Selecting the Optimal Time Series Model using MAE

Timothy Lee

25/07/2020

Intro

The data loaded is a bivariate time series object consisting of two Canadian monthly macroeconomic series.

Variable `emp` contains the raw (unadjusted) number of employed individuals (in 1,000's).
Variable `gdp` contains the *seasonally adjusted* real GDP, chained to 2012 $'s (in 1,000,000's).

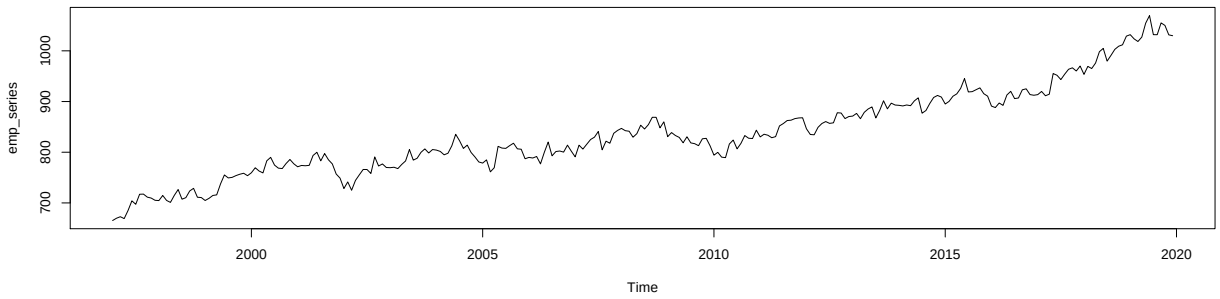
The goal of this mini project is to determine the best time series model to forecast GDP values using the MAE of the forecasts versus the actual GDP values from the latest year.

Exploratory Analysis

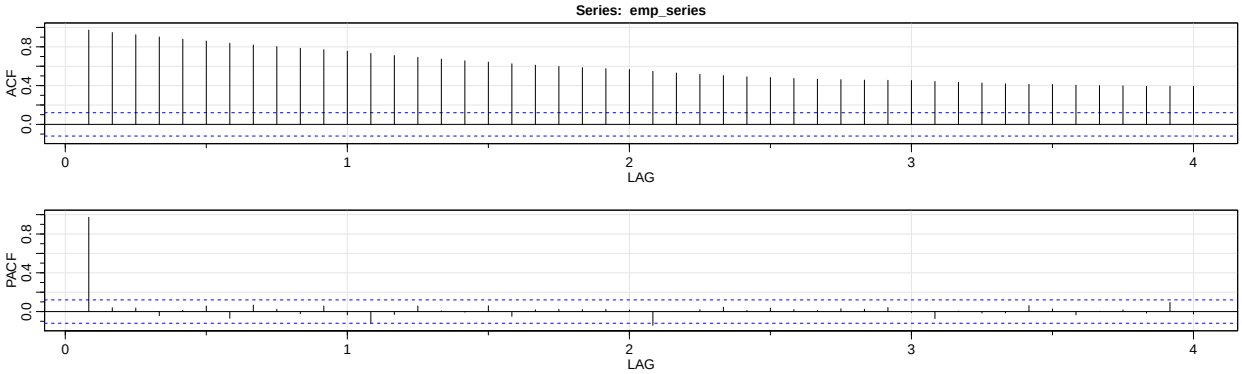
```
#loading neccessary libraries and data
library(astsa)
library(forecast)
library(vars)
library(fGarch)
library(tseries)
load("emp_gdp_data.Rdata") #data loaded as X object

emp_series = as.ts(X[,1], frequency=12)
gdp_series = as.ts(X[,2], frequency=12)

#EMP
plot(emp_series)
```

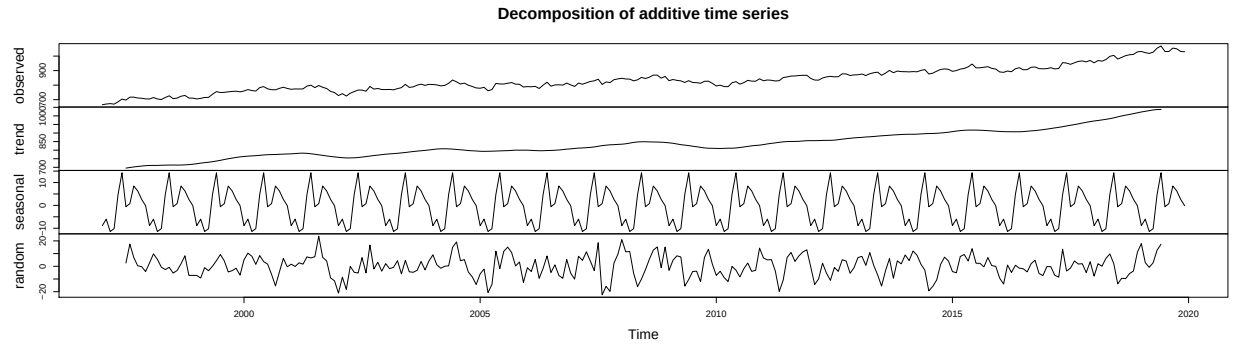



```
acf2(emp_series)
```




```
##      [,1] [,2] [,3]  [,4] [,5] [,6]  [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.97 0.95 0.93  0.90 0.88 0.86  0.84 0.82 0.80  0.78  0.77  0.76  0.73
## PACF 0.97 0.04 0.04 -0.04 0.01 0.06 -0.07 0.07 0.02 -0.02  0.06 -0.03 -0.12
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.71  0.69  0.67  0.66  0.64  0.63  0.61  0.60  0.59  0.57  0.57  0.55
## PACF -0.03  0.06  0.01 -0.01  0.06 -0.05  0.02  0.02  0.02  0.02  0.02 -0.14
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.53  0.52  0.50  0.49  0.48  0.47  0.47  0.46  0.46  0.46  0.45  0.44
## PACF  0.00  0.02  0.04  0.01  0.04  0.02  0.01  0.03  0.02  0.04 -0.01 -0.07
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.43  0.43  0.42  0.41  0.41  0.40   0.4  0.40  0.39  0.39  0.39
## PACF  0.01 -0.01 -0.01  0.06  0.03 -0.03   0.0  0.02 -0.02  0.09 -0.02
```

