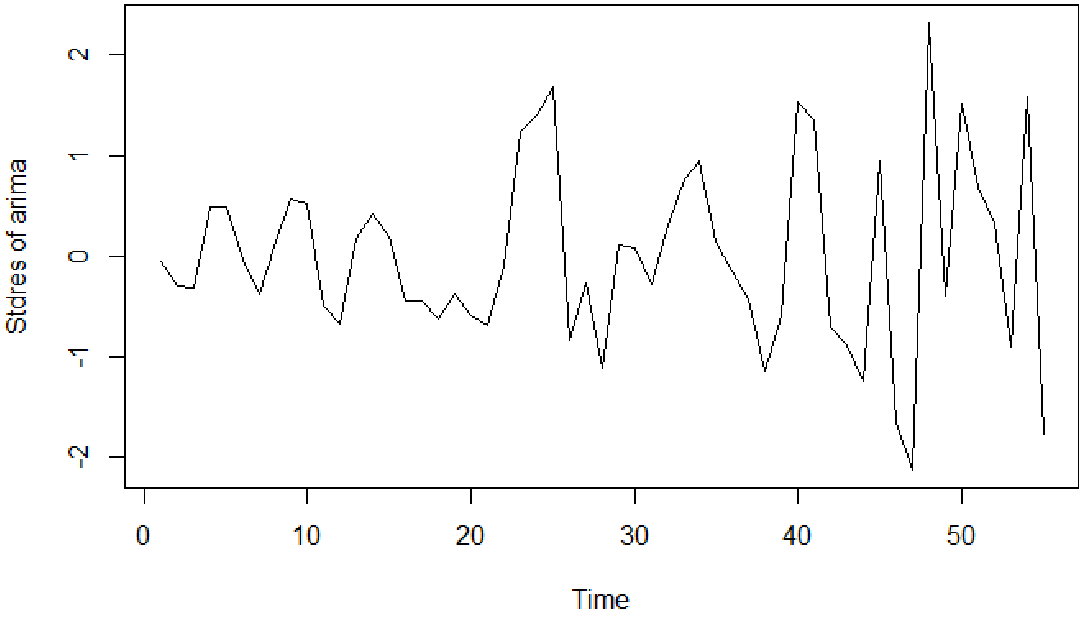
Figure 4[14]: Standard Residual Series

[13] See Appendix B for a note on our Cross Validation

[14] See Appendix C – Code 32 Figure 8

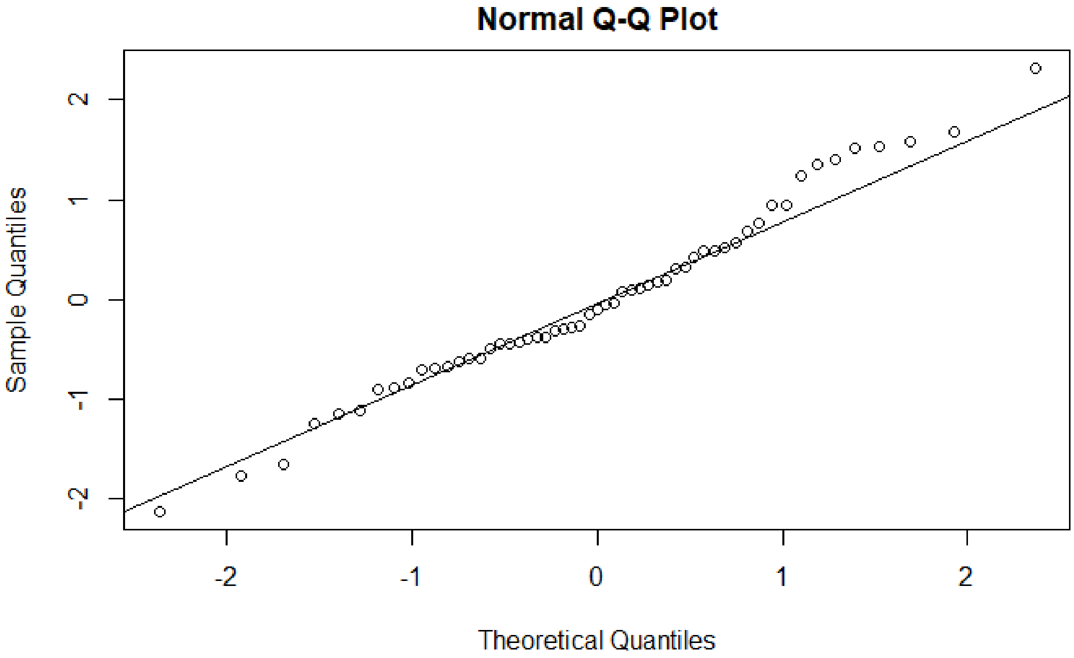
*full external resources include Multifactor productivity, Labour productivity, Capital productivity, Combined labour and capital inputs, Labour input, Capital input, Labour composition, Capital composition, Labour compensation and Capital Cost for this industry Sector

**Arima with full regressor cannot maintain model consistency during pre-set parameters cross validation due to highly correlated data.



Standard residuals appear to be fairly random.

Figure 5[15]: Normal QQ Plot of the Standard Residuals



Normal QQ plot indicates the residuals are distributed approximately normally, with slight fat tail.

[15] See Appendix C – Code 32 Figure 9

Figure 6[16]: ACF Plot of the Standard Residual Series

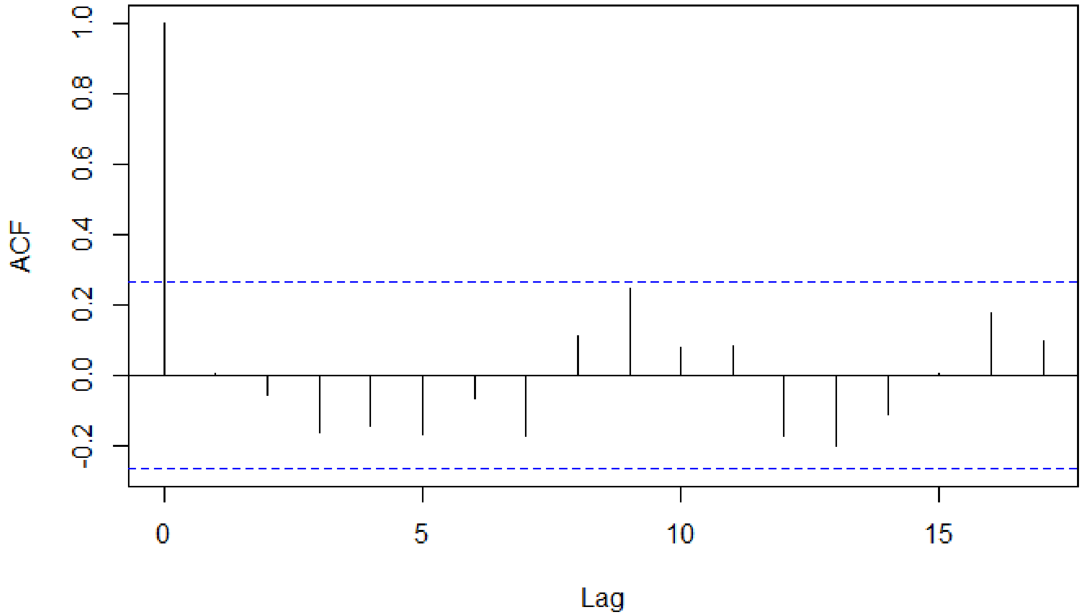
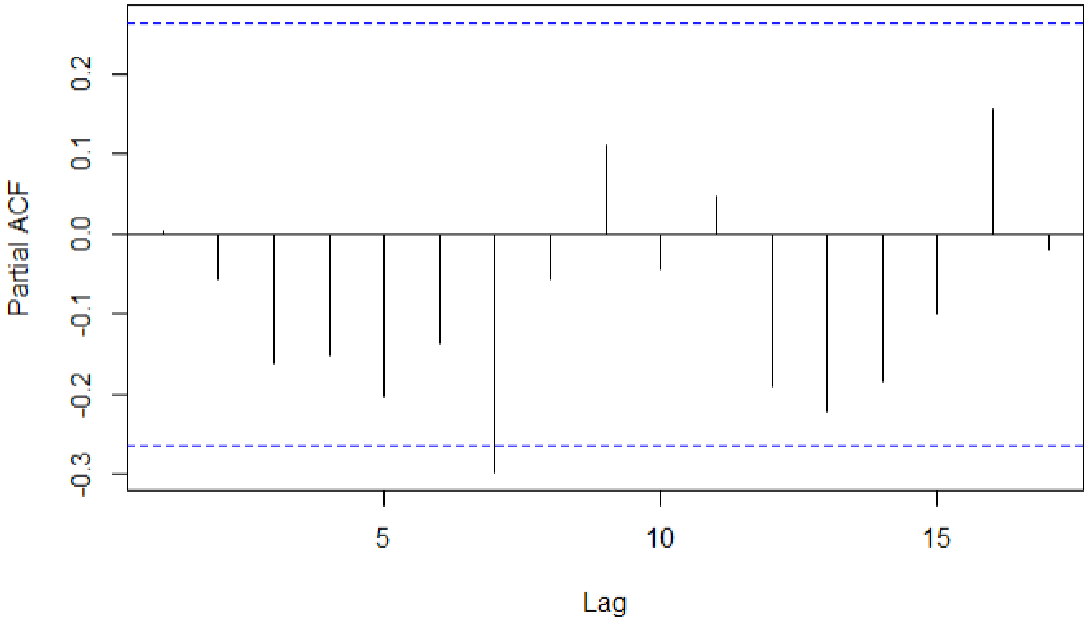


Figure 7[17]: PACF Plot of the Standard Residual Series



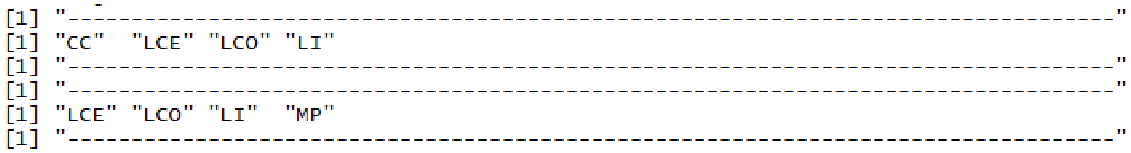
Almost all ACF and PACF's are within the confidence interval. We ignore the slight seasonality. Overall, no suspicious patterns are detected.

[16] See Appendix C – Code 32 Figure 10

[17] See Appendix C – Code 32 Figure 11

We repeated the analysis on other sectors. The results are summarized as below.

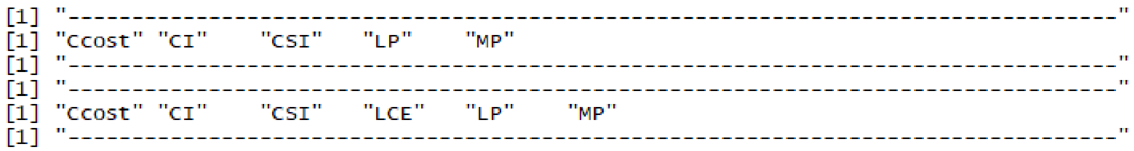
Table 7[18]: The results for Agriculture, Forestry, Fishing and Hunting (AFFH) Sector



```
[1] "--------------------------------------------------------------------------------"
[1] "CC"  "LCE" "LCO" "LI"
[1] "--------------------------------------------------------------------------------"
[1] "--------------------------------------------------------------------------------"
[1] "LCE" "LCO" "LI"  "MP"
[1] "--------------------------------------------------------------------------------"
```

AFFH Sector*	AIC	BIC	Residual	tsCV MSPE	tsCV MAPE
ARIMA Only	339.75	347.71	212.8212	36.69845	0.0791358
ARIMA With all external regressors	342.14	366.23	192.3395	102.7845	0.1342035
Model A	342.96	356.89	207.0629	41.25409	0.0873745
Model B	336.11	352.17	192.4487	39.66413	0.0791207
Model C	-101.43	-93.47	224.45	41.98703	0.0814649

Table 8[19]: The results for Manufacturing (M) Sector



```
[1] "--------------------------------------------------------------------------------"
[1] "Ccost" "CI"    "CSI"   "LP"    "MP"
[1] "--------------------------------------------------------------------------------"
[1] "--------------------------------------------------------------------------------"
[1] "Ccost" "CI"    "CSI"   "LCE"   "LP"    "MP"
[1] "--------------------------------------------------------------------------------"
```

M Sector*	AIC	BIC	Residual	tsCV MSPE	tsCV MAPE
ARIMA Only	218.15	224.12	76.2154	3.514519	0.0233861
ARIMA With all external regressors	225.32	251.42	65.17737	9.459617	0.0376728

[18] See Appendix C – Code 34,35 Solution 19,20

[19] See Appendix C – Code 53,54 Solution 36,37

*full external resources include Multifactor productivity, Labour productivity, Capital productivity, Combined labour and capital inputs, Labour input, Capital input, Labour composition, Capital composition, Labour compensation and Capital Cost for this industry Sector

**Arima with full regressor cannot maintain model consistency during pre-set parameters cross validation due to highly correlated data.

Model A**	230.09	248.16	71.48589	7.958426	0.0345886
Model B**	225.32	251.42	65.17737	6.432225	0.0298882
Model C	-222.71	-216.8	75.05578	4.193989	0.0234752

Tests on AFFH sector and M sector show that Model A and Model B generates close AIC, BIC, residuals, MSPE and MAPE values. We conclude Model A would perform better on Oil and Agriculture related industry, and Model B would perform better on Agriculture related industry. For Manufacturing industry, the pure ARIMA is preferred.

Conclusion

Generally, the introduction of external regressors can help with the prediction of GDP Deflators. For different Sectors, different combinations of regressors would be most predictive.

Interpretation of the model

For MOGE Sector, CI, CP, LI, LP, Ccost are the factors that explained most of the variances of the model.
Moreover, the coefficient of CI, CP, LI and LP is positive, whereas the coefficient of Ccost is negative[20].

This reveals the fact that for Oil industry is sensitive to the change in capital and labour; Which is understandable, since the Oil industry is both labour and capital intensive. On the other hand, a higher Capital Cost would drive down the profit of the business, hence the incentive for production would be lower.

Interesting interpretations can be made on other sectors as well.

Limitation of the model

For AFFH and M sector, our method is not significantly better than simple ARIMA fitting.

This insensitivity to external regressors could be due to the structure of the industries. For example, for MOGE sector, the increase in labour and capital input would be reflected in the short-term, whereas it may take longer for the inputs to take effect on AFFH and M Deflators.

Thus, not all predictions can benefit from our approach. To improve the model performance, more data are needed for training; one could also consider including a richer set of regressors. Beyond that, some essential investigation for the pattern of industry are also required.

[20] See Appendix C –Code 15 Solution 6

Appendix

Appendix A: Definitions & Data Requirements

Definitions:

Time Series (TS): Time series analysis is the collection of data at specific intervals over a period of time, with the purpose of identifying trends, cycles, and seasonal variances to aid in the forecasting of a future event.

White Noise (WN): A time series is white noise if the variables are independent and identically distributed with a mean of zero. Then all variables have the same variance of sigma squared and each value has a zero correlation with all other values in the series.

Lag Operator: A lag 1 autocorrelation (i.e., k = 1) is the correlation between values that are one time period apart. More generally, a lag k autocorrelation is the correlation between values that are k time periods apart.

ACF (auto-correlated function): gives values of auto correlation of any series with its lagged values. It describes how well the present value of the series is related with its past values

PACF (partial auto-correlation function): gives partial correlation of a stationary times series with its own lagged values, regressed the values of the time series at all shorter lags

ARIMA (autoregressive integrated moving average): a forecasting technique that projects the future values of a series based entirely on its own inertia

Groups of interest: (Three industrial groups provided with their real and nominal GDP values.)

1. Mining and Oil and Gas Extraction (MOGE)
2. Agriculture, Forestry, Fishing and Hunting (AFFH)
3. Manufacturing (M)

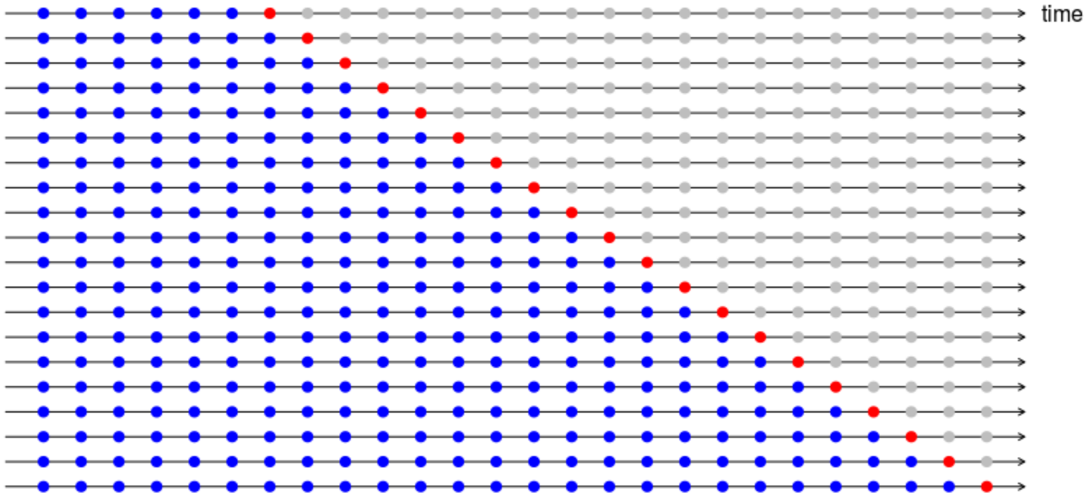
This study will be carried out using data from Statistics Canada. We choose R as our programming language. The ‘cansim’ package is used to extract StatCan time series information. Also, times series are extracted from data tables that contain various economic indicators. After reading from package, we decided to calculate the GDP deflator.

