Time Series Analysis and Prediction

Tianze Hua

Description The goal is to fit a model for predicting "current" GDP, call it Y_t, based on current and lagged values of the other variables (e.g. $X_{1,t}, X_{1,t-1}, X_{2,t}$) and possibly lagged values of GDP (Y_{t-1}). For this, you will use VAR and regression with ARMA error models.

Note: Most economic time-series are integrated of order 1, so you might need to difference the data

1. Plot of the (nominal) GDP series and perform an `adf.test` for stationarity. Report the p-value and the conclusion for your series (integrated or stationary).

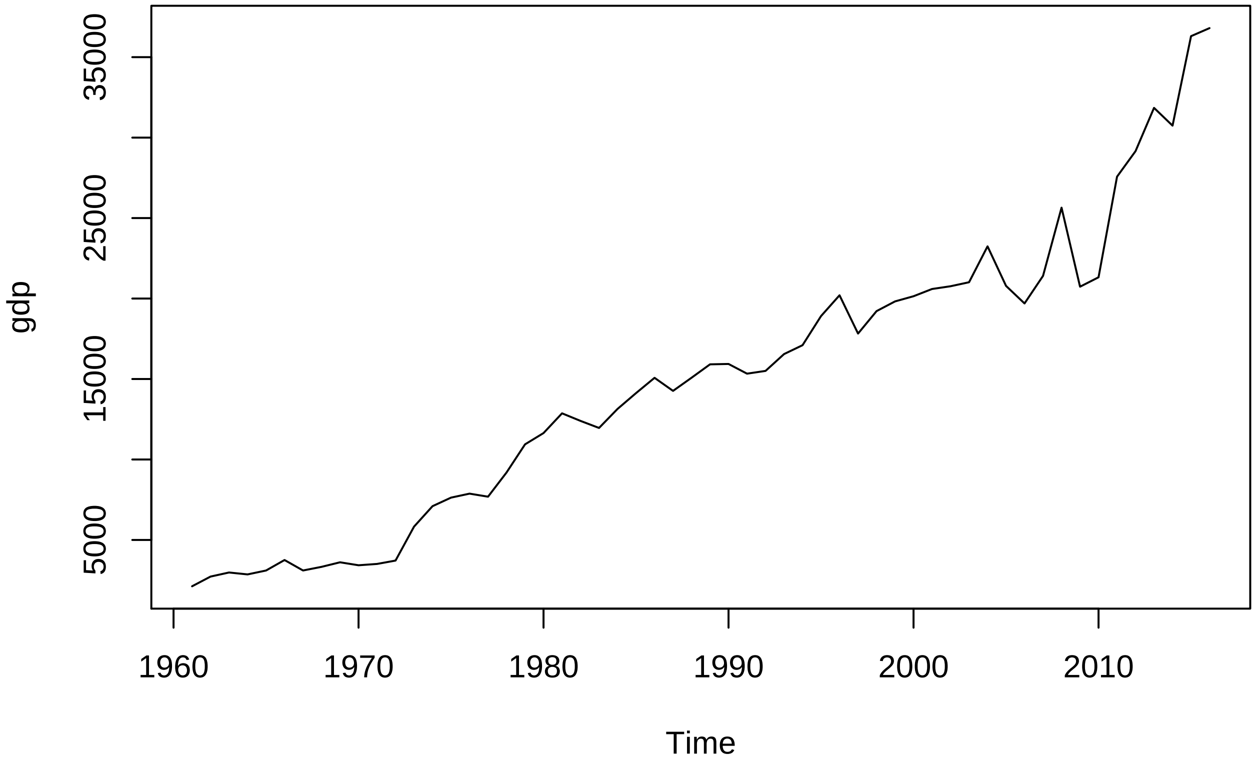
```
library(cansim)
```

```
## Warning: The packages `ellipsis` (>= 0.3.2) and `vctrs` (>= 0.3.8) are required
## as of rlang 1.0.0.

## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_unnamed' by
## 'rlang::check_dots_unnamed' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_used' by
## 'rlang::check_dots_used' when loading 'tibble'

## Warning: replacing previous import 'ellipsis::check_dots_empty' by
## 'rlang::check_dots_empty' when loading 'tibble'
```

```
library(tidyverse)

# Data for Agriculture, forestry, fishing and hunting; Canada
# Gross domestic product (GDP) (dollars x 1,000,000)
gdp = get_cansim_vector( "v41713154", start_time = "1961-01-01", end_time = "2016-12-01") %>%
  pull(VALUE) %>% ts( start = c(1961,1), frequency = 1)
```