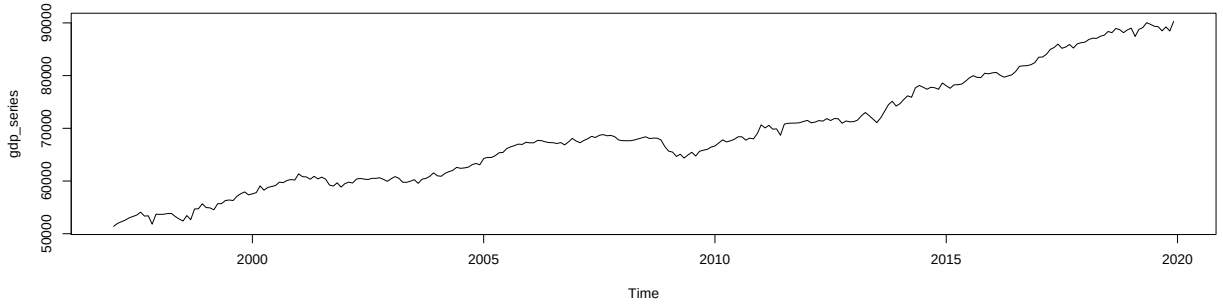
```
plot(decompose(emp_series))
```



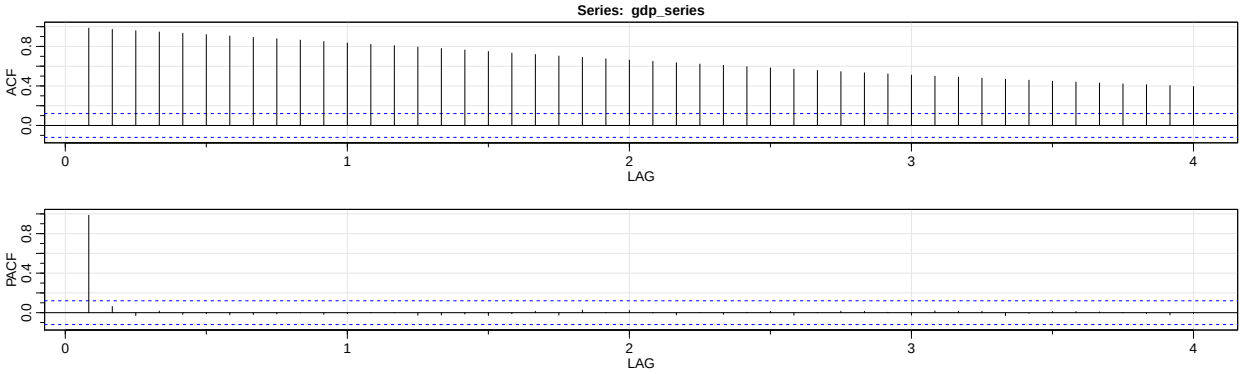
For the `emp` series, we can see from the original graph, there is a clear increasing upward trend, and based on the ACF plot with very high and non-decreasing ACF values, it seems like it is a random walk process. There is also strong evidence of seasonality from the original series where there are some cyclic "spikes", which is also reflected in the decomposed plot (period of 12 months). Hence, due to the random walk behaviour, we can conclude that the original `emp` series itself is non-stationary and integrated.

```
gdp_series = as.ts(X[,2], frequency=12)

#GDP
plot(gdp_series)
```

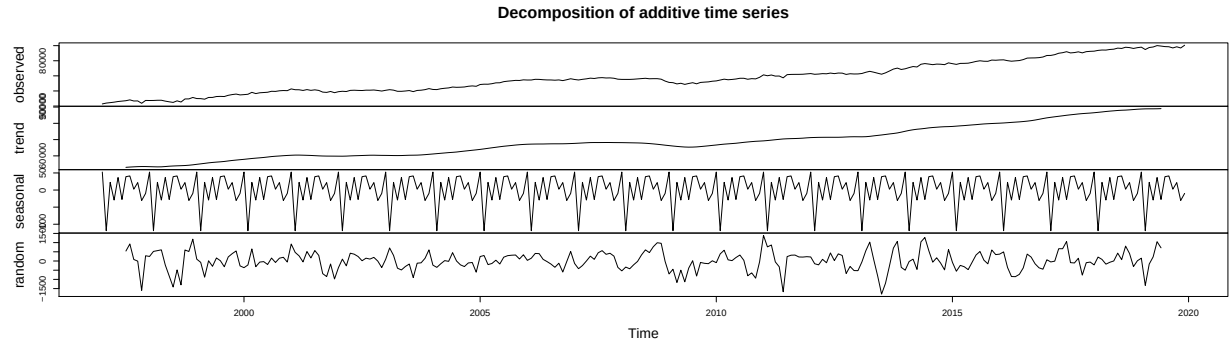



```
acf2(gdp_series)
```




```
##       [,1] [,2]  [,3] [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11] [,12] [,13]
## ACF   0.99 0.97  0.96 0.95  0.93  0.92  0.91  0.89  0.88  0.86  0.85  0.84  0.82
## PACF  0.99 0.06 -0.03 0.02 -0.02 -0.01 -0.02 -0.02 -0.01  0.00 -0.02 -0.01  0.00
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF    0.81  0.79  0.78  0.76  0.75  0.73  0.72  0.70  0.69  0.67  0.66  0.65
## PACF   0.00 -0.03 -0.02 -0.01 -0.03 -0.01  0.01 -0.02  0.03  0.00  0.02  0.00
##       [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF    0.63  0.62  0.61  0.59  0.58  0.57  0.56  0.54  0.53  0.52  0.51  0.50
## PACF  -0.02  0.00 -0.02  0.00  0.01 -0.03  0.00  0.02  0.01  0.00 -0.02  0.02
##       [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF    0.49  0.48  0.47  0.46  0.45  0.44  0.43  0.42  0.41  0.40  0.39
## PACF   0.01  0.01 -0.03 -0.01  0.01  0.01  0.01  0.00 -0.01 -0.02 -0.01
```

```
plot(decompose(gdp_series))
```