A list of external regressors are considered, namely, Multifactor productivity (MP), Labour productivity (LP), Capital productivity (CP), Combined labour and capital inputs (CSI), Labour input (LI), Capital input (CI), Labour composition(LCO), Capital composition(CC), Labour compensation(LCE), Capital Cost(Ccost).

Appendix B: A note on Cross Validation

Traditionally AIC or BIC is considered for model selection. However, we intend to evaluate the absolute performance of our models. Since AIC and BIC are in sample test, they might be misleading by overfitting. We would introduce cross validation for performance assessment, which could encounter this problem by out sample test.

Time series data is different from the usual data. As there exist high correlations between time lag(s), a traditional leave-one-out validation would have a poor performance, because only the data collected before our test data will affect the model, potentially decrease the amount of data could be used for validating. We could use a specific designed Time Series Cross Validation in "**forecast**" package. The ideology is using leave-one-out based on time lag. This approach overcome the disadvantage of traditional method by using every sample data for validation.



Time Series Cross Validation

Model consistency could become a concern if we have highly correlated features. In another word, the model that predict value at time lag i is different at time lag j using Auto.Arima. If we set the parameters of ARIMA model to be consistent, some optimization errors could appear.

The reasons could be various, the most two possible explanations are correlations and model generalization. We have mentioned the high correlation of some features at the beginning. In optimization, this could lead the xreg matrix to a singular matrix, which is not invertible and throw non optimizable error during Hessian optimization process. Another possibility is the ARIMA model we set is for the whole datasets. It is totally possible it does not fit some partial data at all, which will cause non-optimizable error. Intend to encounter this problem, we could trade the model consistency for a doable validation method; use Auto.Arima instead of pre-set ARIMA model. However, Auto.Arima could only be used to validate the efficiency of xreg, the model effect will be excluded.

An intention to maintain model consistency, the dropping of highly correlated data should be considered in Feature Engineering. The comparison of model with/without xreg shows the xreg and the pre-select significantly improve the performance of our model.

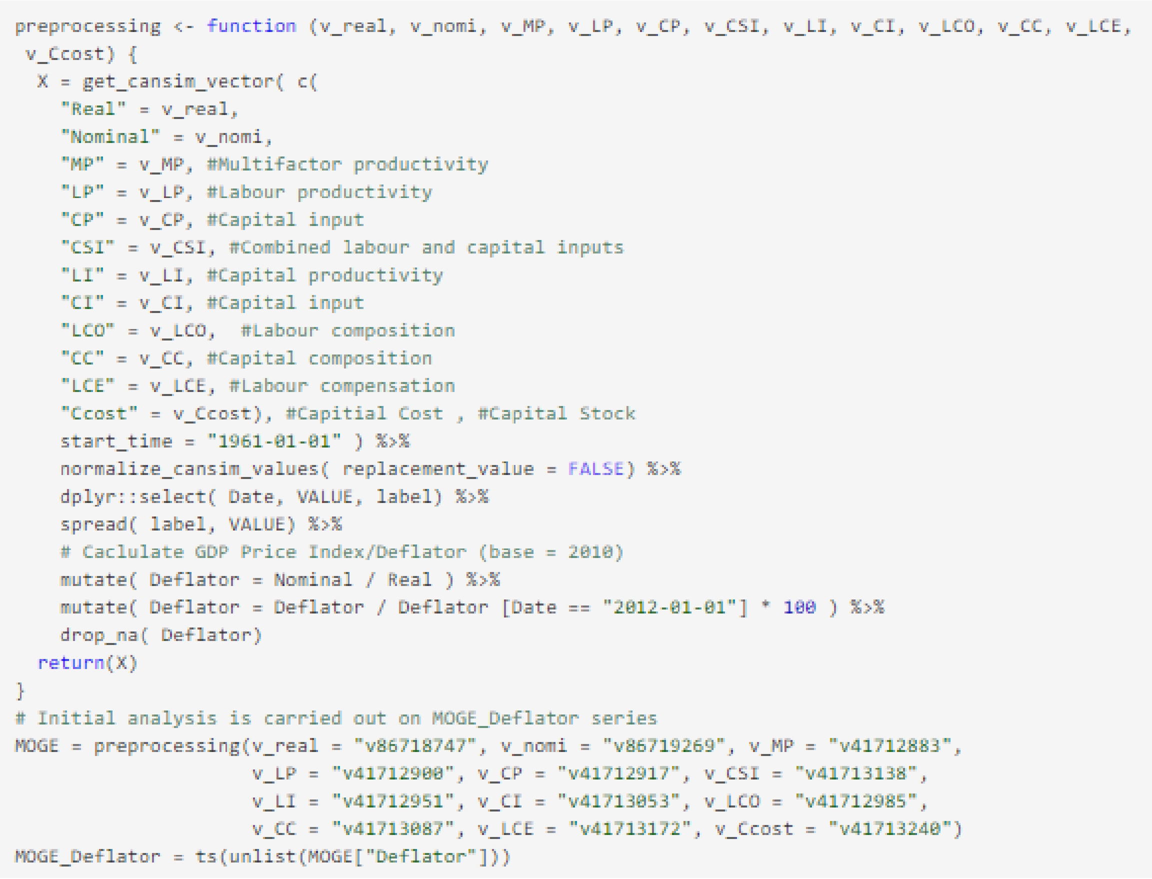
We measure the model performance on three datasets. We use AIC, BIC and In-Sample residual for our train error measurements, and MSPE, MAPE as out test error measurements. Given an example for MOGE Sector, ARIMA shows clear pattern on under fit, ARIMA with full external regressors shows clear pattern for over fit. Our approach balanced trade-off and shows great predict consistency. In general, Approach A is preferred than Approach B for predicting Oil GDP deflator.

Appendix C: Code & Graphs

--- MOGE --

--- Data Preprocessing ---

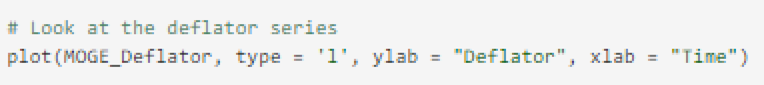
Code 1



```
preprocessing <- function (v_real, v_nomi, v_MP, v_LP, v_CP, v_CSI, v_LI, v_CI, v_LCO, v_CC, v_LCE,
 v_Ccost) {
  X = get_cansim_vector( c(
    "Real" = v_real,
    "Nominal" = v_nomi,
    "MP" = v_MP, #Multifactor productivity
    "LP" = v_LP, #Labour productivity
    "CP" = v_CP, #Capital input
    "CSI" = v_CSI, #Combined labour and capital inputs
    "LI" = v_LI, #Capital productivity
    "CI" = v_CI, #Capital input
    "LCO" = v_LCO,  #Labour composition
    "CC" = v_CC, #Capital composition
    "LCE" = v_LCE, #Labour compensation
    "Ccost" = v_Ccost), #Capitial Cost , #Capital Stock
    start_time = "1961-01-01" ) %>%
    normalize_cansim_values( replacement_value = FALSE) %>%
    dplyr::select( Date, VALUE, label) %>%
    spread( label, VALUE) %>%
    # Caclulate GDP Price Index/Deflator (base = 2010)
    mutate( Deflator = Nominal / Real ) %>%
    mutate( Deflator = Deflator / Deflator [Date == "2012-01-01"] * 100 ) %>%
    drop_na( Deflator)
  return(X)
}
# Initial analysis is carried out on MOGE_Deflator series
MOGE = preprocessing(v_real = "v86718747", v_nomi = "v86719269", v_MP = "v41712883",
                     v_LP = "v41712900", v_CP = "v41712917", v_CSI = "v41713138",
                     v_LI = "v41712951", v_CI = "v41713053", v_LCO = "v41712985",
                     v_CC = "v41713087", v_LCE = "v41713172", v_Ccost = "v41713240")
MOGE_Deflator = ts(unlist(MOGE["Deflator"]))
```

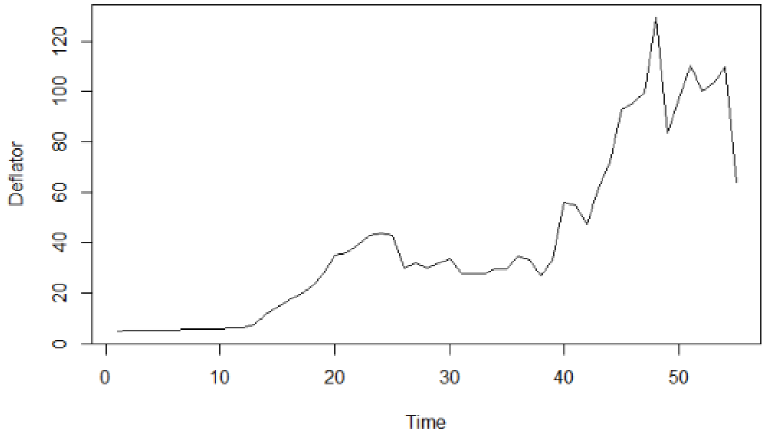
--- INITIAL ANALYSIS ---

Code 2: Plot MOGE Sector deflator series (Figure 1)

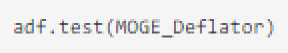


```
# Look at the deflator series
plot(MOGE_Deflator, type = 'l', ylab = "Deflator", xlab = "Time")
```

Figure 1: Line plot of MOGE Sector deflator with Deflator vs Time

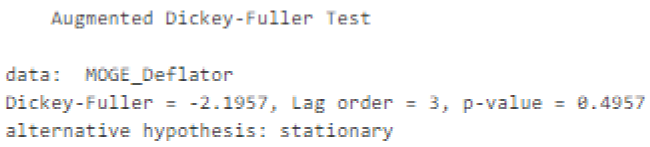


Code 3: Augmented Dickey-Filler Test for MOGE deflator



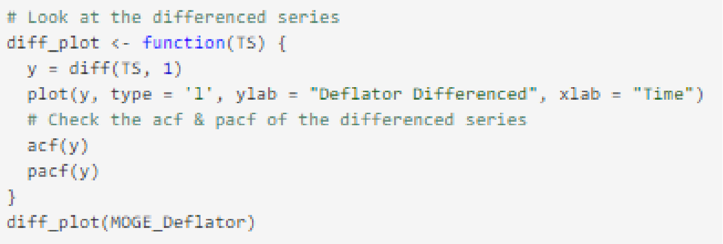
```
adf.test(MOGE_Deflator)
```

Solution 1: R solution of Code 3



```
	Augmented Dickey-Fuller Test

data:  MOGE_Deflator
Dickey-Fuller = -2.1957, Lag order = 3, p-value = 0.4957
alternative hypothesis: stationary
```

Code 4: Differentiate Time Series Analysis and test ACF and PACF (Figure 2, 3, 4)



```
# Look at the differenced series
diff_plot <- function(TS) {
  y = diff(TS, 1)
  plot(y, type = 'l', ylab = "Deflator Differenced", xlab = "Time")
  # Check the acf & pacf of the differenced series
  acf(y)
  pacf(y)
}
diff_plot(MOGE_Deflator)
```

Figure 2: Differenced MOGE Sector with Deflator Differenced vs Time

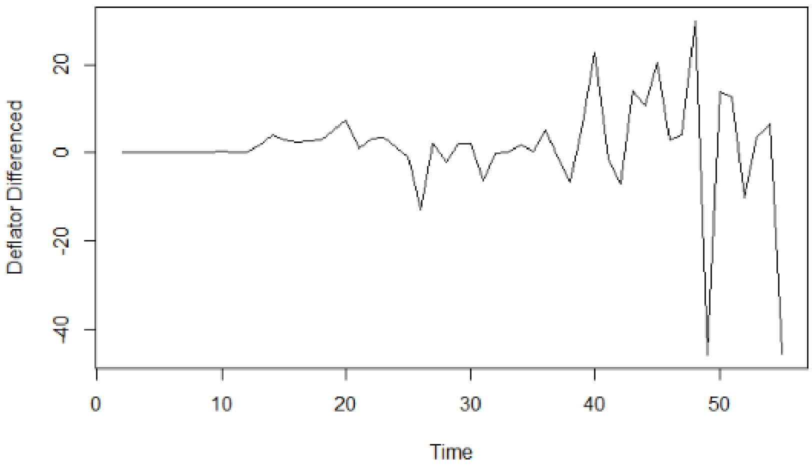


Figure 3: Diff MOGE Sector ACF plot with ACF vs Lag

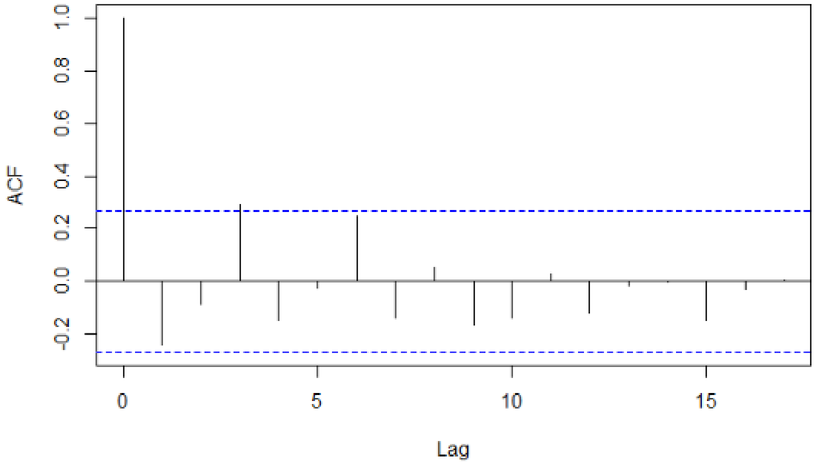
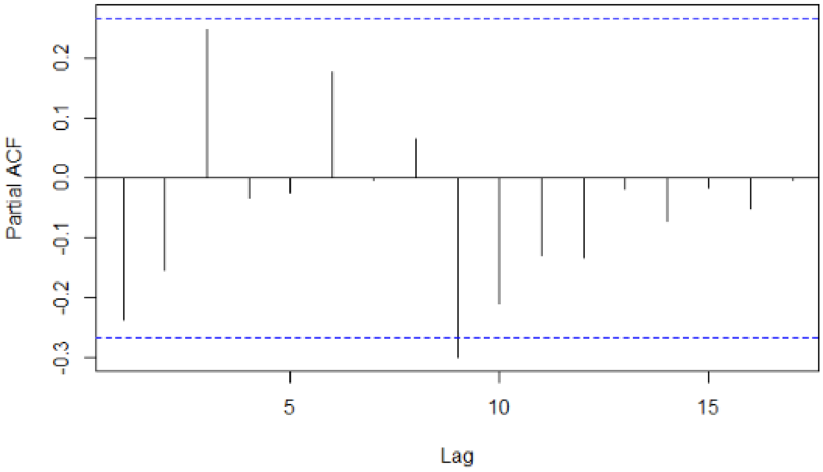
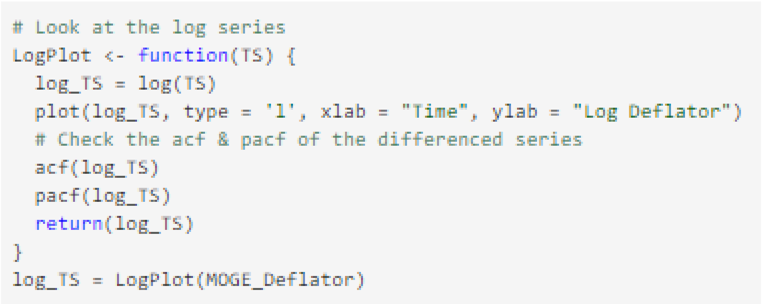


Figure 4: Diff MOGE Sector PACF plot with PACF vs Lag



Code 5: Log Series approach for MOGE Sector, plot Log Time Series for MOGE Deflator (Figure 5), and test ACF and PACF (Figure 6, 7)



```
# Look at the log series
LogPlot <- function(TS) {
  log_TS = log(TS)
  plot(log_TS, type = 'l', xlab = "Time", ylab = "Log Deflator")
  # Check the acf & pacf of the differenced series
  acf(log_TS)
  pacf(log_TS)
  return(log_TS)
}
log_TS = LogPlot(MOGE_Deflator)
```