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```r
plot(gdp)
```



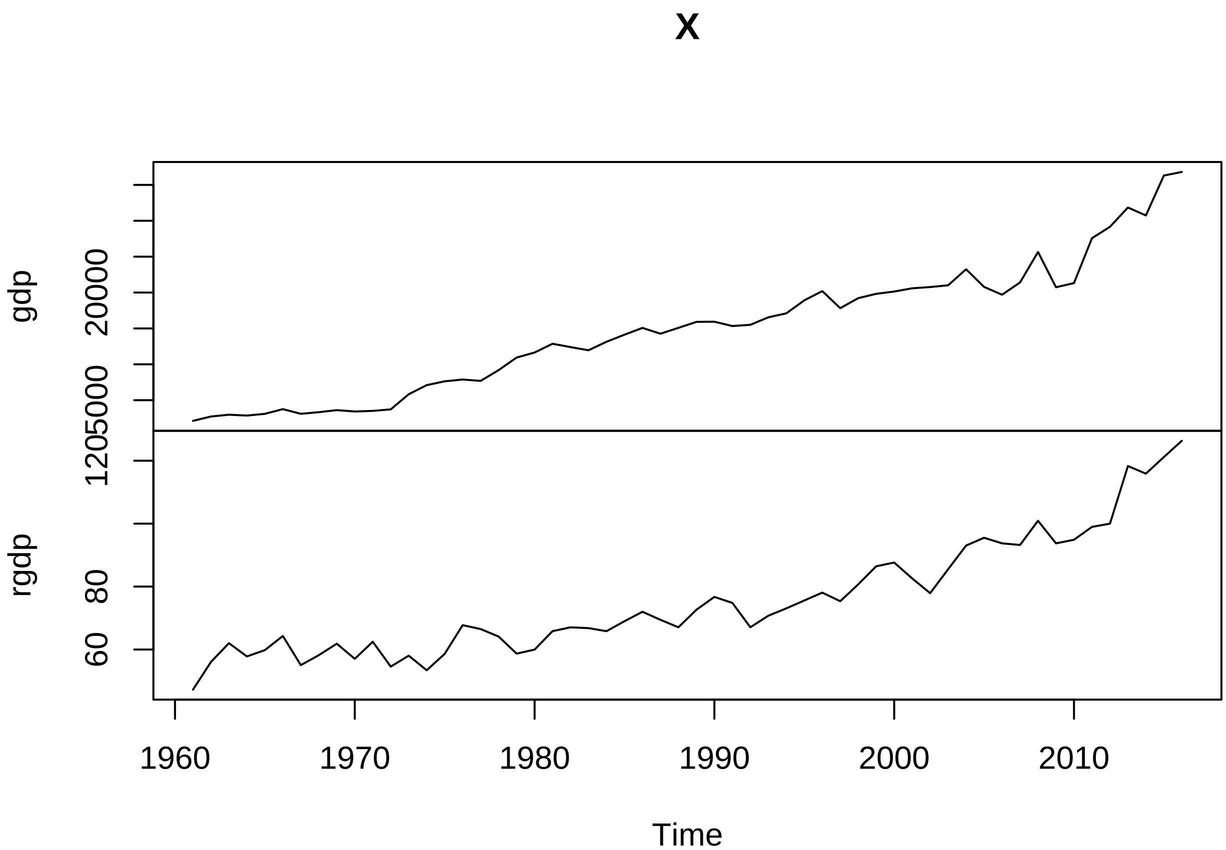

```r
tseries::adf.test(gdp)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  gdp
## Dickey-Fuller = -0.35978, Lag order = 3, p-value = 0.9849
## alternative hypothesis: stationary
```

The GDP series looks like a RW, and this is confirmed by the ADF test, which fails to reject the null hypothesis of non-stationarity with a p-value close to 1.

2. Fit a bivariate VAR(1) model on (nominal) GDP and Real GDP. Do not transform the series, but include both constant and trend term in your model. Report the coefficient matrix and check whether the model is stationary, i.e. its eigen-values are within the unit disk (use functions `eigen` and `Mod`).