For the GDP series, we can see from the original graph, there is also a clear increasing upward trend, and

based on the ACF plot it seems like it is also a random walk process (with very high correlation across all lags). There is also less evidence of seasonality, since this series is already *seasonally adjusted*. Hence, again, due to the random walk behaviour, we can easily conclude that the original `GDP` series itself is non-stationary and integrated.

This integrated behaviour of both series is also reinforced by the `adf` test as follows (where we failed to reject the null-hypthosis of both series being non-stationary):

```
tseries::adf.test( emp_series, k= 12)
```

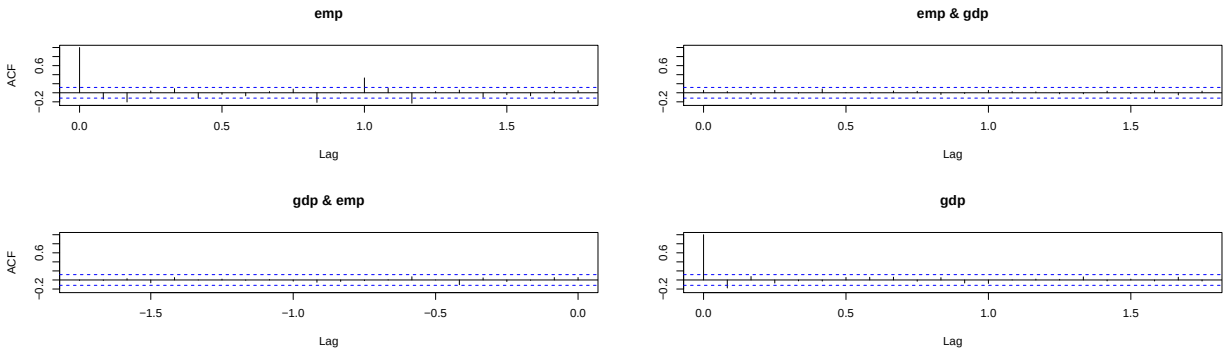
```
##
##  Augmented Dickey-Fuller Test
##
## data:  emp_series
## Dickey-Fuller = -1.5356, Lag order = 12, p-value = 0.7717
## alternative hypothesis: stationary
```

```
tseries::adf.test( gdp_series, k= 12)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gdp_series
## Dickey-Fuller = -1.274, Lag order = 12, p-value = 0.8819
## alternative hypothesis: stationary
```

Hence, I have performed a first order difference on both series, resulting in a stationary-like cross correlation plot as follows. This suggests that both series are integrated with order of `1`.

```
acf(diff(X))
```

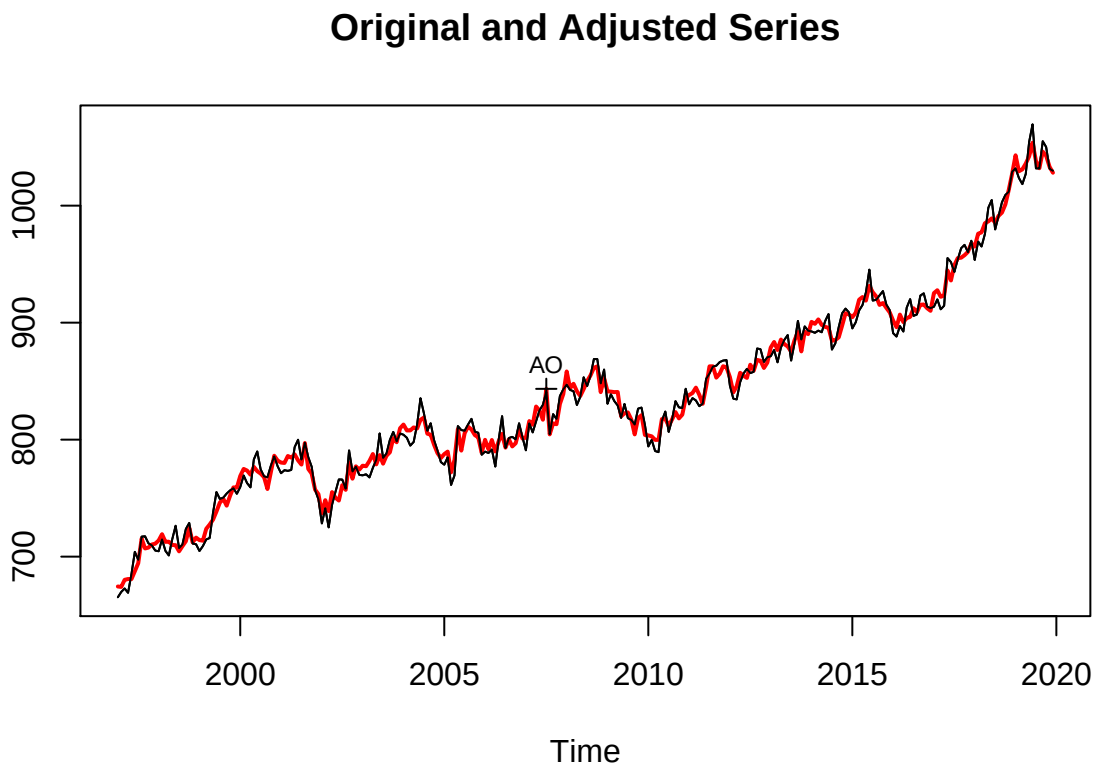


Addressing seasonality

Now, I will attempt to address the *seasonality* component of the employment (`emp`) series using the X11 decomposition method, then plotting the original and seasonally adjusted series on the same plot.

```
dcmp_X11 = seasonal::seas( emp_series, x11 = "") #new emp series

plot((dcmp_X11))
lines(emp_series) #overlay original emp series again
```