Figure 5: Log MOGE Sector with Log Deflator vs Time

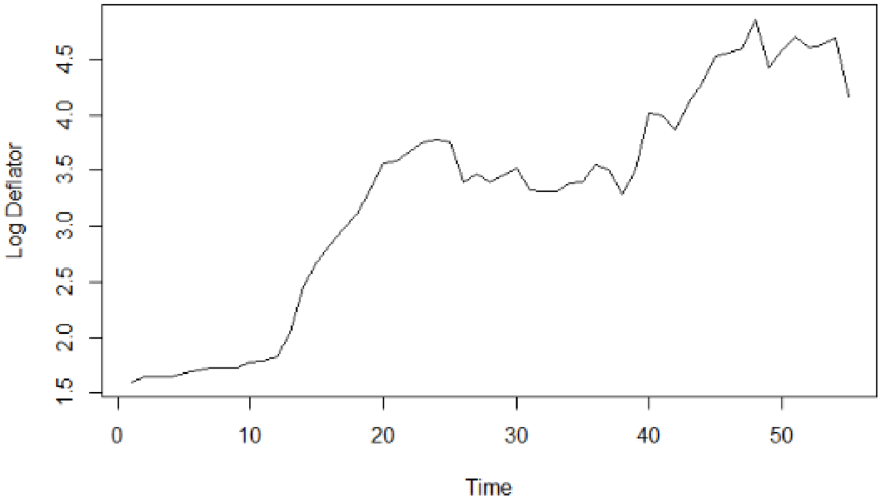


Figure 6: Log MOGE Sector ACF plot with ACF vs Lag

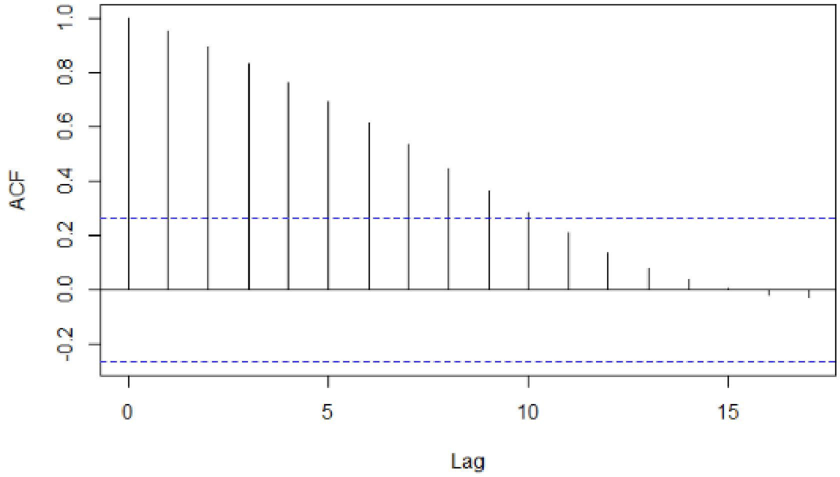
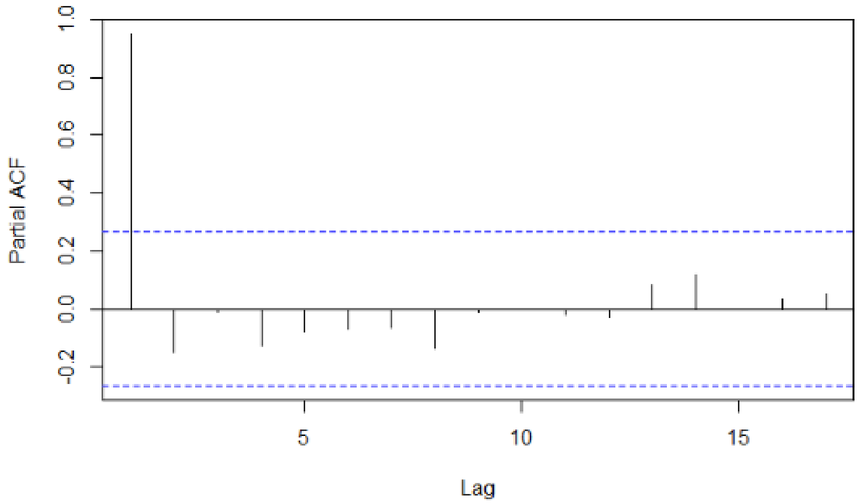
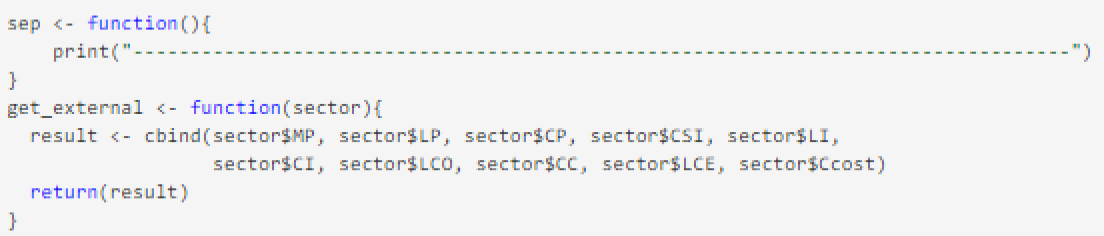


Figure 7: Log MOGE Sector PACF plot with PACF vs Lag



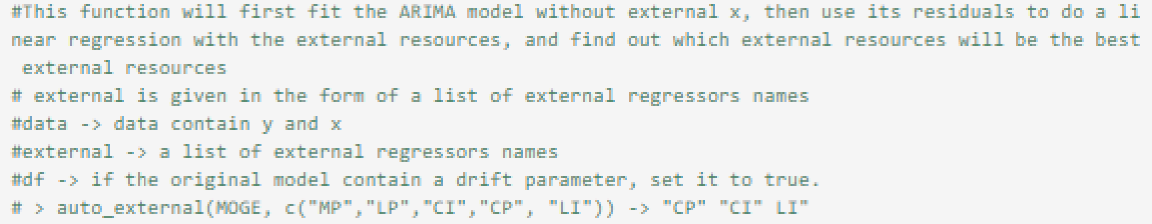
--- MODEL SELECTION ---

Code 6: Fit all external regressors to MOGE Sector

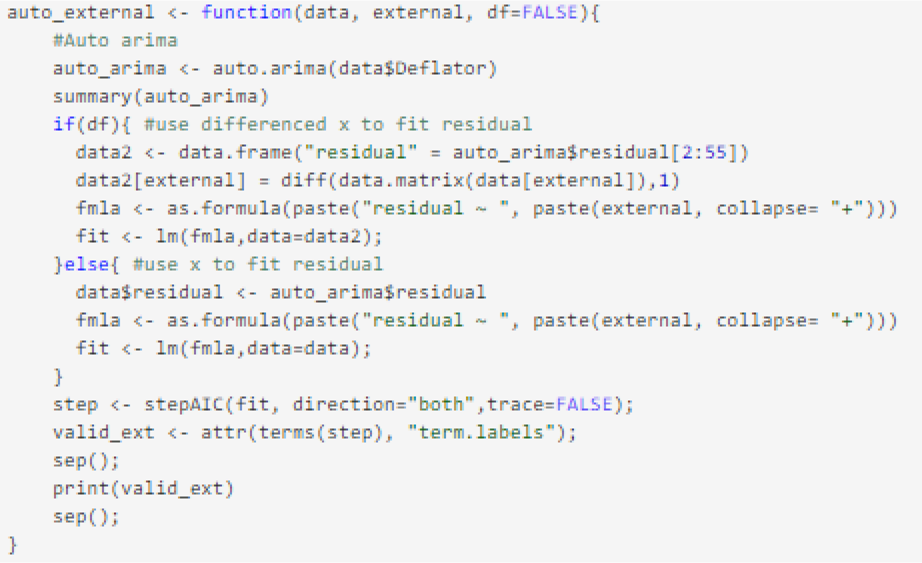


```
sep <- function(){
    print("-----------------------------------------------------------------------------------")
}
get_external <- function(sector){
  result <- cbind(sector$MP, sector$LP, sector$CP, sector$CSI, sector$LI,
                  sector$CI, sector$LCO, sector$CC, sector$LCE, sector$Ccost)
  return(result)
}
```

Code 7: auto.arima function & look for valid externals

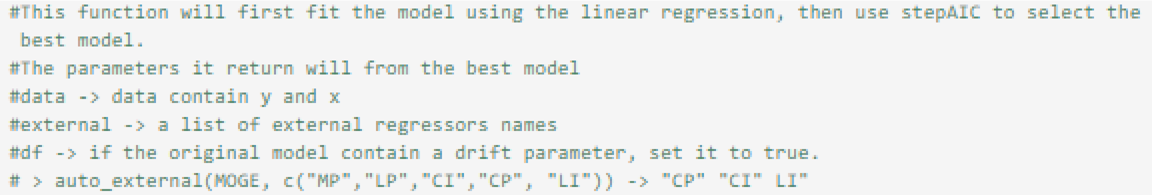


```
#This function will first fit the ARIMA model without external x, then use its residuals to do a li
near regression with the external resources, and find out which external resources will be the best
 external resources
# external is given in the form of a list of external regressors names
#data -> data contain y and x
#external -> a list of external regressors names
#df -> if the original model contain a drift parameter, set it to true.
# > auto_external(MOGE, c("MP","LP","CI","CP", "LI")) -> "CP" "CI" LI"
```

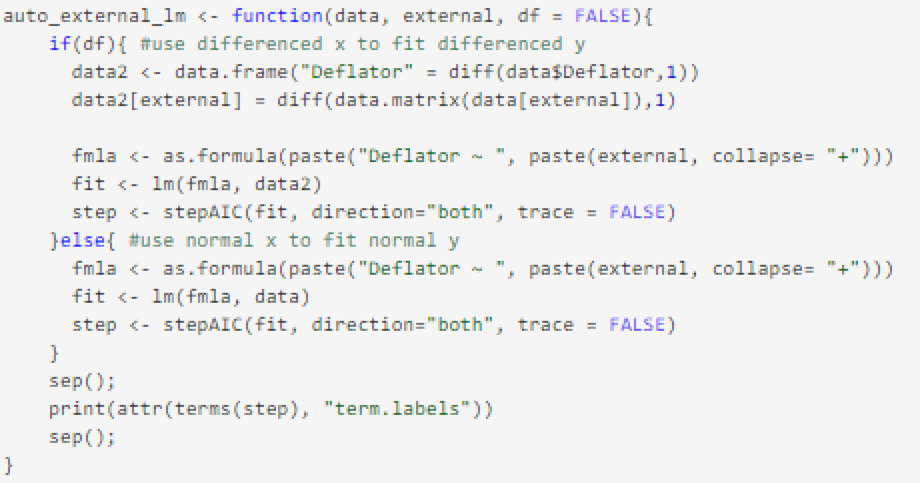


```
auto_external <- function(data, external, df=FALSE){
    #Auto arima
    auto_arima <- auto.arima(data$Deflator)
    summary(auto_arima)
    if(df){ #use differenced x to fit residual
      data2 <- data.frame("residual" = auto_arima$residual[2:55])
      data2[external] = diff(data.matrix(data[external]),1)
      fmla <- as.formula(paste("residual ~ ", paste(external, collapse= "+")))
      fit <- lm(fmla,data=data2);
    }else{ #use x to fit residual
      data$residual <- auto_arima$residual
      fmla <- as.formula(paste("residual ~ ", paste(external, collapse= "+")))
      fit <- lm(fmla,data=data);
    }
    step <- stepAIC(fit, direction="both",trace=FALSE);
    valid_ext <- attr(terms(step), "term.labels");
    sep();
    print(valid_ext)
    sep();
}
```

Code 8: Auto external function

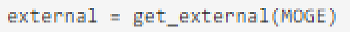


```
#This function will first fit the model using the linear regression, then use stepAIC to select the
 best model.
#The parameters it return will from the best model
#data -> data contain y and x
#external -> a list of external regressors names
#df -> if the original model contain a drift parameter, set it to true.
# > auto_external(MOGE, c("MP","LP","CI","CP", "LI")) -> "CP" "CI" LI"
```



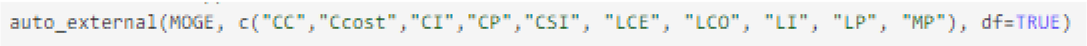
```
auto_external_lm <- function(data, external, df = FALSE){
    if(df){ #use differenced x to fit differenced y
      data2 <- data.frame("Deflator" = diff(data$Deflator,1))
      data2[external] = diff(data.matrix(data[external]),1)

      fmla <- as.formula(paste("Deflator ~ ", paste(external, collapse= "+")))
      fit <- lm(fmla, data2)
      step <- stepAIC(fit, direction="both", trace = FALSE)
    }else{ #use normal x to fit normal y
      fmla <- as.formula(paste("Deflator ~ ", paste(external, collapse= "+")))
      fit <- lm(fmla, data)
      step <- stepAIC(fit, direction="both", trace = FALSE)
    }
    sep();
    print(attr(terms(step), "term.labels"))
    sep();
}
```

Code 9: External resources of MOGE Sector



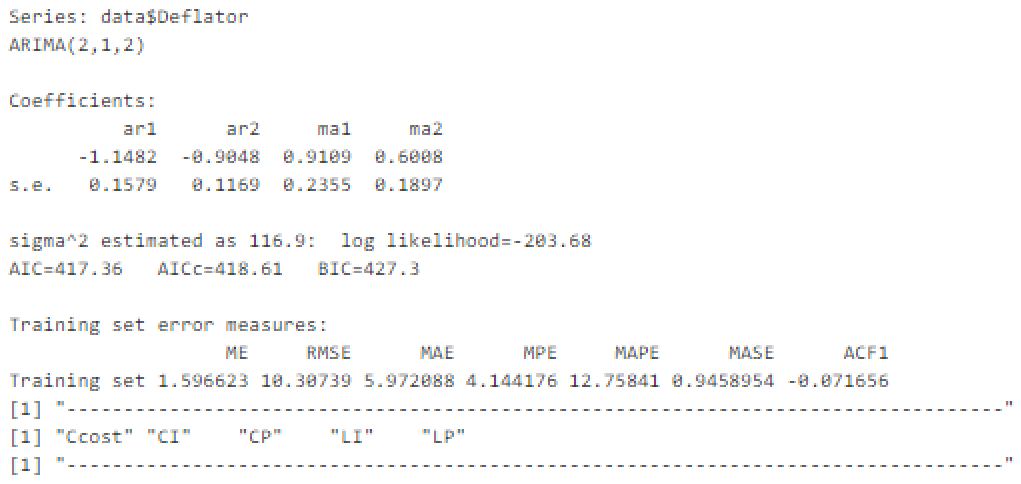
```
external = get_external(MOGE)
```

Code 10: Auto selection Approach A to return true parameters (Solution 2)



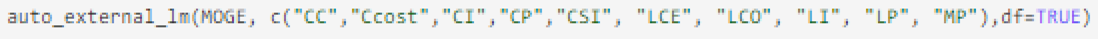
```
auto_external(MOGE, c("CC","Ccost","CI","CP","CSI", "LCE", "LCO", "LI", "LP", "MP"), df=TRUE)
```

Solution 2: Auto selection Approach A to return Ccost, CI, CP, LI, LP



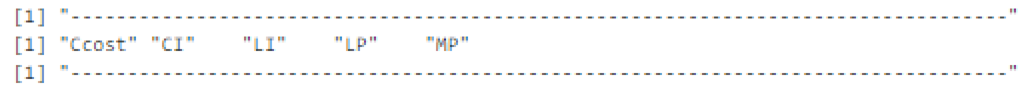
```
Series: data$Deflator
ARIMA(2,1,2)

Coefficients:
          ar1      ar2     ma1     ma2
      -1.1482  -0.9048  0.9109  0.6008
s.e.   0.1579   0.1169  0.2355  0.1897

sigma^2 estimated as 116.9:  log likelihood=-203.68
AIC=417.36   AICc=418.61   BIC=427.3

Training set error measures:
                   ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
Training set 1.596623 10.30739 5.972088 4.144176 12.75841 0.9458954 -0.071656
[1] "----------------------------------------------------------------------------------"
[1] "Ccost" "CI"    "CP"    "LI"    "LP"
[1] "----------------------------------------------------------------------------------"
```

Code 11: Auto selection Approach B to return true parameters (Solution 3)



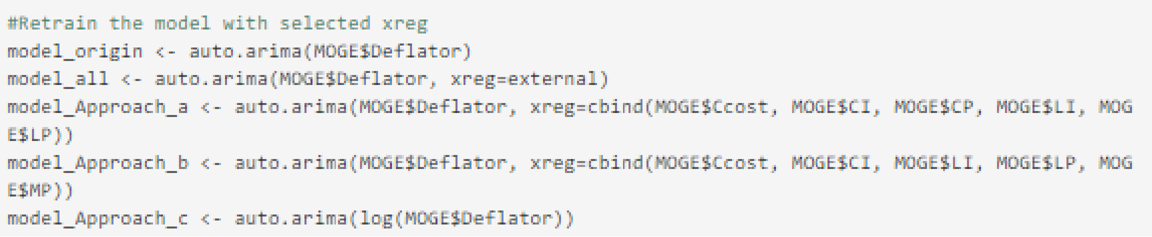
```
auto_external_lm(MOGE, c("CC","Ccost","CI","CP","CSI", "LCE", "LCO", "LI", "LP", "MP"),df=TRUE)
```

Solution 3: Auto selection Approach B to return Ccost, CI, LI, LP, MP



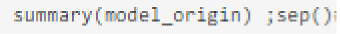
```
[1] "----------------------------------------------------------------------------------"
[1] "Ccost" "CI"    "LI"    "LP"    "MP"
[1] "----------------------------------------------------------------------------------"
```