```r
# Real gross domestic product (GDP)
rgdp = get_cansim_vector( "v41712933", start_time = "1961-01-01", end_time = "2016-12-01") %>%
  pull(VALUE) %>% ts( start = c(1961,1), frequency = 1)

X = cbind( gdp, rgdp)
plot(X)
```




```
library(vars)
out.var = VAR( X, lag.max = 1, type = "both" )
summary(out.var)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: gdp, rgdp
## Deterministic variables: both
## Sample size: 55
## Log Likelihood: -639.898
## Roots of the characteristic polynomial:
## 0.7755 0.7458
## Call:
## VAR(y = X, type = "both", lag.max = 1)
##
##
## Estimation results for equation gdp:
## ====================================
## gdp = gdp.l1 + rgdp.l1 + const + trend
##
##          Estimate Std. Error t value Pr(>|t|)
## gdp.l1     0.7811     0.1234   6.331 6.19e-08 ***
## rgdp.l1   -0.3129    36.7849  -0.009   0.9932
## const    -15.9815  1782.2488  -0.009   0.9929
## trend    132.6801    59.7481   2.221   0.0308 *
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1644 on 51 degrees of freedom
## Multiple R-Squared: 0.968,   Adjusted R-squared: 0.9661
## F-statistic: 513.6 on 3 and 51 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation rgdp:
## =====================================
## rgdp = gdp.l1 + rgdp.l1 + const + trend
##
##           Estimate Std. Error t value Pr(>|t|)
## gdp.l1   0.0006334  0.0003718   1.704   0.0945 .
## rgdp.l1  0.7401645  0.1108587   6.677 1.76e-08 ***
## const   12.4294006  5.3711579   2.314   0.0247 *
## trend   -0.0263425  0.1800628  -0.146   0.8843
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 4.953 on 51 degrees of freedom
## Multiple R-Squared: 0.9305,  Adjusted R-squared: 0.9264
## F-statistic: 227.5 on 3 and 51 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##          gdp    rgdp
## gdp  2701346 3928.41
## rgdp    3928   24.53
##
## Correlation matrix of residuals:
##         gdp   rgdp
## gdp  1.0000 0.4825
## rgdp 0.4825 1.0000
```

The model is

$$\begin{bmatrix} GDP_t \\ rGDP_t \end{bmatrix} = \overbrace{\begin{bmatrix} -15.9815 \\ 12.4294006 \end{bmatrix}}^{const} + \overbrace{\begin{bmatrix} 132.6801 \\ -0.0263425 \end{bmatrix}}^{trend} + \begin{bmatrix} 0.7811 & -0.3129 \\ 0.0006334 & 0.7401645 \end{bmatrix} \begin{bmatrix} GDP_{t-1} \\ rGDP_{t-1} \end{bmatrix} + \begin{bmatrix} W_{1,t} \\ W_{2,t} \end{bmatrix} \quad \begin{bmatrix} W_{1,t} \\ W_{2,t} \end{bmatrix} \sim \mathrm{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 2701346 & 392 \\ 3928.41 & 24 \end{bmatrix}$$

To check for stationarity:

```r
(Phi = matrix( c( out.var$varresult$gdp$coefficients[1:2],
                  out.var$varresult$rgdp$coefficients[1:2]
                  ), 2, byrow = T))
```

```
##              [,1]       [,2]
## [1,] 0.7810749004 -0.3128808
## [2,] 0.0006334209  0.7401645
```