Fitting ARIMA model on GDP

First, I will fit an `ARIMA` model to the `GDP` series, selecting the model specification by `AIC` (default).

```
arima_out = forecast::auto.arima( gdp_series, ic = "aic")
summary(arima_out)
```

```
## Series: gdp_series
## ARIMA(0,1,1)(2,0,0)[12] with drift
##
## Coefficients:
##           ma1     sar1     sar2     drift
##       -0.1697  -0.1264  -0.1851  139.8089
## s.e.   0.0574   0.0646   0.0660   21.5741
##
## sigma^2 estimated as 312764:  log likelihood=-2128.51
## AIC=4267.01   AICc=4267.24   BIC=4285.1
##
## Training set error measures:
##                      ME     RMSE      MAE         MPE      MAPE      MASE
## Training set 0.7658949 554.1641 419.7738 -0.01142376 0.6202567 0.2068411
##                      ACF1
## Training set -0.01156402
```

```
MAE_arima = mean( abs( gdp_series - fitted(arima_out) ), na.rm = T )
MAE_arima
```

```
## [1] 419.7738
```

The MAE we get for this ARIMA model purely using the `GDP` series itself is 419.7737635.

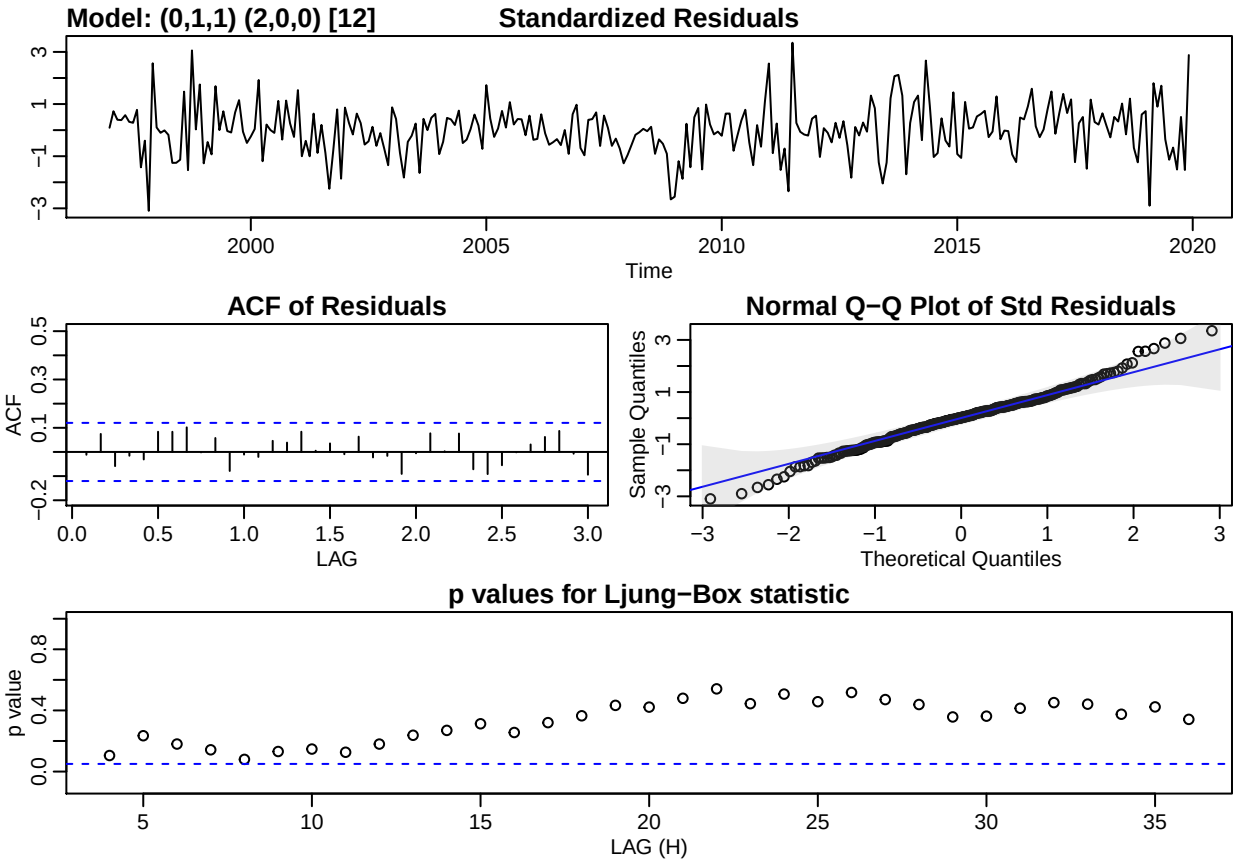
The fitted model is $SARIMA(0,1,1)(2,0,0)_{12}$ or could be re-written as:

$$(1 + 0.1264B^{12} + 0.1851B^{24})\nabla(GDP_t - 139.8089t) = (1 - 0.1697B)W_t$$

Comments on model fit and diagnositics

```
gdp_model = sarima(gdp_series, 0,1,1,2,0,0,12) #use sarima for diagnositic plots
```

```
## initial  value 6.308917
## iter   2 value 6.277719
## iter   3 value 6.277528
## iter   4 value 6.277527
## iter   4 value 6.277527
## iter   4 value 6.277527
## final  value 6.277527
## converged
## initial  value 6.322357
## iter   2 value 6.321121
## iter   3 value 6.321085
## iter   4 value 6.321084
## iter   4 value 6.321084
## iter   4 value 6.321084
## final  value 6.321084
## converged
```



From the standardized residual plots, we can see that the residuals have more or less constant variance and mean (fluctuates at around 0). The ACF plot also don't have any significant autocorrelation, which is good. The Normal QQ Plot also has more or less a good fit, with very little deviations around the tails. All p-values of the Ljung-box are above the 5% significance level, indicating there are no significant auto-correlaion between the residuals (similar to White Noise). Overall, this ARIMA model has a pretty good fit for the series.