Code 12: **In Sample Testing** of all model approach bonded with MOGE deflator and true regressors from each approach (Solution 4)



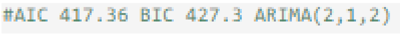
```
#Retrain the model with selected xreg
model_origin <- auto.arima(MOGE$Deflator)
model_all <- auto.arima(MOGE$Deflator, xreg=external)
model_Approach_a <- auto.arima(MOGE$Deflator, xreg=cbind(MOGE$Ccost, MOGE$CI, MOGE$CP, MOGE$LI, MOG
E$LP))
model_Approach_b <- auto.arima(MOGE$Deflator, xreg=cbind(MOGE$Ccost, MOGE$CI, MOGE$LI, MOGE$LP, MOG
E$MP))
model_Approach_c <- auto.arima(log(MOGE$Deflator))
```

Code 13: Original Model Summary

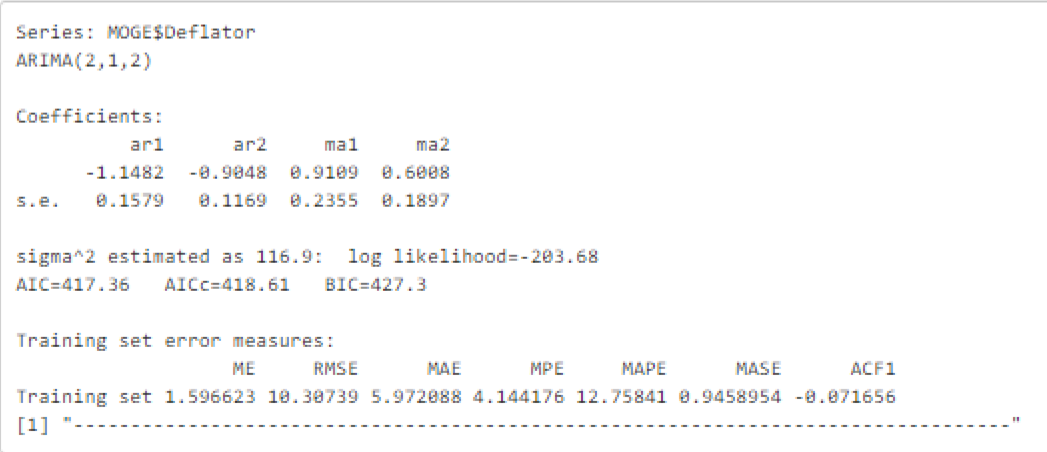


```
summary(model_origin) ;sep()
```

Solution 4: Code 13 results

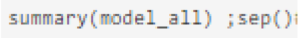


```
#AIC 417.36 BIC 427.3 ARIMA(2,1,2)
```



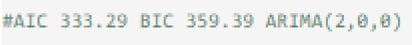
```
Series: MOGE$Deflator
ARIMA(2,1,2)

Coefficients:
          ar1      ar2     ma1     ma2
      -1.1482  -0.9048  0.9109  0.6008
s.e.   0.1579   0.1169  0.2355  0.1897

sigma^2 estimated as 116.9:  log likelihood=-203.68
AIC=417.36   AICc=418.61   BIC=427.3

Training set error measures:
                   ME     RMSE      MAE      MPE     MAPE      MASE      ACF1
Training set 1.596623 10.30739 5.972088 4.144176 12.75841 0.9458954 -0.071656
[1] "---------------------------------------------------------------------------------"
```

Code 14: All MOGE Deflator model summary

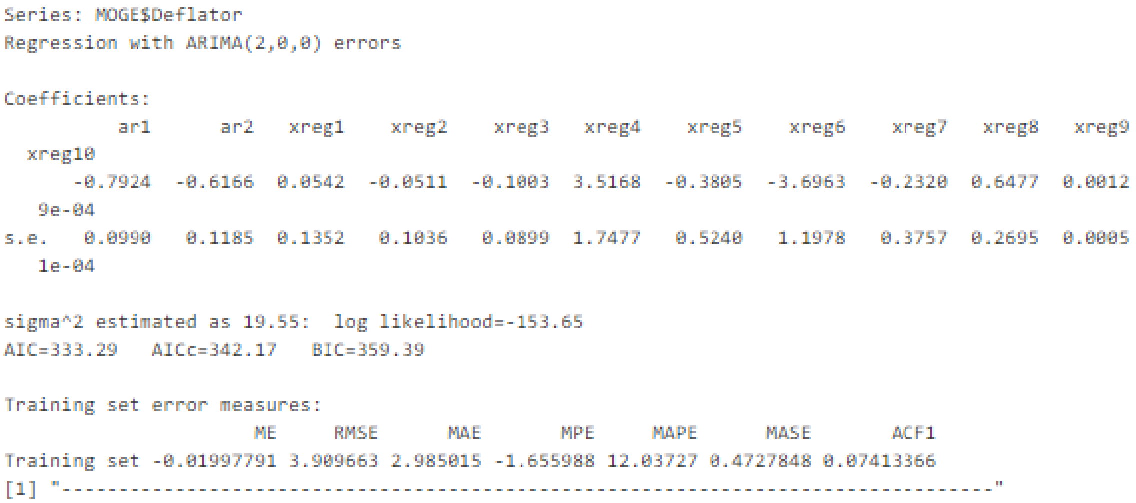


```
summary(model_all) ;sep()
```

Solution 5: Code 14 results

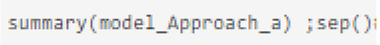


```
#AIC 333.29 BIC 359.39 ARIMA(2,0,0)
```



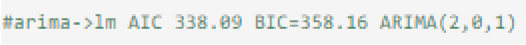
```
Series: MOGE$Deflator
Regression with ARIMA(2,0,0) errors

Coefficients:
          ar1      ar2   xreg1     xreg2     xreg3   xreg4     xreg5     xreg6     xreg7   xreg8   xreg9
  xreg10
      -0.7924  -0.6166  0.0542  -0.0511  -0.1003  3.5168  -0.3805  -3.6963  -0.2320  0.6477  0.0012
   9e-04
s.e.   0.0990   0.1185  0.1352   0.1036   0.0899  1.7477   0.5240   1.1978   0.3757  0.2695  0.0005
   1e-04

sigma^2 estimated as 19.55:  log likelihood=-153.65
AIC=333.29   AICc=342.17   BIC=359.39

Training set error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set -0.01997791 3.909663 2.985015 -1.655988 12.03727 0.4727848 0.07413366
[1] "---------------------------------------------------------------------------------"
```

Code 15: Approach A summary

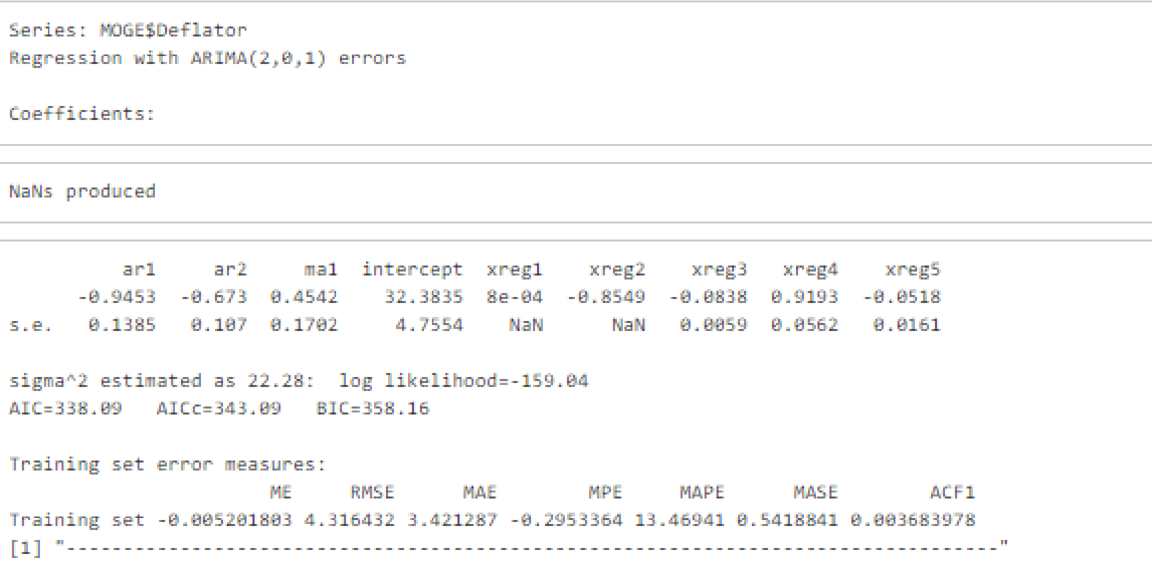


```
summary(model_Approach_a) ;sep()
```

Solution 6: Code 15 results



```
#arima->lm AIC 338.09 BIC=358.16 ARIMA(2,0,1)
```



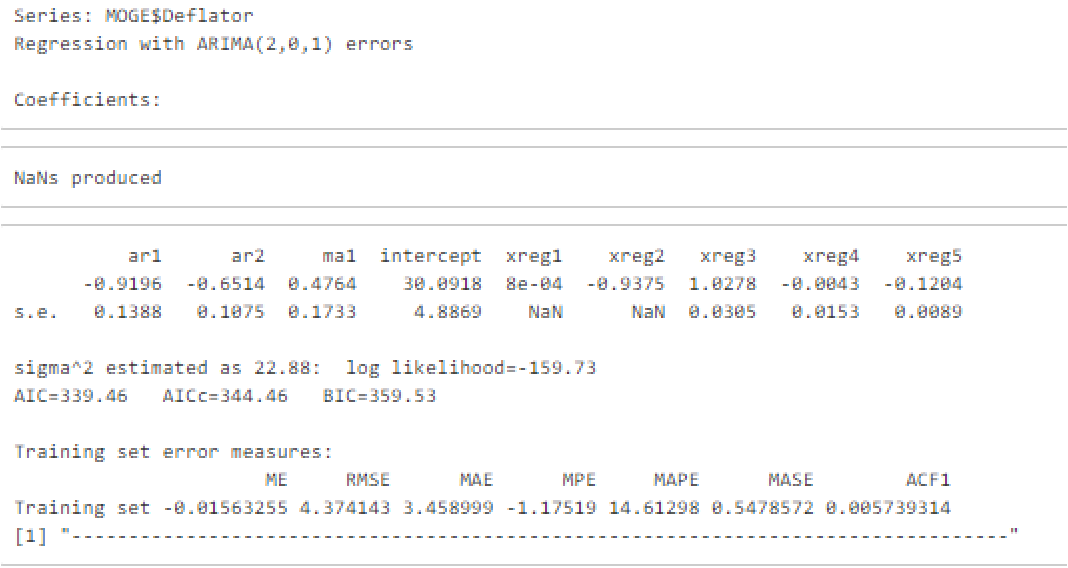
```
Series: MOGE$Deflator
Regression with ARIMA(2,0,1) errors

Coefficients:

NaNs produced

          ar1     ar2     ma1  intercept  xreg1    xreg2    xreg3   xreg4    xreg5
      -0.9453  -0.673  0.4542    32.3835  8e-04  -0.8549  -0.0838  0.9193  -0.0518
s.e.   0.1385   0.107  0.1702     4.7554    NaN      NaN   0.0059  0.0562   0.0161

sigma^2 estimated as 22.28:  log likelihood=-159.04
AIC=338.09   AICc=343.09   BIC=358.16

Training set error measures:
                       ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
Training set -0.005201803 4.316432 3.421287 -0.2953364 13.46941 0.5418841 0.003683978
[1] "--------------------------------------------------------------------------------"
```

Code 16: Approach B summary

```
summary(model_Approach_b) ;sep()#
```

Solution 7: Code 16 results

```
#lm->arima AIC 339.46 BIC=359.53 ARIMA(2,0,1)
```



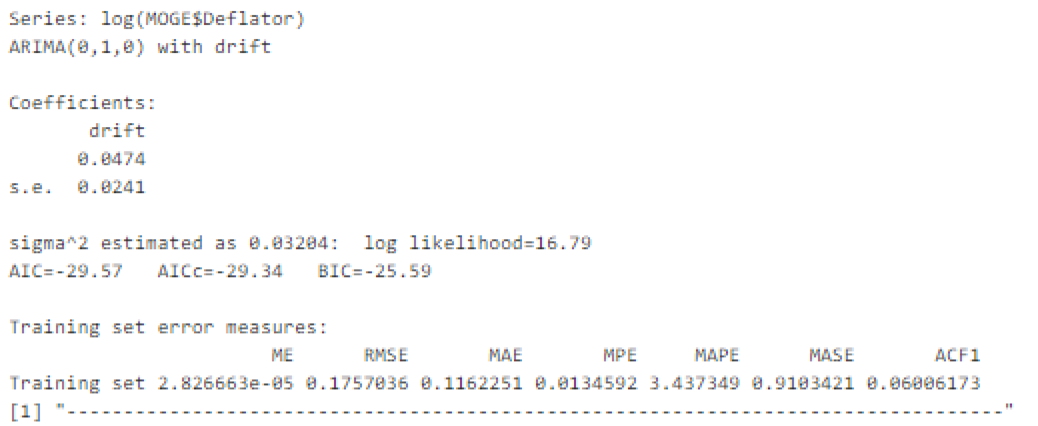
```
Series: MOGE$Deflator
Regression with ARIMA(2,0,1) errors

Coefficients:

NaNs produced

          ar1      ar2     ma1  intercept  xreg1    xreg2   xreg3    xreg4    xreg5
      -0.9196  -0.6514  0.4764    30.0918  8e-04  -0.9375  1.0278  -0.0043  -0.1204
s.e.   0.1388   0.1075  0.1733     4.8869    NaN      NaN  0.0305   0.0153   0.0089

sigma^2 estimated as 22.88:  log likelihood=-159.73
AIC=339.46   AICc=344.46   BIC=359.53

Training set error measures:
                      ME     RMSE      MAE      MPE     MAPE      MASE         ACF1
Training set -0.01563255 4.374143 3.458999 -1.17519 14.61298 0.5478572 0.005739314
[1] "--------------------------------------------------------------------------------"
```

Code 17: Approach C summary

```
summary(model_Approach_c) ;sep()#
```

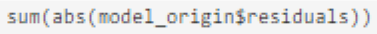
Solution 8: Code 17 results

```
#log->arima AIC -29.57 BIC=-25.59 ARIMA(0,1,0)
```



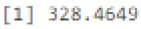
```
Series: log(MOGE$Deflator)
ARIMA(0,1,0) with drift

Coefficients:
       drift
      0.0474
s.e.  0.0241

sigma^2 estimated as 0.03204:  log likelihood=16.79
AIC=-29.57   AICc=-29.34   BIC=-25.59

Training set error measures:
                       ME      RMSE       MAE       MPE     MAPE      MASE       ACF1
Training set 2.826663e-05 0.1757036 0.1162251 0.0134592 3.437349 0.9103421 0.06006173
[1] "-------------------------------------------------------------------------------------"
```

Code 18: Sum of absolute values of all residuals from original MOGE model set



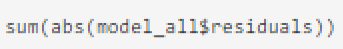
```
sum(abs(model_origin$residuals))
```

Solution 9: Code 18 results



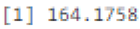
```
[1] 328.4649
```

Code 19: Sum of absolute values of all residuals from all MOGE model set



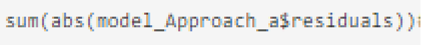
```
sum(abs(model_all$residuals))
```

Solution 10: Code 19 results



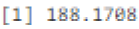
```
[1] 164.1758
```

Code 20: Sum of absolute values of all residuals from Approach A model



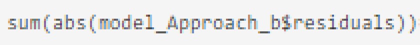
```
sum(abs(model_Approach_a$residuals))
```

Solution 11: Code 20 results



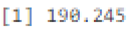
```
[1] 188.1708
```

Code 21: Sum of absolute values of all residuals from Approach B model



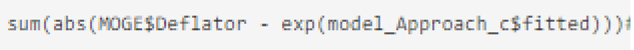
```
sum(abs(model_Approach_b$residuals))
```

Solution 12: Code 21 results



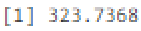
```
[1] 190.245
```

Code 22: Sum of absolute values, differenced between MOGE Deflator and fitted Approach C



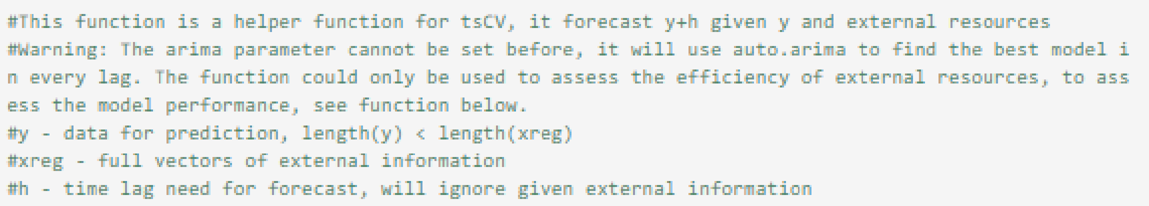
```
sum(abs(MOGE$Deflator - exp(model_Approach_c$fitted)))
```