```r
(eigen_vals = eigen(Phi)$values)
```

```
## [1] 0.7754599 0.7457795
```

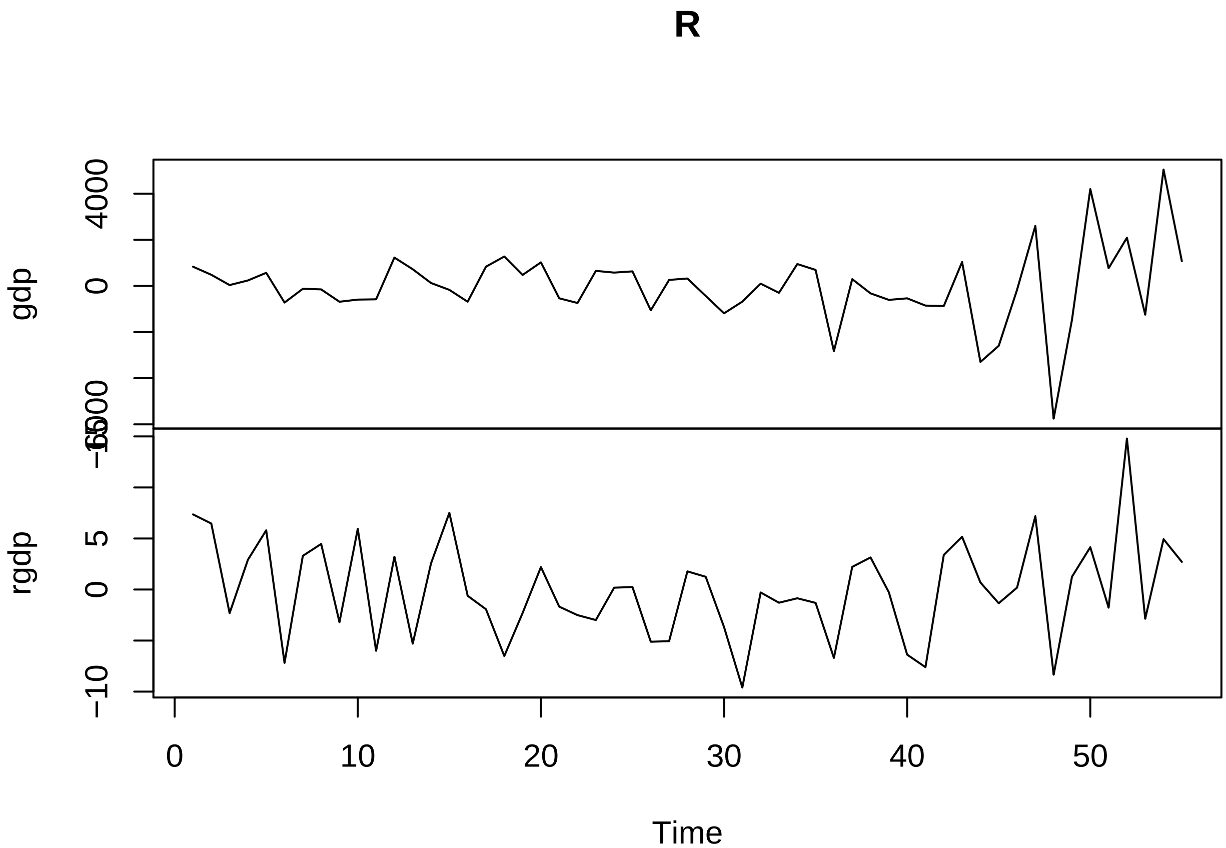
```r
(Mod(eigen_vals) < 1)
```

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

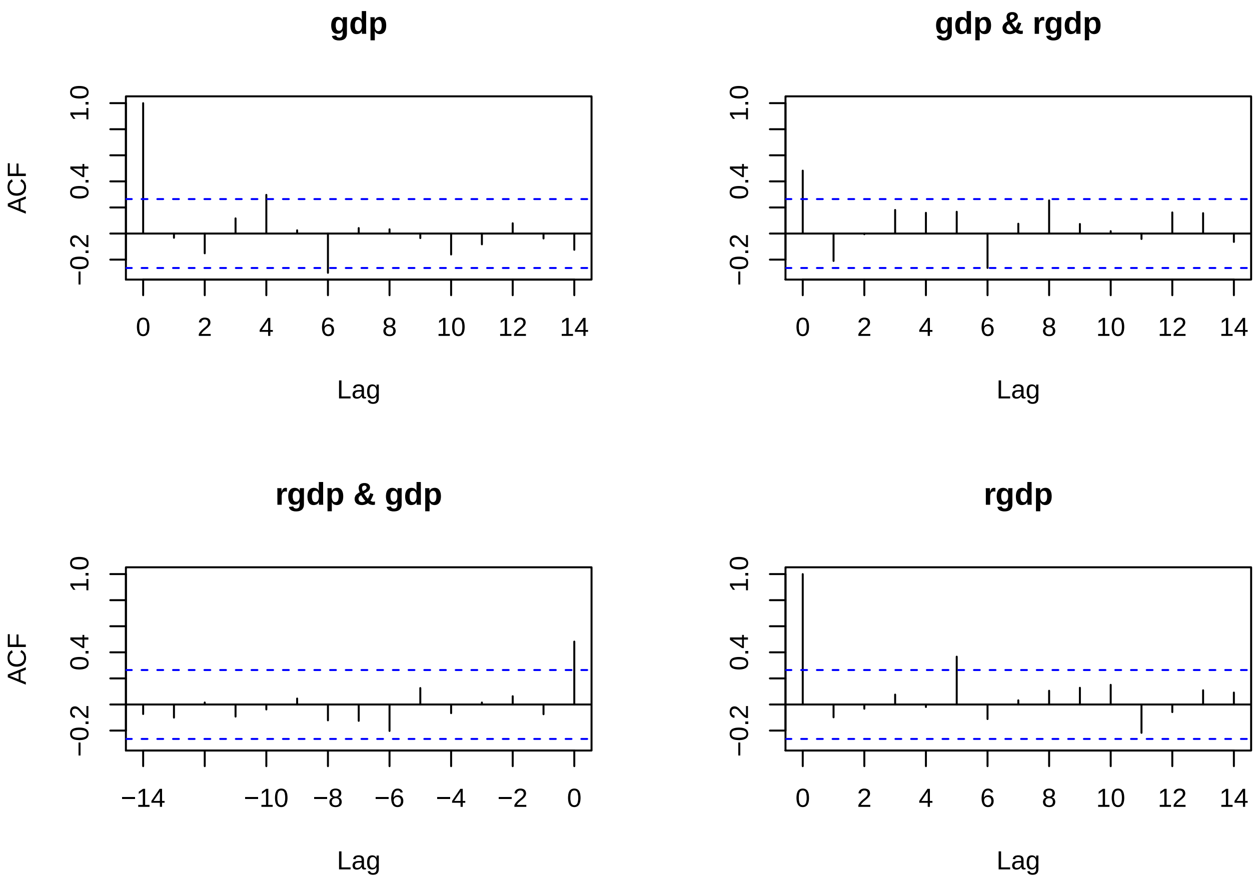
Since both eigen-values of the $\mathbf{\Phi}_1$ matrix are less than one, the model is stationary.

3. Plot the residuals and their ACF/CCF from the previous VAR(1) model, and comment on its fit. Report the residual MAPE for (nominal) GDP only.