Fitting a regression model with ARIMA errors for GDP, with the seasonally adjusted employment as the external regressor

Now, I will experiment with with a regression model with ARIMA errors for GDP series, with the seasonally adjusted employment `emp` series as the external regressor. For consistency with the previous ARIMA model, we will use `AIC` as the model selection criteria and also use the MAE of the residuals for comparisons.

```
seas_emp = seasadj(dcmp_X11) #fitted series

xreg_model = auto.arima( gdp_series, xreg = seas_emp, ic = "aic")

#model summary
summary(xreg_model)
```

```
## Series: gdp_series
## Regression with ARIMA(0,1,1)(2,0,1)[12] errors
##
## Coefficients:
##           ma1    sar1     sar2     sma1     drift     xreg
##       -0.1978  0.4697  -0.1195  -0.6795  120.2946  12.4354
## s.e.   0.0571  0.1727   0.0854   0.1706   15.1633   4.1674
##
## sigma^2 estimated as 298781:  log likelihood=-2122.09
## AIC=4258.18   AICc=4258.6   BIC=4283.5
##
## Training set error measures:
##                     ME     RMSE     MAE         MPE      MAPE      MASE
## Training set 1.218352 539.6328 404.233 -0.01188335 0.5967106 0.1991834
##                      ACF1
## Training set -0.01738599
```

The MAE for this model is:

```
#MAE
#mae(gdp_series, xreg_model$fitted ) #using library(Metrics)
mean( abs( gdp_series - fitted(xreg_model) ), na.rm = T )
```

```
## [1] 404.233
```

This fitted model can be re-written as:

$$(1 - 0.4697B^{12} + 0.1195B^{24})\nabla(GDP_t - 120.2946t - 12.4354EMP_t) = (1 - 0.1978B^{12})(1 - 0.1978B)W_t$$

Fitting a bivariate VAR model to GDP and the seasonally adjusted employment (emp) series

Finally, I will fit a bivariate VAR model to `GDP` and the seasonally adjusted employment (`emp`) series to account for a pontential casuality relationship between the two series. Again, we will also use `AIC` as the model selection criteria and the MAE of the residuals for consistent comparisons.

```r
binded = cbind(gdp_series, seas_emp) #binding both series

VARselect(binded) #AIC criteria is order 2
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2
##
## $criteria
##                   1            2            3            4            5
## AIC(n) 1.708496e+01 1.702038e+01 1.703381e+01 1.705478e+01 1.707132e+01
## HQ(n)  1.711744e+01 1.707450e+01 1.710958e+01 1.715220e+01 1.719038e+01
## SC(n)  1.716579e+01 1.715509e+01 1.722242e+01 1.729727e+01 1.736770e+01
## FPE(n) 2.629699e+07 2.465239e+07 2.498630e+07 2.551641e+07 2.594295e+07
##                   6            7            8            9           10
## AIC(n) 1.708948e+01 1.711496e+01 1.713694e+01 1.715674e+01 1.717428e+01
## HQ(n)  1.723019e+01 1.727733e+01 1.732095e+01 1.736240e+01 1.740159e+01
## SC(n)  1.743974e+01 1.751912e+01 1.759498e+01 1.766867e+01 1.774010e+01
## FPE(n) 2.641999e+07 2.710432e+07 2.770951e+07 2.826769e+07 2.877278e+07
```

```r
var_model = VAR(binded, 2)
var_model$varresult$gdp_series
```

```
##
## Call:
## lm(formula = y ~ -1 + ., data = datamat)
##
## Coefficients:
## gdp_series.l1    seas_emp.l1  gdp_series.l2    seas_emp.l2          const
##        0.8135         2.3514         0.1928        -2.8821       175.0922
```

```r
summary(var_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gdp_series, seas_emp
## Deterministic variables: const
## Sample size: 274
## Log Likelihood: -3096.027
## Roots of the characteristic polynomial:
## 1.002 0.9253 0.2367 0.2012
## Call:
## VAR(y = binded, p = 2)
##
##
## Estimation results for equation gdp_series:
## ===========================================
## gdp_series = gdp_series.l1 + seas_emp.l1 + gdp_series.l2 + seas_emp.l2 + const
##
##                Estimate Std. Error t value Pr(>|t|)
## gdp_series.l1   0.81348    0.06124  13.285  < 2e-16 ***
## seas_emp.l1     2.35145    4.00401   0.587  0.55751
## gdp_series.l2   0.19279    0.06260   3.080  0.00229 **
## seas_emp.l2    -2.88207    3.96058  -0.728  0.46744
```

```
## const         175.09218  579.58812   0.302  0.76281
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 570.4 on 269 degrees of freedom
## Multiple R-Squared: 0.9969,  Adjusted R-squared: 0.9969
## F-statistic: 2.181e+04 on 4 and 269 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation seas_emp:
## =========================================
## seas_emp = gdp_series.l1 + seas_emp.l1 + gdp_series.l2 + seas_emp.l2 + const
##
##                Estimate Std. Error t value Pr(>|t|)
## gdp_series.l1 5.548e-04  9.138e-04   0.607 0.544259
## seas_emp.l1   6.756e-01  5.975e-02  11.308  < 2e-16 ***
## gdp_series.l2 2.752e-04  9.342e-04   0.295 0.768570
## seas_emp.l2   2.248e-01  5.910e-02   3.803 0.000177 ***
## const         2.787e+01  8.649e+00   3.223 0.001427 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 8.512 on 269 degrees of freedom
## Multiple R-Squared: 0.9898,  Adjusted R-squared: 0.9896
## F-statistic:  6502 on 4 and 269 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##            gdp_series seas_emp
## gdp_series   325368.4   608.45
## seas_emp        608.5    72.45
##
## Correlation matrix of residuals:
##            gdp_series seas_emp
## gdp_series     1.0000   0.1253
## seas_emp       0.1253   1.0000
```