Solution 13: Code 22 results

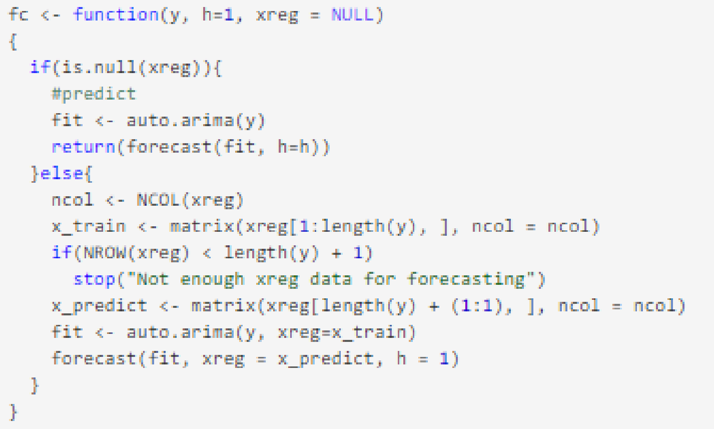


```
[1] 323.7368
```

Code 23: Out of Sample Testing

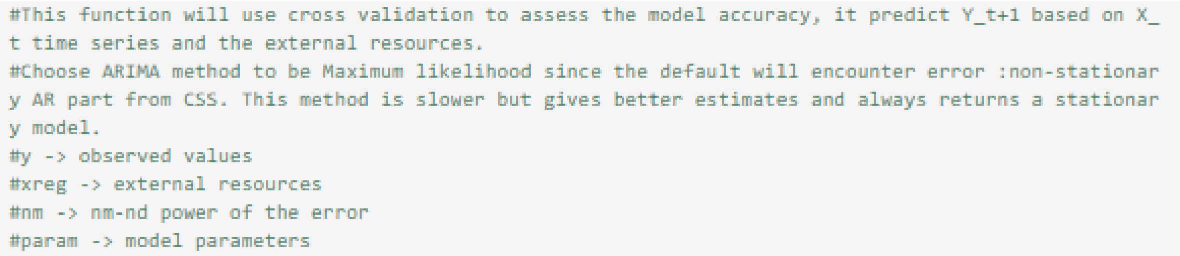


```
#This function is a helper function for tsCV, it forecast y+h given y and external resources
#Warning: The arima parameter cannot be set before, it will use auto.arima to find the best model in every lag. The function could only be used to assess the efficiency of external resources, to assess the model performance, see function below.
#y - data for prediction, length(y) < length(xreg)
#xreg - full vectors of external information
#h - time lag need for forecast, will ignore given external information
```

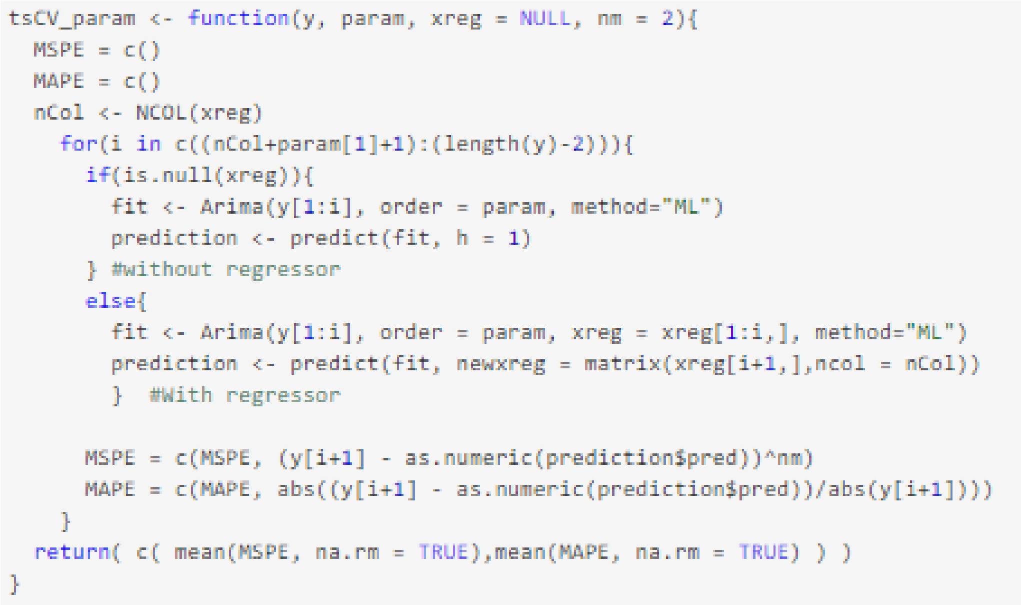


```
fc <- function(y, h=1, xreg = NULL)
{
  if(is.null(xreg)){
    #predict
    fit <- auto.arima(y)
    return(forecast(fit, h=h))
  }else{
    ncol <- NCOL(xreg)
    x_train <- matrix(xreg[1:length(y), ], ncol = ncol)
    if(NROW(xreg) < length(y) + 1)
      stop("Not enough xreg data for forecasting")
    x_predict <- matrix(xreg[length(y) + (1:1), ], ncol = ncol)
    fit <- auto.arima(y, xreg=x_train)
    forecast(fit, xreg = x_predict, h = 1)
  }
}
```

Code 24:

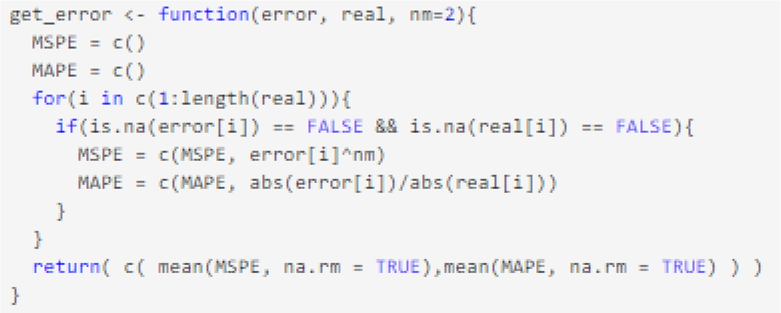


```
#This function will use cross validation to assess the model accuracy, it predict Y_t+1 based on X_t time series and the external resources.
#Choose ARIMA method to be Maximum likelihood since the default will encounter error :non-stationary AR part from CSS. This method is slower but gives better estimates and always returns a stationary model.
#y -> observed values
#xreg -> external resources
#nm -> nm-nd power of the error
#param -> model parameters
```



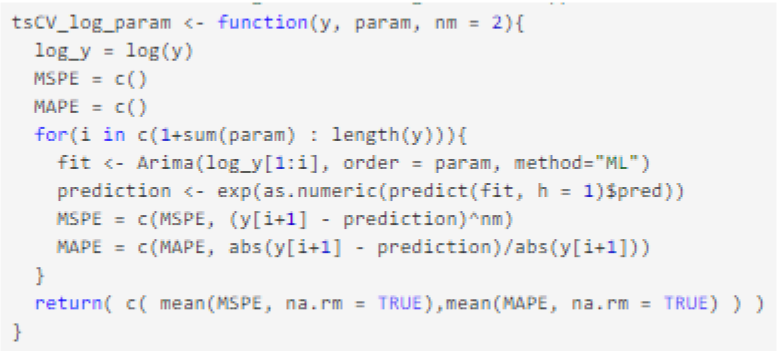
```
tsCV_param <- function(y, param, xreg = NULL, nm = 2){
  MSPE = c()
  MAPE = c()
  nCol <- NCOL(xreg)
    for(i in c((nCol+param[1]+1):(length(y)-2))){
      if(is.null(xreg)){
        fit <- Arima(y[1:i], order = param, method="ML")
        prediction <- predict(fit, h = 1)
      } #without regressor
      else{
        fit <- Arima(y[1:i], order = param, xreg = xreg[1:i,], method="ML")
        prediction <- predict(fit, newxreg = matrix(xreg[i+1,],ncol = nCol))
        }  #With regressor

      MSPE = c(MSPE, (y[i+1] - as.numeric(prediction$pred))^nm)
      MAPE = c(MAPE, abs((y[i+1] - as.numeric(prediction$pred))/abs(y[i+1])))
    }
  return( c( mean(MSPE, na.rm = TRUE),mean(MAPE, na.rm = TRUE) ) )
}
```

Code 25: Helper function for tsCV result



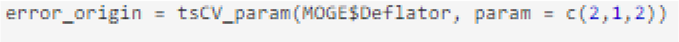
```r
get_error <- function(error, real, nm=2){
  MSPE = c()
  MAPE = c()
  for(i in c(1:length(real))){
    if(is.na(error[i]) == FALSE && is.na(real[i]) == FALSE){
      MSPE = c(MSPE, error[i]^nm)
      MAPE = c(MAPE, abs(error[i])/abs(real[i]))
    }
  }
  return( c( mean(MSPE, na.rm = TRUE),mean(MAPE, na.rm = TRUE) ) )
}
```

Code 26: Log error calculator while doing an ARIMA approach

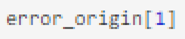


```r
tsCV_log_param <- function(y, param, nm = 2){
  log_y = log(y)
  MSPE = c()
  MAPE = c()
  for(i in c(1+sum(param) : length(y))){
    fit <- Arima(log_y[1:i], order = param, method="ML")
    prediction <- exp(as.numeric(predict(fit, h = 1)$pred))
    MSPE = c(MSPE, (y[i+1] - prediction)^nm)
    MAPE = c(MAPE, abs(y[i+1] - prediction)/abs(y[i+1]))
  }
  return( c( mean(MSPE, na.rm = TRUE),mean(MAPE, na.rm = TRUE) ) )
}
```

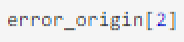
Code 27: Cross Validation for original model error, MSPE, MAPE



```r
error_origin = tsCV_param(MOGE$Deflator, param = c(2,1,2))
```

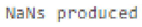


```r
error_origin[1]
```

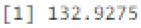


```r
error_origin[2]
```

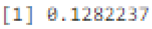
Solution 14: Code 27 results



```
NaNs produced
```

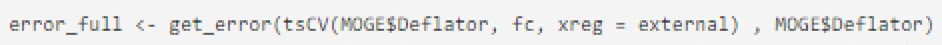


```
[1] 132.9275
```

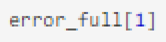


```
[1] 0.1282237
```

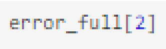
Code 28: Cross Validation for full model error, MSPE, MAPE



```r
error_full <- get_error(tsCV(MOGE$Deflator, fc, xreg = external) , MOGE$Deflator)
```

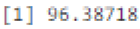


```r
error_full[1]
```

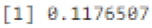


```r
error_full[2]
```

Solution 15: Code 28 results

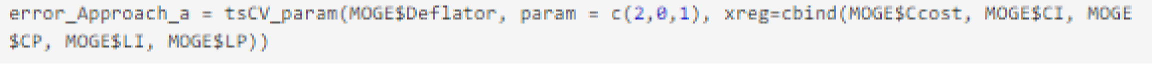


```
[1] 96.38718
```

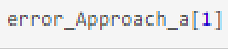


```
[1] 0.1176507
```

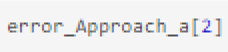
Code 29: Cross Validation for Approach A error, MSPE, MAPE



```r
error_Approach_a = tsCV_param(MOGE$Deflator, param = c(2,0,1), xreg=cbind(MOGE$Ccost, MOGE$CI, MOGE
$CP, MOGE$LI, MOGE$LP))
```

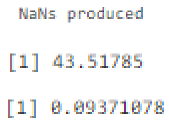


```r
error_Approach_a[1]
```



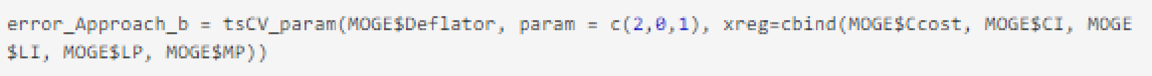
```
error_Approach_a[2]
```

Solution 16: Code 29 results

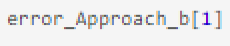


```
NaNs produced

[1] 43.51785

[1] 0.09371078
```

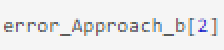
Code 30: Cross Validation for Approach B error, MSPE, MAPE



```
error_Approach_b = tsCV_param(MOGE$Deflator, param = c(2,0,1), xreg=cbind(MOGE$Ccost, MOGE$CI, MOGE
$LI, MOGE$LP, MOGE$MP))
```

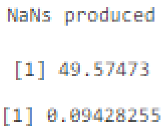


```
error_Approach_b[1]
```



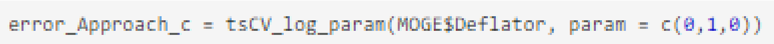
```
error_Approach_b[2]
```

Solution 17: Code 30 results

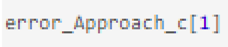


```
NaNs produced

[1] 49.57473

[1] 0.09428255
```

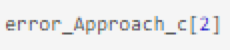
Code 31: Cross Validation for Approach C error, MSPE, MAPE



```
error_Approach_c = tsCV_log_param(MOGE$Deflator, param = c(0,1,0))
```

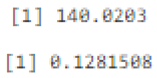


```
error_Approach_c[1]
```



```
error_Approach_c[2]
```

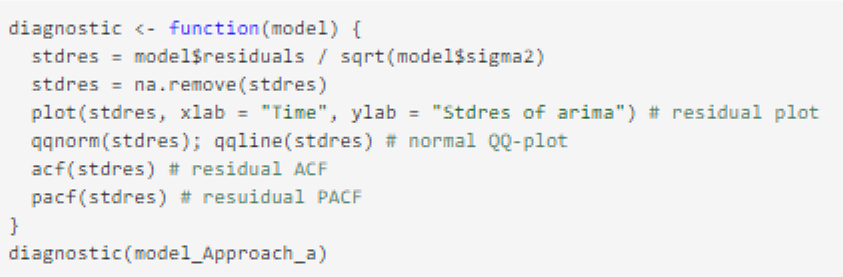
Solution 18: Code 31 results



```
[1] 140.0203

[1] 0.1281508
```

--- MODEL DIAGNOSTIC ---

Code 32: Model diagnostic of Approach A (Figure 8, 9, 10, 11)



```
diagnostic <- function(model) {
  stdres = model$residuals / sqrt(model$sigma2)
  stdres = na.remove(stdres)
  plot(stdres, xlab = "Time", ylab = "Stdres of arima") # residual plot
  qqnorm(stdres); qqline(stdres) # normal QQ-plot
  acf(stdres) # residual ACF
  pacf(stdres) # resuidual PACF
}
diagnostic(model_Approach_a)
```