```r
R = ts( residuals(out.var) )
plot(R)
```




```r
acf(R)
```




```
# Can also perform a Ljung-Box type test, with
vars::serial.test(out.var, lags.pt = 10)
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object out.var
## Chi-squared = 43.893, df = 36, p-value = 0.1718
```

The residuals for nominal GDP do not look stationary, as their variance is fanning out. This is an indication that a log-transformation is necessary (i.e. model log-differences, or continuously compounded increase rates). The residuals seem generally uncorrelated.

The MAPE for nominal GDP is

```
mean( abs(gdp[-1] - fitted(out.var)[,1]) / gdp[-1] )
```

```
## [1] 0.08167714
```

4. Now fit an ARMA-error regression model for (nominal) GDP (Y_t) with simultaneous Real GDP (X_t) as the external regressor. Use `forecast::auto.arima` to select the order of the model (including differencing) and report the final model, its AIC and MAPE.

```
library(forecast)
out.arimax = auto.arima( gdp, xreg = rgdp)
summary(out.arimax)
```

```
## Series: gdp
## Regression with ARIMA(2,1,0) errors
```

```
##
## Coefficients:
##           ar1      ar2     drift      xreg
##       -0.0554  -0.4499  382.2726  169.4405
## s.e.   0.1264   0.1326  134.2086   37.7453
##
## sigma^2 estimated as 2020581:  log likelihood=-475.46
## AIC=960.92   AICc=962.15   BIC=970.96
##
## Training set error measures:
##                     ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -15.63917 1356.529 1031.996 -2.660239 10.07983 0.8665007
##                     ACF1
## Training set -0.02131251
```

The fitted model is

$$(1 + 0.0554B + 0.4499B^2)\nabla(GDP_t - 382.2726t - 169.4405rGDP_t) = W_t$$

with $AIC = 960.92$ and $MAPE = 10.07983$

An alternative approach would be to model the log-GDP.

```
lgdp = log(gdp)
out.log.arimax = auto.arima(lgdp , xreg = rgdp)
summary(out.log.arimax)
```

```
## Series: lgdp
## Regression with ARIMA(0,1,1) errors
##
## Coefficients:
##          ma1   drift    xreg
##       0.4802  0.0351  0.0117
## s.e.  0.1136  0.0184  0.0021
##
## sigma^2 estimated as 0.008863:  log likelihood=53.33
## AIC=-98.67   AICc=-97.87   BIC=-90.64
##
## Training set error measures:
##                         ME       RMSE        MAE         MPE      MAPE
## Training set -0.0006410015 0.09071949 0.07016652 0.007554853 0.7563111
##                   MASE        ACF1
## Training set 0.7906418 -0.03801071
```

Note that the AIC is not comparable because of the transformation, but we can compare MAPE's for the original data.

```
mean( abs( gdp - exp( fitted(out.log.arimax) ) ) / gdp )
```

```
## [1] 0.07010105
```

The log-model gives a MAPE of 7.01%, which is better than the previous model's.

5. Finally, fit an ARMA-error regression model for (nominal) GDP with any of the other variables (Real GDP, Labour/Capital productivity/input/cost, etc.) as external regressors, simultaneous or lagged. Find a model that gives a better AIC than the previous part, or report three different models that you tried with worse AIC. Report the best-AIC model's MAPE and plot its diagnostics, commenting briefly on its fit.

Consider the additional external variables:

```
# Multifactor productivity
mfp = get_cansim_vector( "v41712882", start_time = "1961-01-01", end_time = "2016-12-01") %>%
  pull(VALUE) %>% ts( start = c(1961,1), frequency = 1)
# Labour input
lin = get_cansim_vector( "v41712950", start_time = "1961-01-01", end_time = "2016-12-01") %>%
  pull(VALUE) %>% ts( start = c(1961,1), frequency = 1)
# Capital input
cin = get_cansim_vector( "v41713052", start_time = "1961-01-01", end_time = "2016-12-01") %>%
  pull(VALUE) %>% ts( start = c(1961,1), frequency = 1)
# Combined labour and capital inputs
clcin = get_cansim_vector( "v41713137", start_time = "1961-01-01", end_time = "2016-12-01") %>%
  pull(VALUE) %>% ts( start = c(1961,1), frequency = 1)
```

Trying out different models on the raw data, we get:

```
auto.arima( gdp, xreg = mfp) %>% AIC
```

```
## [1] 968.5178
```

```
auto.arima( gdp, xreg = lin) %>% AIC
```

```
## [1] 963.3589
```

```
auto.arima( gdp, xreg = cin) %>% AIC
```

```
## [1] 954.0566
```

```
auto.arima( gdp, xreg = clcin) %>% AIC
```

```
## [1] 963.721
```

```
auto.arima( gdp, xreg = cbind(rgdp)) %>% AIC
```

```
## [1] 960.9233
```

The best model seems to be the one with only Capital Input as an external regressor, giving a MAPE of 7.78%.