The MAE for this model is:

```
#MAE
gdp_residuals = var_model$varresult$gdp_series$residuals
mean(abs(gdp_residuals))
```

```
## [1] 433.7605
```

```
#mean( abs( gdp_series[-(1:2)] - fitted(var_model)[,"gdp_series"] ) )
```

Now, we will perform a Granger-causality test to check if employment helps predict GDP.

```
causality(var_model, cause = "seas_emp")
```

```
## $Granger
##
##  Granger causality H0: seas_emp do not Granger-cause gdp_series
```

```
##
## data:  VAR object var_model
## F-Test = 0.27497, df1 = 2, df2 = 538, p-value = 0.7597
##
##
## $Instant
##
##  H0: No instantaneous causality between: seas_emp and gdp_series
##
## data:  VAR object var_model
## Chi-squared = 4.2365, df = 1, p-value = 0.03956
```

The p-value is 0.7597, so we fail to reject the null hypothesis that `seas_emp` (employment series) do not Granger-cause GDP (`gdp_series`), i.e., employment might not help predict GDP, beyond the past of using the GDP series itself.

Conclusion

Hence, the regression model with ARIMA errors for GDP, with the seasonally adjusted employment (`seas_emp`) has the smallest MAE, so we will use this model. This is also consistent with the Granger causality test. Now, we will first fit this model again to all but the last year of data (i.e. exclude the last 12 observations).

```
#Arima() model using previous parameter with seas_emp as xreg
out_sample_model = Arima( gdp_series[1:(length(gdp_series)-12)], order = c(0,1,1),
                  seasonal = list( order = c(2,0,1), period = 12),
                 xreg = seas_emp[1:(length(seas_emp)-12)], include.drift = T )

summary(out_sample_model)
```

```
## Series: gdp_series[1:(length(gdp_series) - 12)]
## Regression with ARIMA(0,1,1)(2,0,1)[12] errors
##
## Coefficients:
##           ma1    sar1     sar2     sma1     drift     xreg
##       -0.1716  0.5106  -0.1188  -0.7214  119.1643  11.4048
## s.e.   0.0575  0.1627   0.0858   0.1604   14.7760   4.1143
##
## sigma^2 estimated as 277142:  log likelihood=-2019.74
## AIC=4053.48   AICc=4053.92   BIC=4078.49
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE      MAPE      MASE
## Training set 1.304234 519.4167 390.5297 -0.01045756 0.5873175 0.8932799
##                      ACF1
## Training set -0.01535215
```

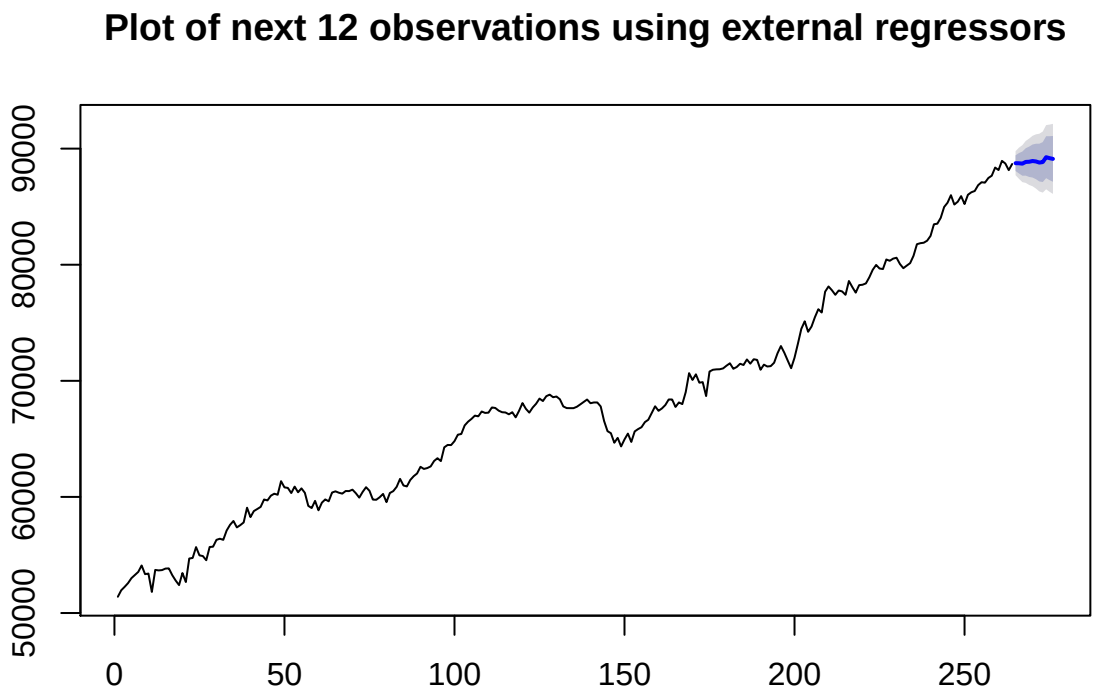
Then, we will use this model to forecast the last year of GDP data, and report the MAE of the forecasts versus the actual GDP values of the last year (out of sample prediction).

```
#next 12 predictions
forecasts = forecast(out_sample_model, xreg = seas_emp[length(seas_emp)-(11:0)], h=12)
predictions = forecasts$mean
predictions
```