Figure 8: Line plot of residual from ARIMA with Stdres of arima vs Time

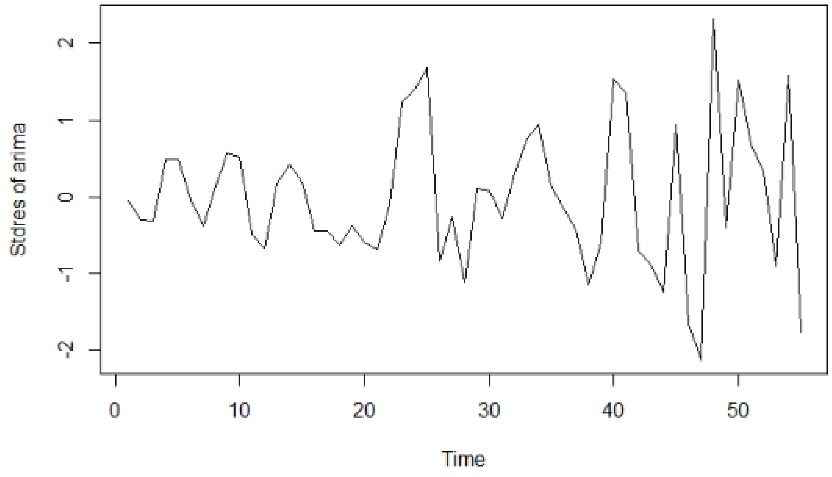


Figure 9: Normal Quantile Plot with Sample Quantiles vs Theoretical Quantiles

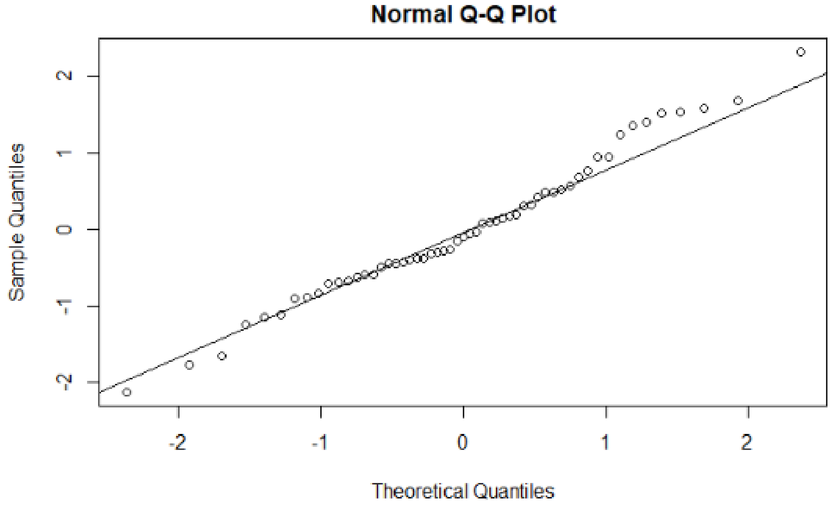


Figure 10: ACF with ACF vs Lag

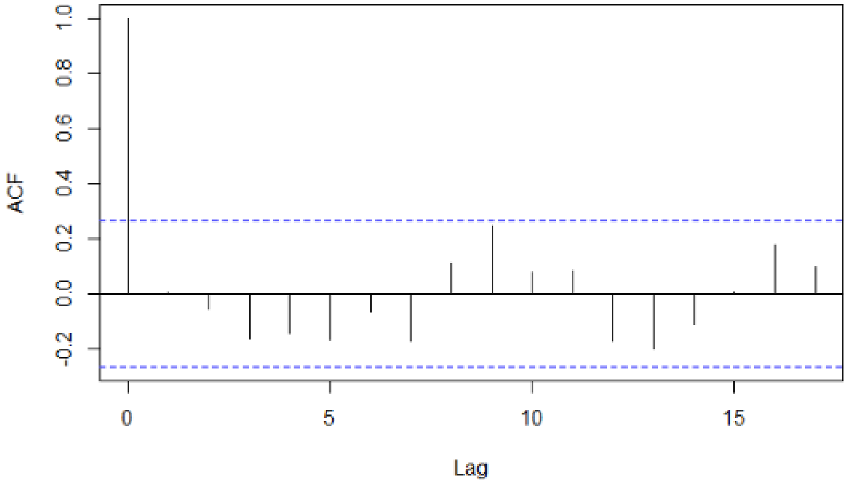
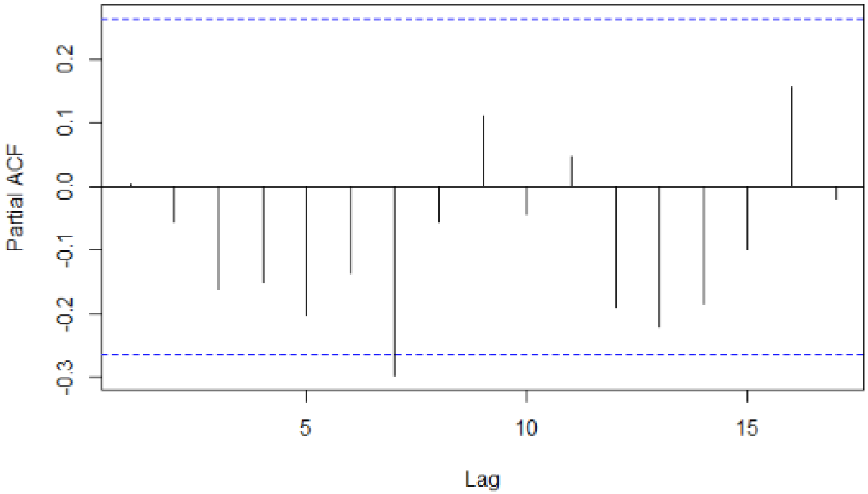


Figure 11: PACF with PACF vs Lag

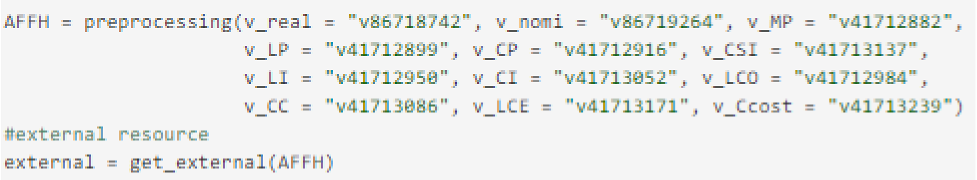


--- AFFH ---

--- Training ---

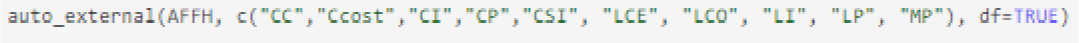
--- Repetition of INITIAL ANALYSIS, MODEL SELECTION, MODEL DIAGNOSTIC ---

Code 33: AFFH Sector Preprocess and find external regressors



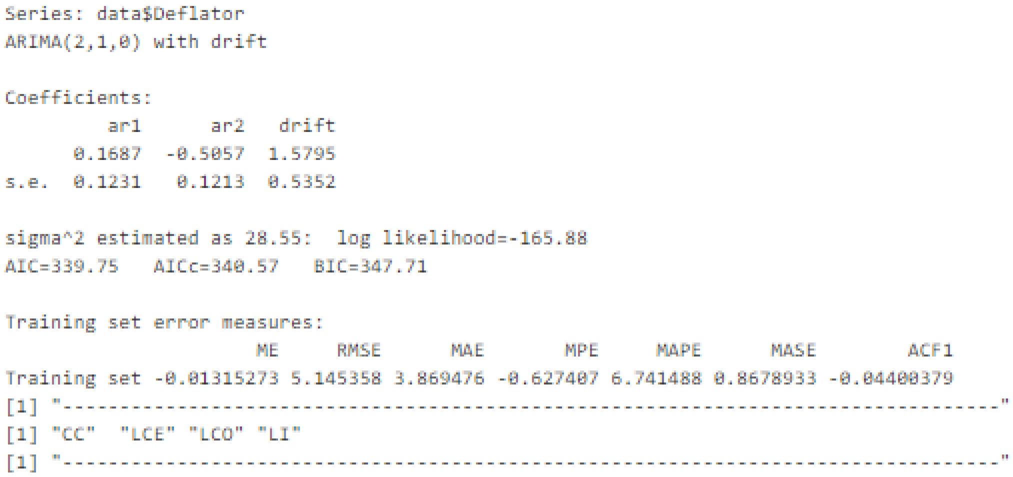
```
AFFH = preprocessing(v_real = "v86718742", v_nomi = "v86719264", v_MP = "v41712882",
                     v_LP = "v41712899", v_CP = "v41712916", v_CSI = "v41713137",
                     v_LI = "v41712950", v_CI = "v41713052", v_LCO = "v41712984",
                     v_CC = "v41713086", v_LCE = "v41713171", v_Ccost = "v41713239")
#external resource
external = get_external(AFFH)
```

Code 34: Auto selection Approach A to return true parameters (Solution 19)



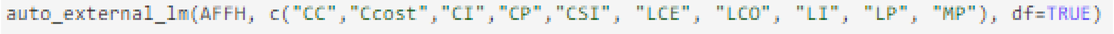
```
auto_external(AFFH, c("CC","Ccost","CI","CP","CSI", "LCE", "LCO", "LI", "LP", "MP"), df=TRUE)
```

Solution 19: Auto selection Approach A to return CC, LCE, LCO, LI



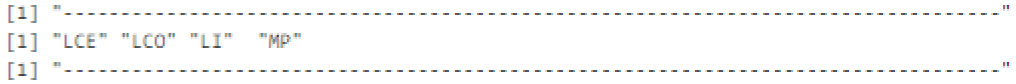
```
Series: data$Deflator
ARIMA(2,1,0) with drift

Coefficients:
         ar1      ar2   drift
      0.1687  -0.5057  1.5795
s.e.  0.1231   0.1213  0.5352

sigma^2 estimated as 28.55:  log likelihood=-165.88
AIC=339.75   AICc=340.57   BIC=347.71

Training set error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set -0.01315273 5.145358 3.869476 -0.627407 6.741488 0.8678933 -0.04400379
[1] "--------------------------------------------------------------------------------"
[1] "CC"  "LCE" "LCO" "LI"
[1] "--------------------------------------------------------------------------------"
```

Code 35: Auto selection Approach B to return true parameters (Solution 20)



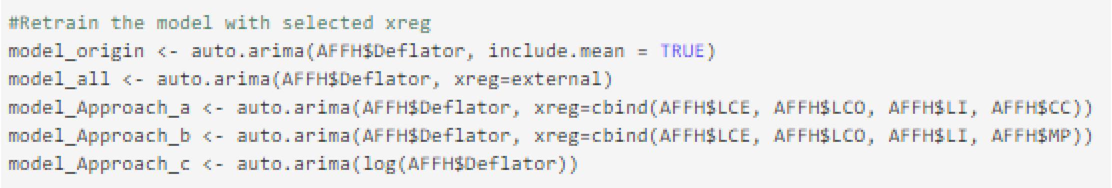
```
auto_external_lm(AFFH, c("CC","Ccost","CI","CP","CSI", "LCE", "LCO", "LI", "LP", "MP"), df=TRUE)
```

Solution 20: Auto selection Approach A to return LCE, LCO, LI, MP



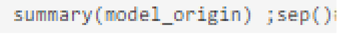
```
[1] "--------------------------------------------------------------------------------"
[1] "LCE" "LCO" "LI"  "MP"
[1] "--------------------------------------------------------------------------------"
```

Code 36: **In Sample Testing** of all model approach bonded with AFFH deflator and true regressors from each approach (Solution 21)



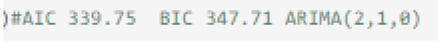
```
#Retrain the model with selected xreg
model_origin <- auto.arima(AFFH$Deflator, include.mean = TRUE)
model_all <- auto.arima(AFFH$Deflator, xreg=external)
model_Approach_a <- auto.arima(AFFH$Deflator, xreg=cbind(AFFH$LCE, AFFH$LCO, AFFH$LI, AFFH$CC))
model_Approach_b <- auto.arima(AFFH$Deflator, xreg=cbind(AFFH$LCE, AFFH$LCO, AFFH$LI, AFFH$MP))
model_Approach_c <- auto.arima(log(AFFH$Deflator))
```

Code 37: Original AFFH Model Summary

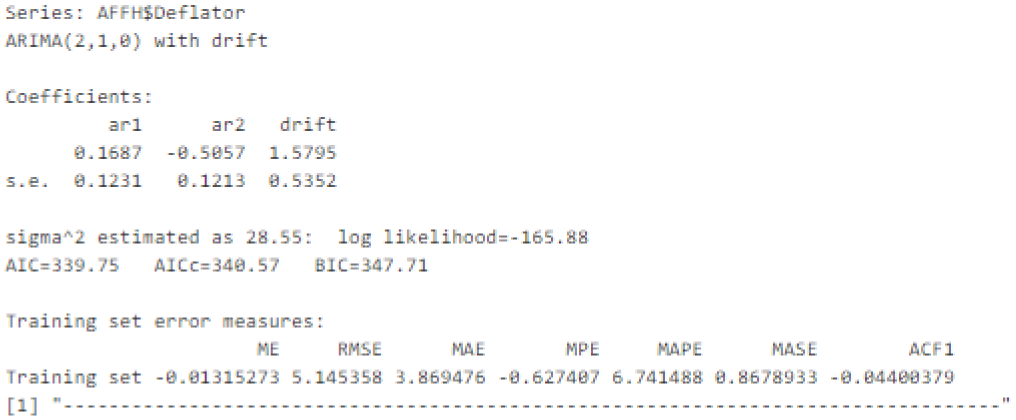


```
summary(model_origin) ;sep()
```

Solution 21: Code 37 results

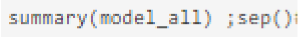


```
)#AIC 339.75  BIC 347.71 ARIMA(2,1,0)
```



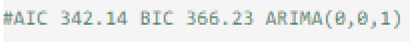
```
Series: AFFH$Deflator
ARIMA(2,1,0) with drift

Coefficients:
         ar1      ar2   drift
      0.1687  -0.5057  1.5795
s.e.  0.1231   0.1213  0.5352

sigma^2 estimated as 28.55:  log likelihood=-165.88
AIC=339.75   AICc=340.57   BIC=347.71

Training set error measures:
                      ME     RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set -0.01315273 5.145358 3.869476 -0.627407 6.741488 0.8678933 -0.04400379
[1] "--------------------------------------------------------------------------------"
```

Code 38: All AFFH Deflator model summary

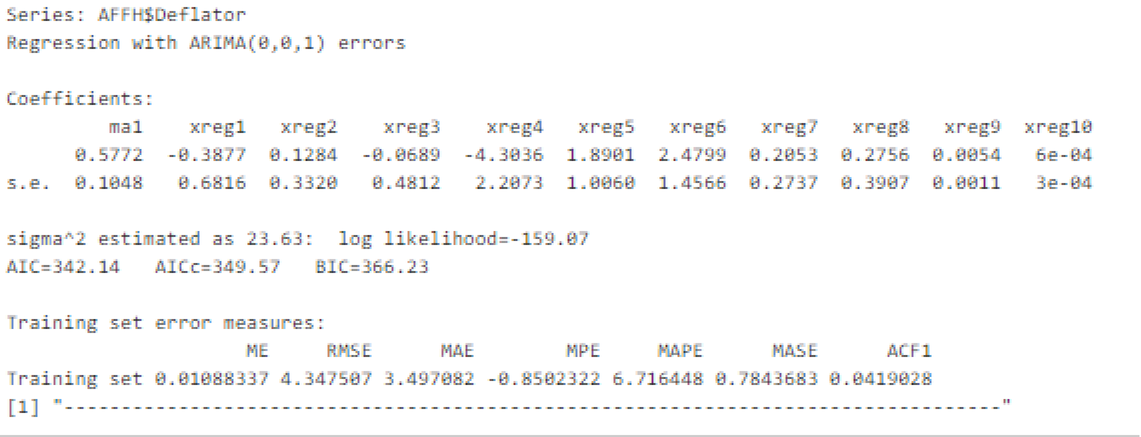


```
summary(model_all) ;sep()
```

Solution 22: Code 38 results



```
#AIC 342.14 BIC 366.23 ARIMA(0,0,1)
```



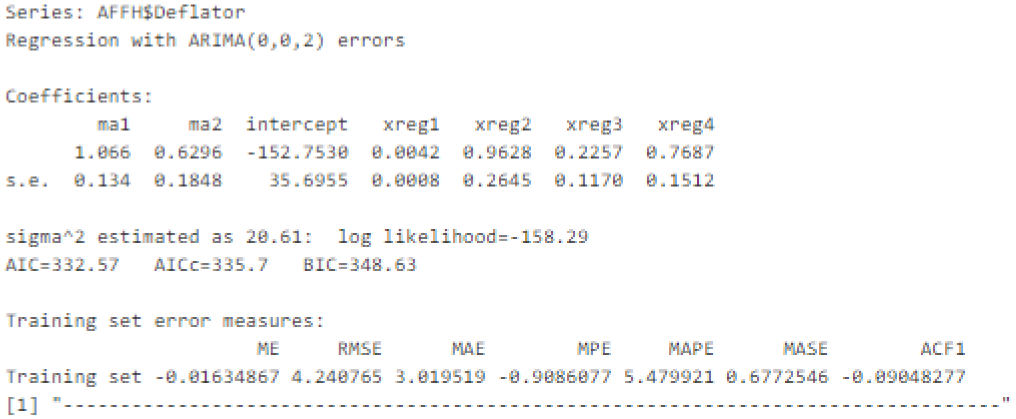
```
Series: AFFH$Deflator
Regression with ARIMA(0,0,1) errors

Coefficients:
         ma1    xreg1   xreg2    xreg3    xreg4   xreg5   xreg6   xreg7   xreg8   xreg9  xreg10
      0.5772  -0.3877  0.1284  -0.0689  -4.3036  1.8901  2.4799  0.2053  0.2756  0.0054   6e-04
s.e.  0.1048   0.6816  0.3320   0.4812   2.2073  1.0060  1.4566  0.2737  0.3907  0.0011   3e-04

sigma^2 estimated as 23.63:  log likelihood=-159.07
AIC=342.14   AICc=349.57   BIC=366.23

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
Training set 0.01088337 4.347507 3.497082 -0.8502322 6.716448 0.7843683 0.0419028
[1] "--------------------------------------------------------------------------------"
```

Code 39: Approach A summary

```
summary(model_Approach_a) ;sep()
```

Solution 23: Code 39 results

```
#arima->lm AIC 332.57 BIC=348.63 ARIMA(0,0,2)
```



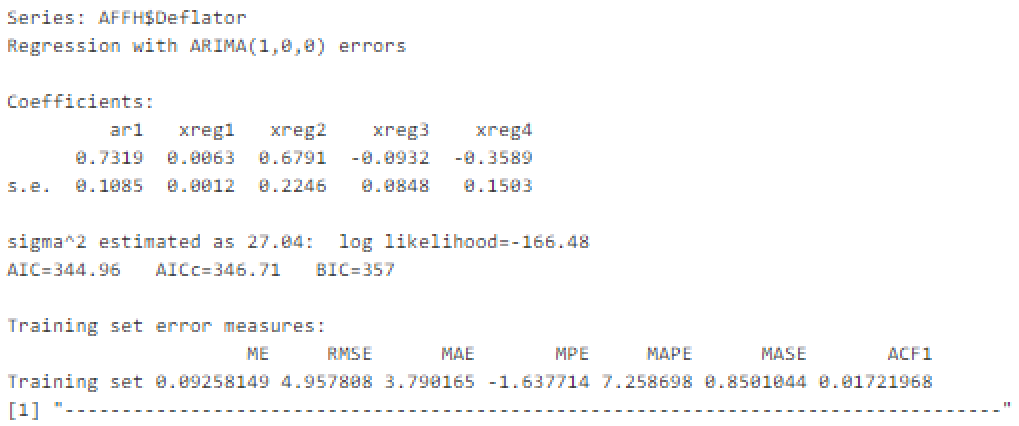
```
Series: AFFH$Deflator
Regression with ARIMA(0,0,2) errors

Coefficients:
        ma1     ma2  intercept   xreg1   xreg2   xreg3   xreg4
      1.066  0.6296  -152.7530  0.0042  0.9628  0.2257  0.7687
s.e.  0.134  0.1848    35.6955  0.0008  0.2645  0.1170  0.1512

sigma^2 estimated as 20.61:  log likelihood=-158.29
AIC=332.57   AICc=335.7   BIC=348.63

Training set error measures:
                      ME     RMSE      MAE        MPE     MAPE      MASE        ACF1
Training set -0.01634867 4.240765 3.019519 -0.9086077 5.479921 0.6772546 -0.09048277
[1] "--------------------------------------------------------------------------------"
```