```
out.arimax.best = auto.arima( gdp, xreg = cin)
summary(out.arimax.best)
```

```
## Series: gdp
## Regression with ARIMA(4,2,0) errors
##
## Coefficients:
##           ar1     ar2      ar3      ar4      xreg
##       -1.2357  -1.253  -0.9536  -0.3813  153.7112
## s.e.   0.1260   0.168   0.1617   0.1321   57.0536
##
## sigma^2 estimated as 2318603:  log likelihood=-471.03
## AIC=954.06   AICc=955.84   BIC=965.99
##
## Training set error measures:
##                    ME     RMSE      MAE       MPE     MAPE      MASE
## Training set 140.9118 1424.352 1008.822 0.3039079 7.781164 0.8470426
##                     ACF1
## Training set -0.03289393
```

Considering the log-transformed data

```
auto.arima( lgdp, xreg = mfp) %>% AIC
```

```
## [1] -94.12962
```

```
auto.arima( lgdp, xreg = lin) %>% AIC
```

```
## [1] -90.04389
```

```
auto.arima( lgdp, xreg = cin) %>% AIC
```

```
## [1] -90.14638
```

```
auto.arima( lgdp, xreg = clcin) %>% AIC
```

```
## [1] -100.0951
```

```
auto.arima( lgdp, xreg = cbind(clcin, rgdp)) %>% AIC
```

```
## [1] -104.3829
```

The best model includes Combined Labour & Capital Input and Real GDP as regressors, and also includes a drift term.

```
out.log.arimax.best = auto.arima( lgdp, xreg = cbind(clcin,rgdp))
summary(out.log.arimax.best)
```

```
## Series: lgdp
## Regression with ARIMA(0,1,0) errors
##
## Coefficients:
##        drift   clcin    rgdp
##       0.0423  0.0201  0.0074
## s.e.  0.0122  0.0051  0.0023
##
## sigma^2 estimated as 0.008026:  log likelihood=56.19
## AIC=-104.38   AICc=-103.58   BIC=-96.35
##
## Training set error measures:
##                        ME       RMSE        MAE        MPE      MAPE      MASE
## Training set 9.159832e-05 0.08632989 0.06751501 0.01627714 0.7293777 0.7607644
##                      ACF1
## Training set 0.0009606225
```

Note this is a simple regression model on the differenced series. It's MAPE in terms of the original data is 6.7%:

```
mean( abs( gdp - exp( fitted(out.log.arimax.best) ) ) / gdp )
```

```
## [1] 0.06760491
```

6. The in-sample MAPE used above is a biased measure of predictive performance. A better measure is given by using time series cross-validation, as described in chapter 3.4 of fpp2. For this part, you have to evaluate the predictive performance of your previous model using TS cross-validation on the last 10 available GDP values. More specifically, create a loop for $i = 1, \ldots, 10$ and do the following:

- Fit the model specification you chose in the previous part to the data from 1961 to $2006 + i = n_i$.
- Use the model to create a 1-step-ahead forecast for (nominal) GDP, call it $Y^{n_i}_{n_i+1}$; make sure to use the appropriate regressor values for *newxreg*.

- Calculate the percentage error: $|Y_{n_i+1} - Y_{n_i+1}^{n_i}|/Y_{n_i+1}$
  In the end, average the percentage errors over all i and report the resulting MAPE value.
  (Note: this will give you a more objective measure of predictive performance, because you are only using *out-of-sample* 1-step-ahead forecasts.)

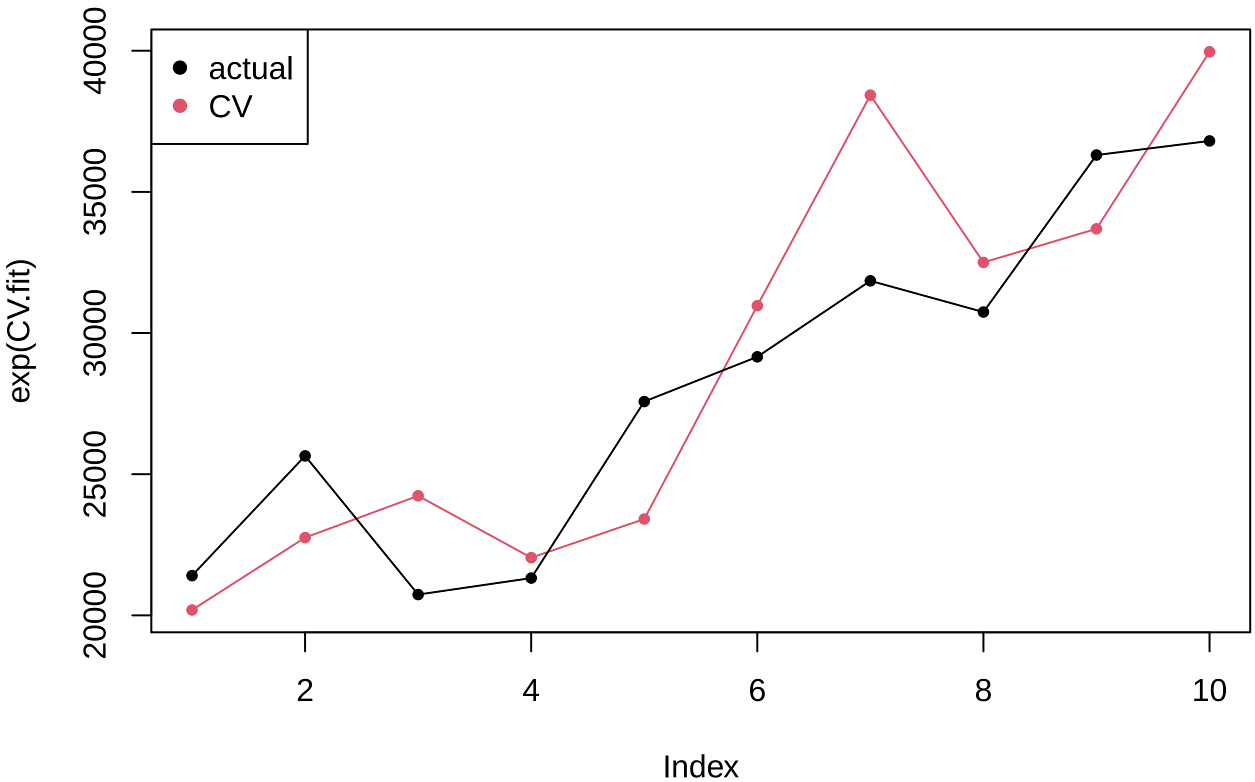
Using the model for the log-GDP

```
n = length(lgdp)
Xreg = cbind( clcin, rgdp)

CV.fit = rep(0,10) # placeholder for cross-validation forecasts

for(i in 1:10){
  # create increasing series
  lgdp.tmp = lgdp[1:(n-11+i)]
  xreg.tmp = Xreg[1:(n-11+i),]
  # fit model
  out.tmp = Arima( lgdp.tmp, order = c(0,1,0), xreg = xreg.tmp, include.drift = T )
  #
  CV.fit[i] = forecast( out.tmp, xreg = t( Xreg[n-10+i,] ) )$mean
}

actual = gdp[(n-9):n]
mean( abs( actual - exp(CV.fit) ) / actual )
```

```
## [1] 0.1006747
```

The Cross-Validation MAPE is 9.68%. Below is a plot of the (un-transformed) actual data and cross-validated predictions.

```
plot( exp(CV.fit), col = 2, type = "o", pch = 20); lines( actual, type = "o", pch = 20);
legend("topleft", col = 1:2, legend = c("actual", "CV"), pch = 16)
```


actual
CV
exp(CV.fit)
20000
25000
30000
35000
40000
2
4
6
8
10
Index