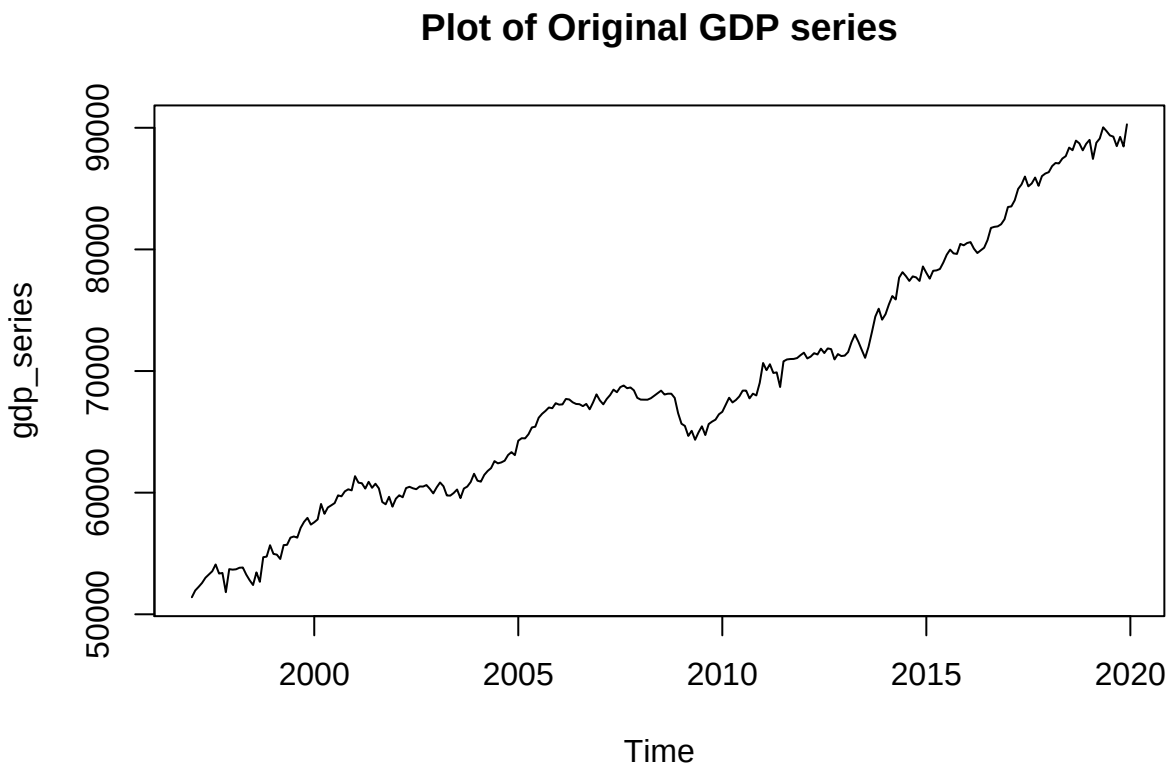
```
## Time Series:
## Start = 265
## End = 276
```

```
## Frequency = 1
##  [1] 88747.97 88740.87 88706.40 88855.05 88868.29 88932.45 88895.12 88799.30
##  [9] 88842.20 89262.70 89178.55 89117.50
```

```
#plot of forecasts
plot(forecasts, main="Plot of next 12 observations using external regressors")
```




```
#plot of original series
plot(gdp_series, main="Plot of Original GDP series")
```



The resulting out-of-sample MAE (using our original last 12 observations minus our predictions) is:

```
#out of sample MAE
mean( abs( gdp_series[ (length(gdp_series)-(11:0)) ] - predictions ) )
```

```
## [1] 586.9364
```

This MAE is quite large (larger than our original MAE without leaving out the last 12 observations). Hence, I have decided to repeat the same procedure again, but using the simple ARIMA model this time.

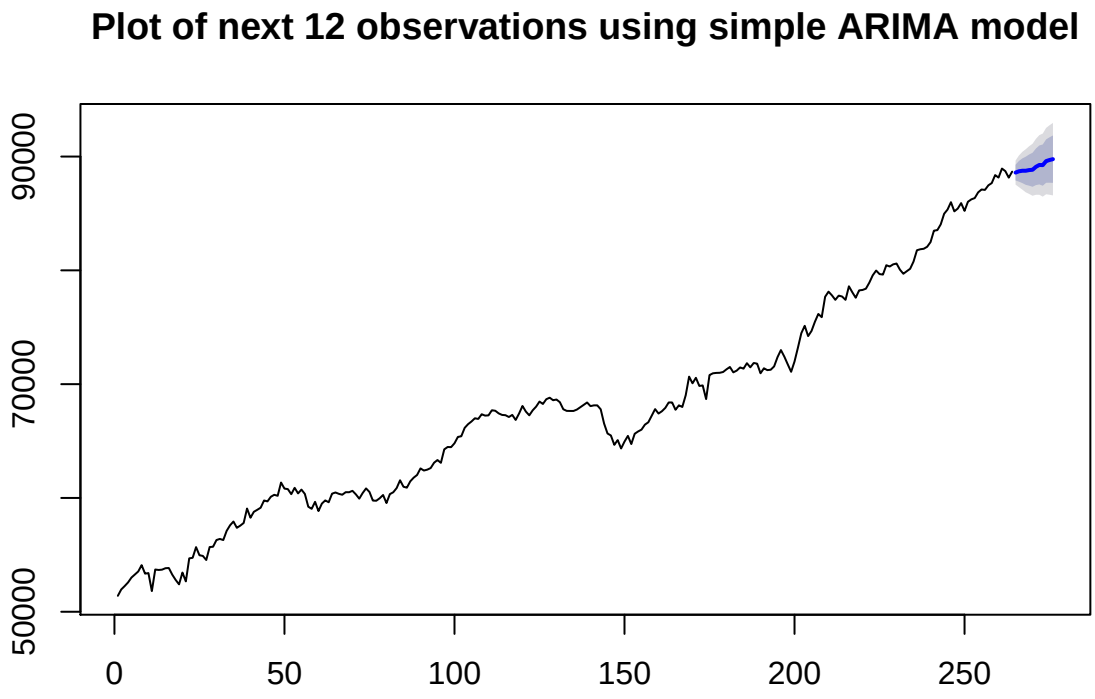
```
#Repeating the same step using simple ARIMA()
out_arima_simple = forecast::Arima( gdp_series[1:(length(gdp_series)-12)], order = c(0,1,1),
                  seasonal = list( order = c(2,0,0), period = 12),
                 include.drift = T )
summary(out_arima_simple)
```

```
## Series: gdp_series[1:(length(gdp_series) - 12)]
## ARIMA(0,1,1)(2,0,0)[12] with drift
##
## Coefficients:
##           ma1     sar1     sar2     drift
##       -0.1414  -0.1267  -0.2043  139.1388
## s.e.   0.0579   0.0631   0.0657   21.6752
##
## sigma^2 estimated as 289921:  log likelihood=-2025.67
## AIC=4061.35   AICc=4061.58   BIC=4079.21
##
## Training set error measures:
```