Code 40: Approach B summary

```
summary(model_Approach_b) ;sep()
```

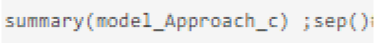
Solution 24: Code 40 results

```
#lm->arima AIC 344.96 BIC=357 ARIMA(1,0,0)
```



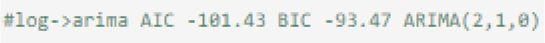
```
Series: AFFH$Deflator
Regression with ARIMA(1,0,0) errors

Coefficients:
         ar1   xreg1   xreg2    xreg3    xreg4
      0.7319  0.0063  0.6791  -0.0932  -0.3589
s.e.  0.1085  0.0012  0.2246   0.0848   0.1503

sigma^2 estimated as 27.04:  log likelihood=-166.48
AIC=344.96   AICc=346.71   BIC=357

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE       ACF1
Training set 0.09258149 4.957808 3.790165 -1.637714 7.258698 0.8501044 0.01721968
[1] "--------------------------------------------------------------------------------"
```

Code 41: Approach C summary

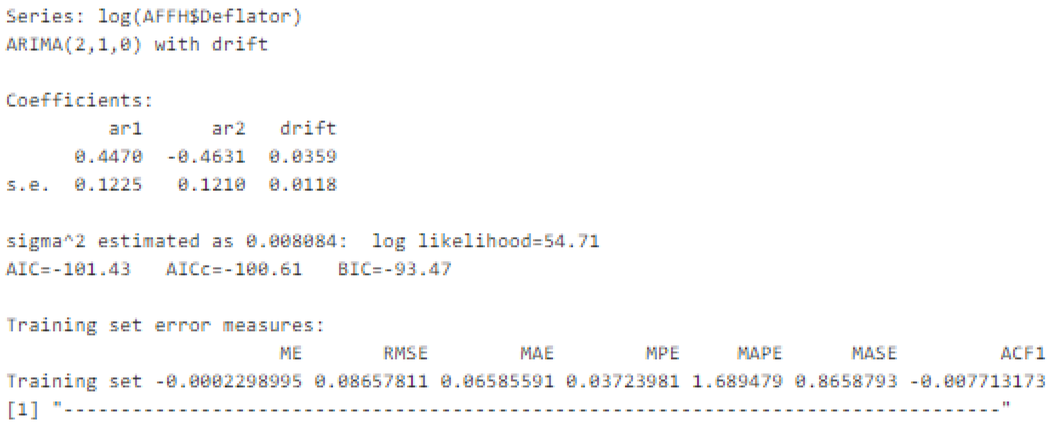


```
summary(model_Approach_c) ;sep()
```

Solution 25: Code 41 results

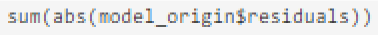


```
#log->arima AIC -101.43 BIC -93.47 ARIMA(2,1,0)
```

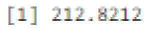


```
Series: log(AFFH$Deflator)
ARIMA(2,1,0) with drift

Coefficients:
         ar1      ar2   drift
      0.4470  -0.4631  0.0359
s.e.  0.1225   0.1210  0.0118

sigma^2 estimated as 0.008084:  log likelihood=54.71
AIC=-101.43   AICc=-100.61   BIC=-93.47

Training set error measures:
                        ME       RMSE        MAE        MPE     MAPE      MASE         ACF1
Training set -0.0002298995 0.08657811 0.06585591 0.03723981 1.689479 0.8658793 -0.007713173
[1] "--------------------------------------------------------------------------------"
```

Code 42 & Solution 26: Sum of absolute values of all residuals from original MOGE model set

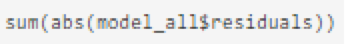


```
sum(abs(model_origin$residuals))
```

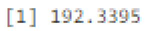


```
[1] 212.8212
```

Code 43 & Solution 27: Sum of absolute values of all residuals from all MOGE model set

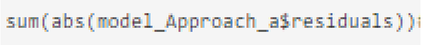


```
sum(abs(model_all$residuals))
```

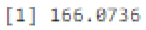


```
[1] 192.3395
```

Code 44 & Solution 28: Sum of absolute values of all residuals from Approach A model

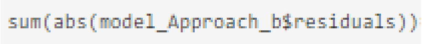


```
sum(abs(model_Approach_a$residuals))
```

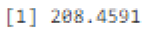


```
[1] 166.0736
```

Code 45 & Solution 29: Sum of absolute values of all residuals from Approach B model

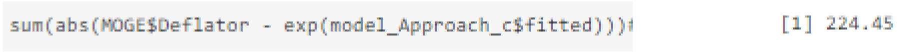


```
sum(abs(model_Approach_b$residuals))
```



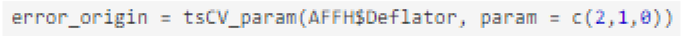
```
[1] 208.4591
```

Code 46 & Solution 30: Sum of absolute values, differenced between MOGE Deflator and fitted Approach C

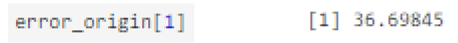


```
sum(abs(MOGE$Deflator - exp(model_Approach_c$fitted))) [1] 224.45
```

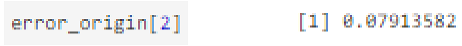
Code 47 & Solution 31: Cross Validation for original model error, MSPE, MAPE



```
error_origin = tsCV_param(AFFH$Deflator, param = c(2,1,0))
```

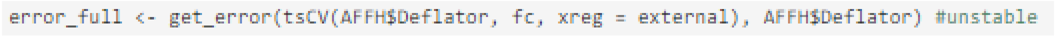


```
error_origin[1]          [1] 36.69845
```

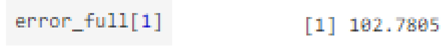


```
error_origin[2]          [1] 0.07913582
```

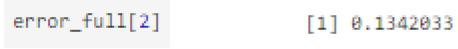
Code 48 & Solution 32: Cross Validation for full model error, MSPE, MAPE



```
error_full <- get_error(tsCV(AFFH$Deflator, fc, xreg = external), AFFH$Deflator) #unstable
```

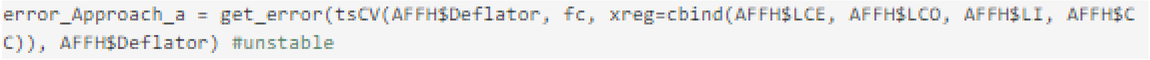


```
error_full[1]          [1] 102.7805
```

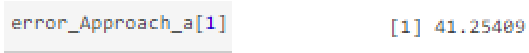


```
error_full[2]          [1] 0.1342033
```

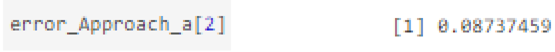
Code 49 & Solution 33: Cross Validation for Approach A error, MSPE, MAPE



```
error_Approach_a = get_error(tsCV(AFFH$Deflator, fc, xreg=cbind(AFFH$LCE, AFFH$LCO, AFFH$LI, AFFH$C
C)), AFFH$Deflator) #unstable
```

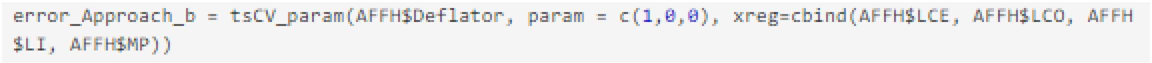


```
error_Approach_a[1]          [1] 41.25409
```

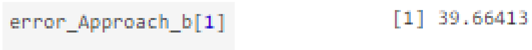


```
error_Approach_a[2]          [1] 0.08737459
```

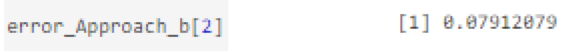
Code 50 & Solution 34: Cross Validation for Approach B error, MSPE, MAPE



```
error_Approach_b = tsCV_param(AFFH$Deflator, param = c(1,0,0), xreg=cbind(AFFH$LCE, AFFH$LCO, AFFH
$LI, AFFH$MP))
```

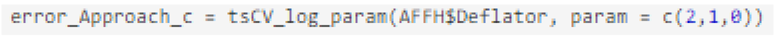


```
error_Approach_b[1]          [1] 39.66413
```

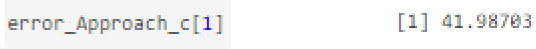


```
error_Approach_b[2]          [1] 0.07912079
```

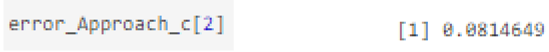
Code 51 & Solution 35: Cross Validation for Approach C error, MSPE, MAPE



```
error_Approach_c = tsCV_log_param(AFFH$Deflator, param = c(2,1,0))
```



```
error_Approach_c[1]          [1] 41.98703
```



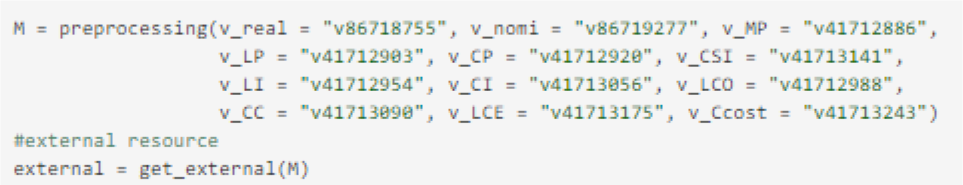
```
error_Approach_c[2]          [1] 0.0814649
```

-- M --

--- Training ---

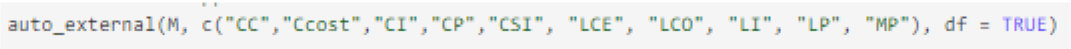
--- Repetition of INITIAL ANALYSIS, MODEL SELECTION, MODEL DIAGNOSTIC ---

Code 52: M Sector Preprocess and find external regressors



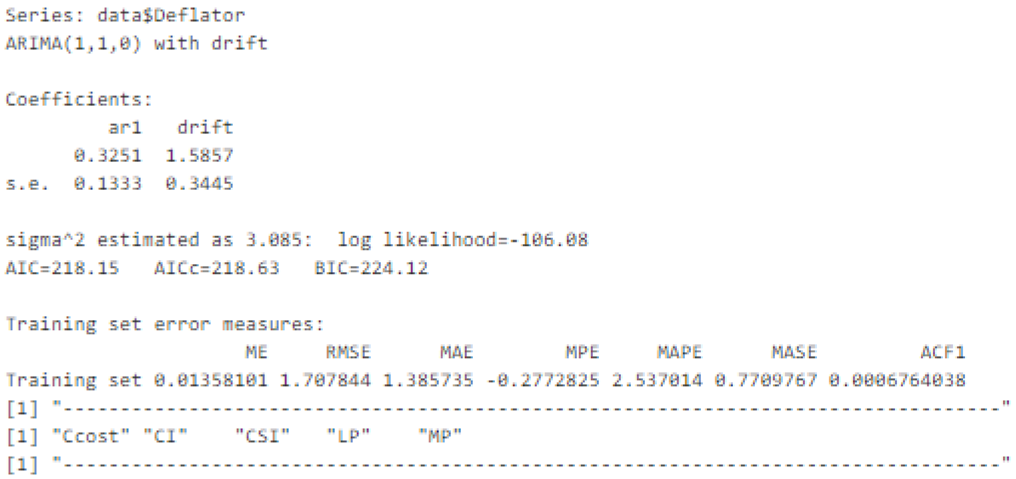
```
M = preprocessing(v_real = "v86718755", v_nomi = "v86719277", v_MP = "v41712886",
                  v_LP = "v41712903", v_CP = "v41712920", v_CSI = "v41713141",
                  v_LI = "v41712954", v_CI = "v41713056", v_LCO = "v41712988",
                  v_CC = "v41713090", v_LCE = "v41713175", v_Ccost = "v41713243")
#external resource
external = get_external(M)
```

Code 53: Auto selection Approach A to return true parameters (Solution 36)



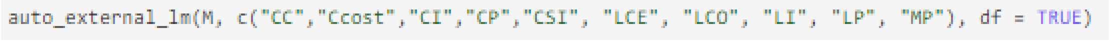
```
auto_external(M, c("CC","Ccost","CI","CP","CSI", "LCE", "LCO", "LI", "LP", "MP"), df = TRUE)
```

Solution 36: Auto selection Approach A to return Ccost, CI, CSI, LP, MP



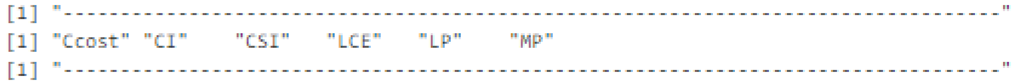
```
Series: data$Deflator
ARIMA(1,1,0) with drift

Coefficients:
         ar1   drift
      0.3251  1.5857
s.e.  0.1333  0.3445

sigma^2 estimated as 3.085:  log likelihood=-106.08
AIC=218.15   AICc=218.63   BIC=224.12

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
Training set 0.01358101 1.707844 1.385735 -0.2772825 2.537014 0.7709767 0.0006764038
[1] "--------------------------------------------------------------------------------"
[1] "Ccost" "CI"    "CSI"   "LP"    "MP"
[1] "--------------------------------------------------------------------------------"
```

Code 54: Auto selection Approach B to return true parameters (Solution 37)



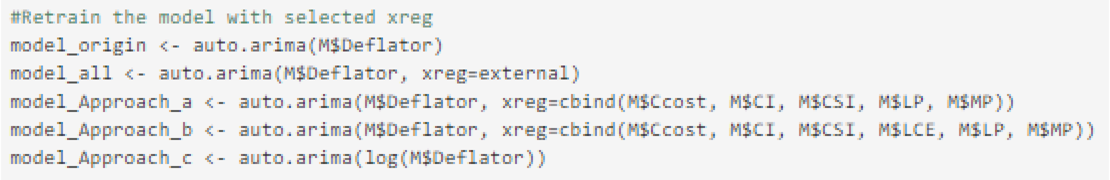
```
auto_external_lm(M, c("CC","Ccost","CI","CP","CSI", "LCE", "LCO", "LI", "LP", "MP"), df = TRUE)
```

Solution 37: Auto selection Approach A to return CI, CSI, LCE, LP, MP



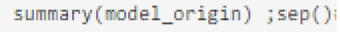
```
[1] "--------------------------------------------------------------------------------"
[1] "Ccost" "CI"    "CSI"   "LCE"   "LP"    "MP"
[1] "--------------------------------------------------------------------------------"
```

Code 55: **In Sample Testing** of all model approach bonded with M deflator and true regressors from each approach (Solution 21)



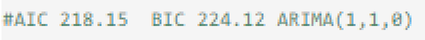
```
#Retrain the model with selected xreg
model_origin <- auto.arima(M$Deflator)
model_all <- auto.arima(M$Deflator, xreg=external)
model_Approach_a <- auto.arima(M$Deflator, xreg=cbind(M$Ccost, M$CI, M$CSI, M$LP, M$MP))
model_Approach_b <- auto.arima(M$Deflator, xreg=cbind(M$Ccost, M$CI, M$CSI, M$LCE, M$LP, M$MP))
model_Approach_c <- auto.arima(log(M$Deflator))
```

Code 56: Original M Model Summary

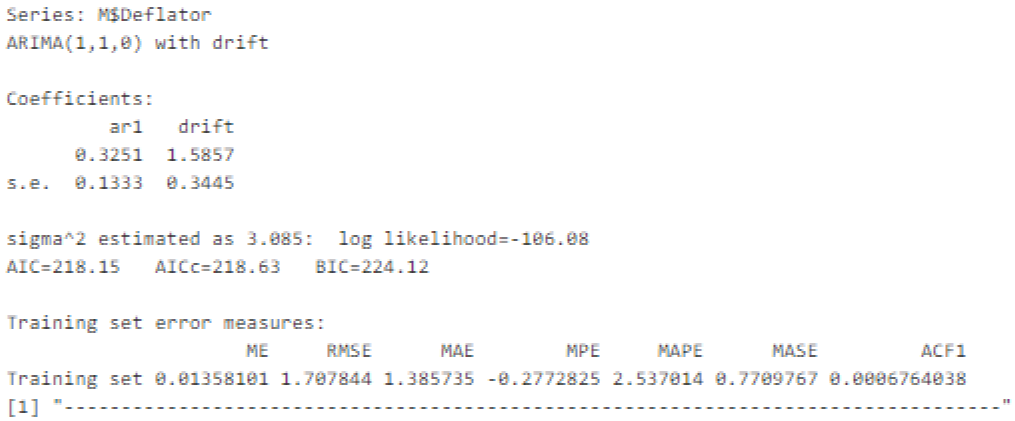


```
summary(model_origin) ;sep()
```

Solution 38: Code 56 results

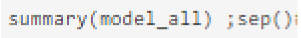


```
#AIC 218.15  BIC 224.12 ARIMA(1,1,0)
```



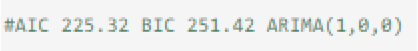
```
Series: M$Deflator
ARIMA(1,1,0) with drift

Coefficients:
         ar1   drift
      0.3251  1.5857
s.e.  0.1333  0.3445

sigma^2 estimated as 3.085:  log likelihood=-106.08
AIC=218.15   AICc=218.63   BIC=224.12

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
Training set 0.01358101 1.707844 1.385735 -0.2772825 2.537014 0.7709767 0.0006764038
[1] "-----------------------------------------------------------------------------------"
```

Code 57: All M Deflator model summary

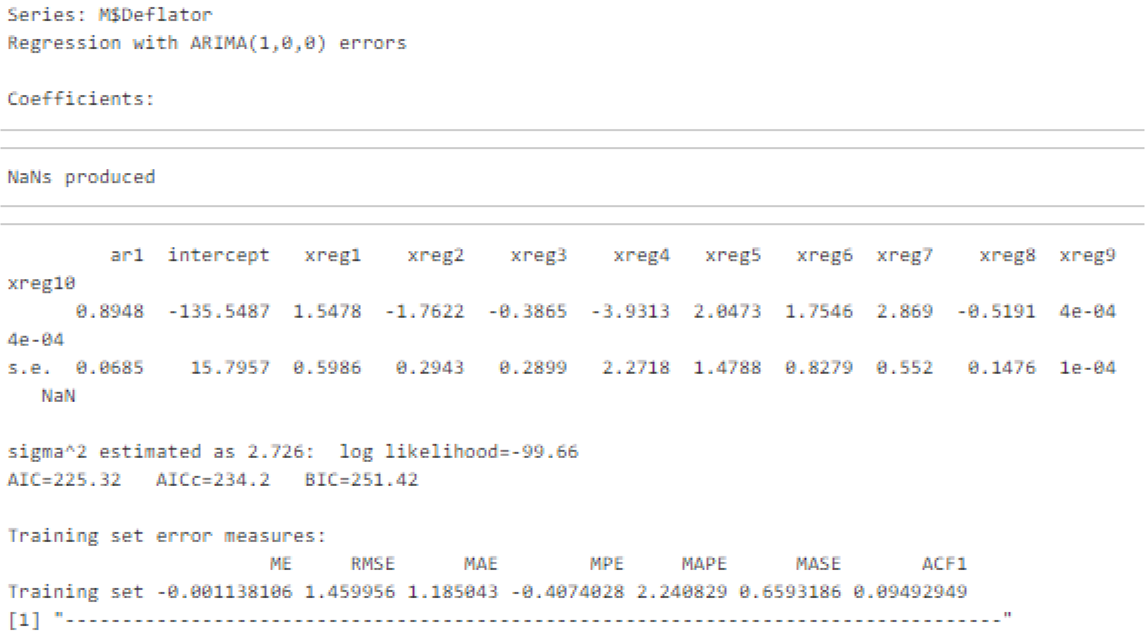


```
summary(model_all) ;sep();
```

Solution 39: Code 57 results

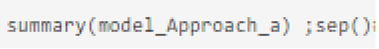


```
#AIC 225.32 BIC 251.42 ARIMA(1,0,0)
```



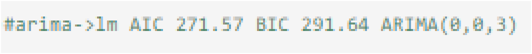
```
Series: M$Deflator
Regression with ARIMA(1,0,0) errors

Coefficients:

NaNs produced

         ar1  intercept   xreg1    xreg2    xreg3    xreg4   xreg5   xreg6  xreg7    xreg8  xreg9
xreg10
      0.8948  -135.5487  1.5478  -1.7622  -0.3865  -3.9313  2.0473  1.7546  2.869  -0.5191  4e-04
4e-04
s.e.  0.0685    15.7957  0.5986   0.2943   0.2899   2.2718  1.4788  0.8279  0.552   0.1476  1e-04
   NaN

sigma^2 estimated as 2.726:  log likelihood=-99.66
AIC=225.32   AICc=234.2   BIC=251.42

Training set error measures:
                       ME     RMSE      MAE        MPE     MAPE      MASE       ACF1
Training set -0.001138106 1.459956 1.185043 -0.4074028 2.240829 0.6593186 0.09492949
[1] "-----------------------------------------------------------------------------------"
```

Code 58: Approach A summary

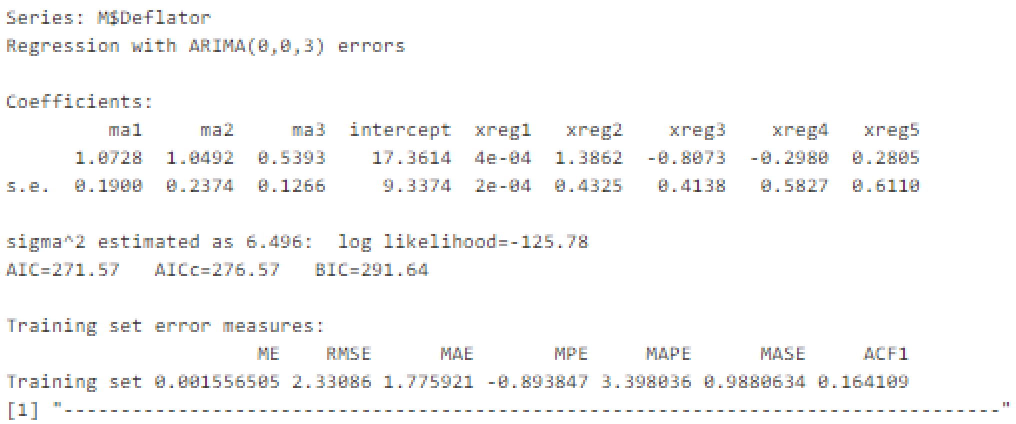


```
summary(model_Approach_a) ;sep();
```

Solution 40: Code 58 results

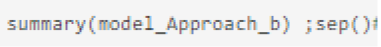


```
#arima->lm AIC 271.57 BIC 291.64 ARIMA(0,0,3)
```



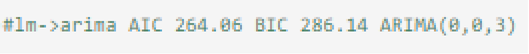
```
Series: M$Deflator
Regression with ARIMA(0,0,3) errors

Coefficients:
         ma1     ma2     ma3  intercept  xreg1   xreg2    xreg3    xreg4   xreg5
      1.0728  1.0492  0.5393    17.3614  4e-04  1.3862  -0.8073  -0.2980  0.2805
s.e.  0.1900  0.2374  0.1266     9.3374  2e-04  0.4325   0.4138   0.5827  0.6110

sigma^2 estimated as 6.496:  log likelihood=-125.78
AIC=271.57   AICc=276.57   BIC=291.64

Training set error measures:
                      ME    RMSE      MAE       MPE     MAPE      MASE     ACF1
Training set 0.001556505 2.33086 1.775921 -0.893847 3.398036 0.9880634 0.164109
[1] "----------------------------------------------------------------------------------"
```

Code 59: Approach B summary

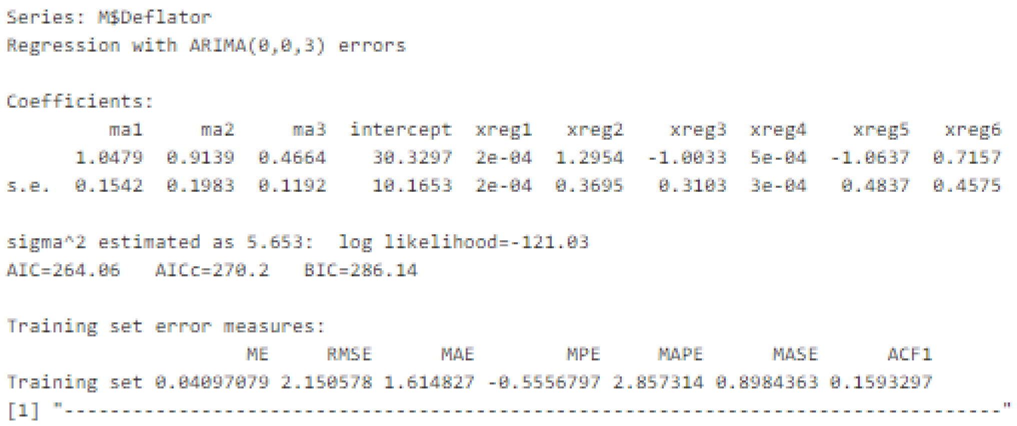


```
summary(model_Approach_b) ;sep()
```

Solution 41: Code 59 results

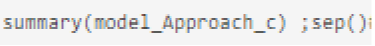


```
#lm->arima AIC 264.06 BIC 286.14 ARIMA(0,0,3)
```



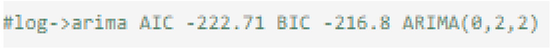
```
Series: M$Deflator
Regression with ARIMA(0,0,3) errors

Coefficients:
         ma1     ma2     ma3  intercept  xreg1   xreg2    xreg3  xreg4    xreg5   xreg6
      1.0479  0.9139  0.4664    30.3297  2e-04  1.2954  -1.0033  5e-04  -1.0637  0.7157
s.e.  0.1542  0.1983  0.1192    10.1653  2e-04  0.3695   0.3103  3e-04   0.4837  0.4575

sigma^2 estimated as 5.653:  log likelihood=-121.03
AIC=264.06   AICc=270.2   BIC=286.14

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE      ACF1
Training set 0.04097079 2.150578 1.614827 -0.5556797 2.857314 0.8984363 0.1593297
[1] "----------------------------------------------------------------------------------"
```

Code 60: Approach C summary

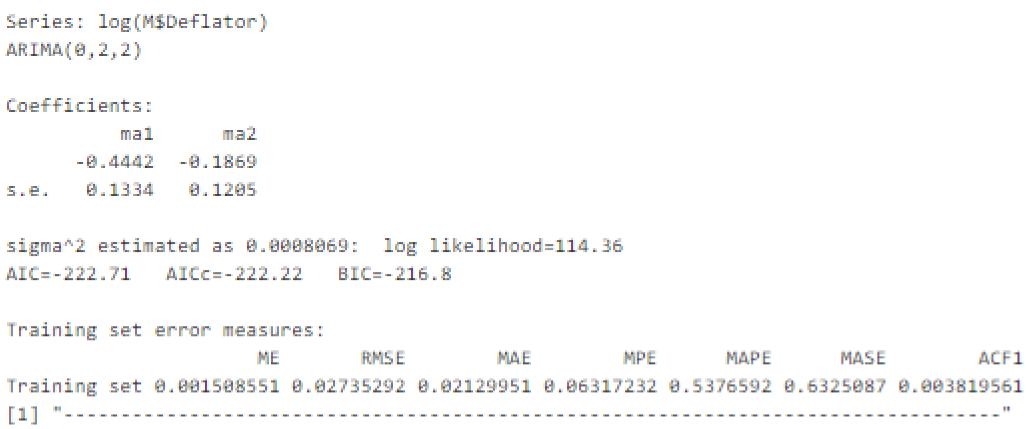


```
summary(model_Approach_c) ;sep()
```

Solution 42: Code 60 results

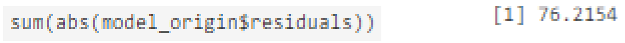


```
#log->arima AIC -222.71 BIC -216.8 ARIMA(0,2,2)
```



```
Series: log(M$Deflator)
ARIMA(0,2,2)

Coefficients:
          ma1      ma2
      -0.4442  -0.1869
s.e.   0.1334   0.1205

sigma^2 estimated as 0.0008069:  log likelihood=114.36
AIC=-222.71   AICc=-222.22   BIC=-216.8

Training set error measures:
                      ME       RMSE        MAE        MPE      MAPE      MASE        ACF1
Training set 0.001508551 0.02735292 0.02129951 0.06317232 0.5376592 0.6325087 0.003819561
[1] "----------------------------------------------------------------------------------"
```

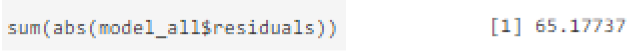
Code 61 & Solution 43: Sum of absolute values of all residuals from original M model set



```
sum(abs(model_origin$residuals))
```

```
[1] 76.2154
```

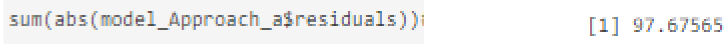
Code 62 & Solution 44: Sum of absolute values of all residuals from all M model set



```
sum(abs(model_all$residuals))
```

```
[1] 65.17737
```

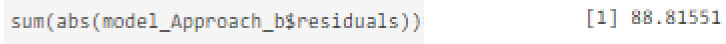
Code 63 & Solution 45: Sum of absolute values of all residuals from Approach A model



```
sum(abs(model_Approach_a$residuals))
```

```
[1] 97.67565
```

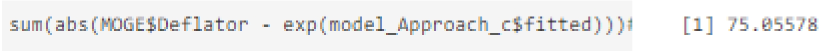
Code 64 & Solution 46: Sum of absolute values of all residuals from Approach B model



```
sum(abs(model_Approach_b$residuals))
```

```
[1] 88.81551
```

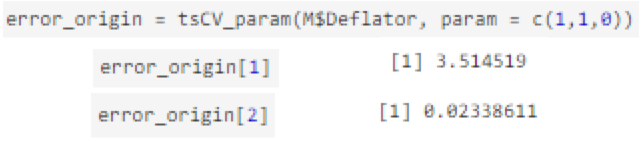
Code 65 & Solution 47: Sum of absolute values, differenced between M Deflator and fitted Approach C



```
sum(abs(MOGE$Deflator - exp(model_Approach_c$fitted)))
```

```
[1] 75.05578
```

Code 66 & Solution 48: Cross Validation for original model error, MSPE, MAPE



```
error_origin = tsCV_param(M$Deflator, param = c(1,1,0))
```

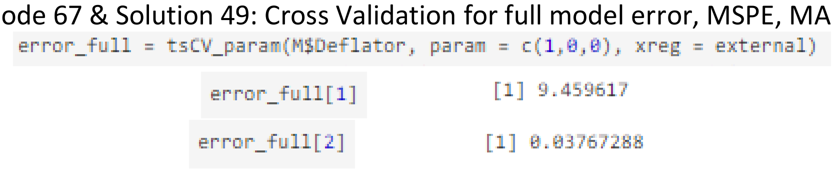
```
error_origin[1]
```

```
[1] 3.514519
```

```
error_origin[2]
```

```
[1] 0.02338611
```

Code 67 & Solution 49: Cross Validation for full model error, MSPE, MAPE



```
error_full = tsCV_param(M$Deflator, param = c(1,0,0), xreg = external)
```

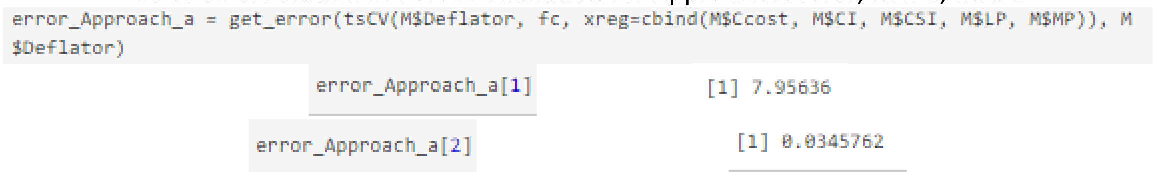
```
error_full[1]
```

```
[1] 9.459617
```

```
error_full[2]
```

```
[1] 0.03767288
```

Code 68 & Solution 50: Cross Validation for Approach A error, MSPE, MAPE



```
error_Approach_a = get_error(tsCV(M$Deflator, fc, xreg=cbind(M$Ccost, M$CI, M$CSI, M$LP, M$MP)), M
$Deflator)
```

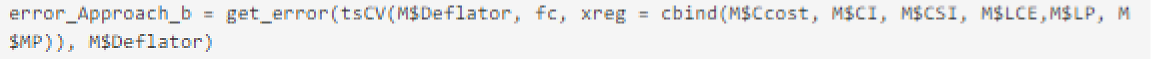
```
error_Approach_a[1]
```

```
[1] 7.95636
```

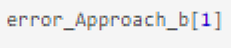
```
error_Approach_a[2]
```

```
[1] 0.0345762
```

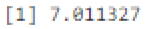
Code 69 & Solution 51: Cross Validation for Approach B error, MSPE, MAPE



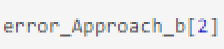
```
error_Approach_b = get_error(tsCV(M$Deflator, fc, xreg = cbind(M$Ccost, M$CI, M$CSI, M$LCE,M$LP, M
$MP)), M$Deflator)
```



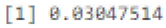
```
error_Approach_b[1]
```



```
[1] 7.011327
```

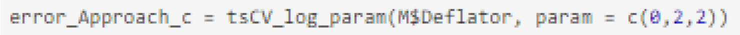


```
error_Approach_b[2]
```

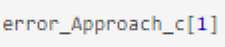


```
[1] 0.03047514
```

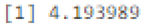
Code 70 & Solution 52: Cross Validation for Approach C error, MSPE, MAPE



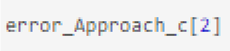
```
error_Approach_c = tsCV_log_param(M$Deflator, param = c(0,2,2))
```



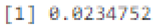
```
error_Approach_c[1]
```



```
[1] 4.193989
```



```
error_Approach_c[2]
```



```
[1] 0.0234752
```