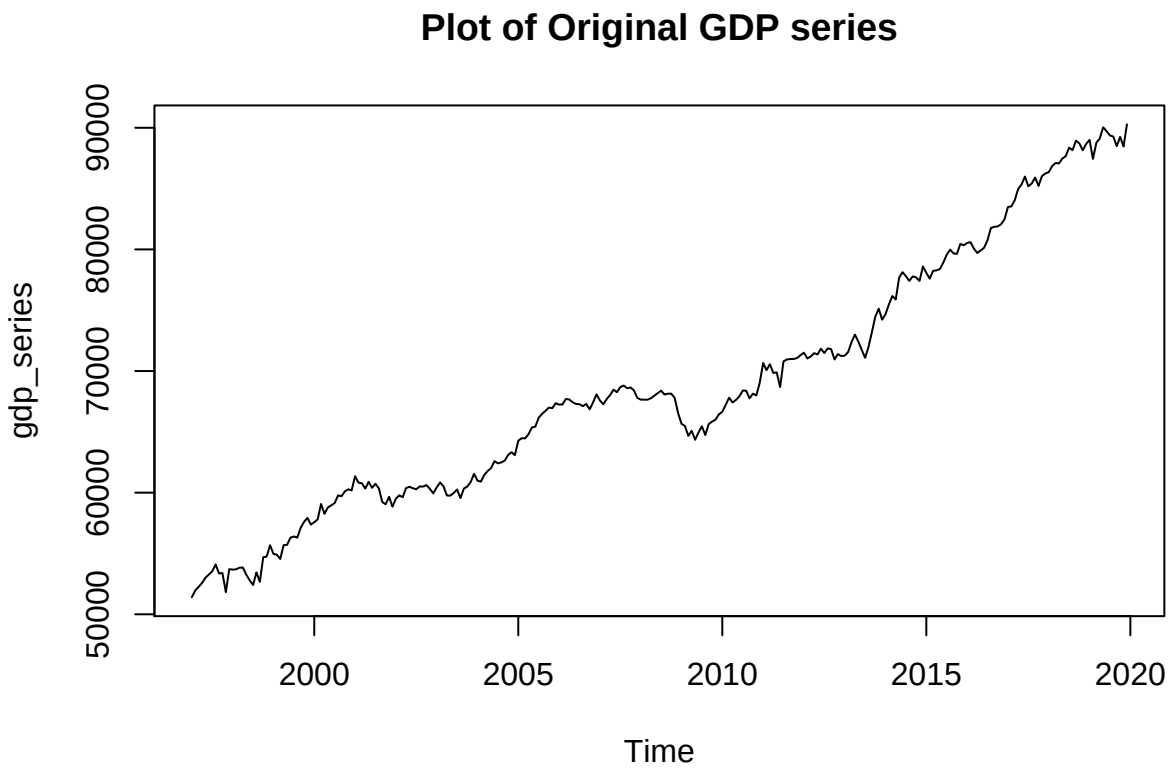
```
##                       ME    RMSE      MAE         MPE      MAPE      MASE
## Training set 0.7729016 533.32 403.6431 -0.01106516 0.6086997 0.9232747
##                       ACF1
## Training set -0.009992438
```

```r
forecast_arima = forecast::forecast(out_arima_simple, h = 12)
prediction_arima = forecast_arima$mean

#plot of forecasts
plot(forecast_arima, main="Plot of next 12 observations using simple ARIMA model")
```




```r
#plot of original series
plot(gdp_series, main="Plot of Original GDP series")
```




```
#Comparisons of the 2 models' last 12 predictions
as.numeric(predictions) #with emp as external regressor
```

```
##  [1] 88747.97 88740.87 88706.40 88855.05 88868.29 88932.45 88895.12 88799.30
##  [9] 88842.20 89262.70 89178.55 89117.50
```

```
as.numeric(prediction_arima) #simple ARIMA
```

```
##  [1] 88590.90 88703.23 88752.65 88752.23 88811.18 88838.61 89099.63 89262.87
##  [9] 89248.84 89602.47 89696.96 89771.21
```

```
gdp_series[ (length(gdp_series)-(11:0))] #original last 12 observations
```

```
##  [1] 89007 87441 88771 89117 90042 89725 89372 89273 88491 89253 88464 90283
```

The resulting MAE of using the simple ARIMA model is:

```
#resulting MAE
mean( abs( gdp_series[ (length(gdp_series)-(11:0)) ] - prediction_arima ) )
```

```
## [1] 609.4331
```

This out-of-sample MAE is larger than that of our previous model's MAE of 586.9364 (regression with ARIMA errors using employment series as external regressors). This again concludes that the external regressor model is indeed the optimal model in terms of MAE and the only reason that this model has a relatively high MAE for the out-of-sample prediction is only due to the nature of our observations (especially the last 12 observations), which is not easily captured with any of our above models.