Forecasting Canadian GDP: a multivariate ARMA-error regression model

Timothy Lee

2020/6/9

Intro

This report is part of October 2019's Statistics Canada: Business Data Scientist Challenge. The goal of this challenge is to create timely estimates of current GDP based on other, more readily available information; (also referred to as `nowcasting`). For simplicity, I have chosen to only work on the Sector/Industry Group of `Retail Trade`, where the data is obtained StatCan Table: 36-10-0208-01 called "Multifactor productivity, value-added, capital input and labour input in the aggregate business sector and major sub-sectors, by industry". The data is also selected with selected with the `North American Industry Classification System (NAICS)` filter and contains annual data from 1961-2018 for a range of economic variables, such as `Labour Productivity`, `Capital Productivity`, `Multifactor Productivity`, etc.

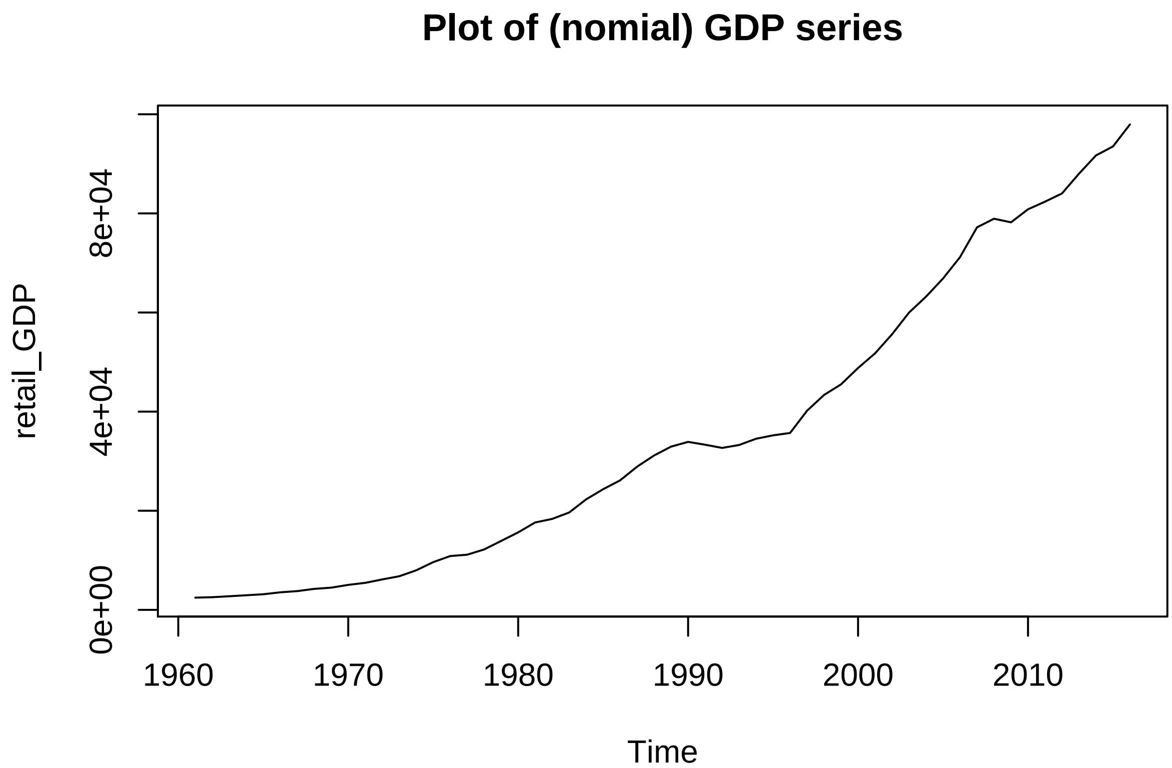
```
library(cansim)
library(tidyverse)

retail_real_GDP = get_cansim_vector( "v41712939", start_time = "1961-01-01",
                                     end_time = "2018-12-01") %>% pull(VALUE) %>%
  ts( start = 1961, end = 2018)
#start 1961, ends in 2018

#(nominal)
retail_GDP = get_cansim_vector( "v41713160", start_time = "1961-01-01",
                                end_time = "2016-12-01") %>% pull(VALUE) %>%
  ts(start = 1961, end = 2016)
#start 1961, ends in 2016
```

Plot of the (nominal) GDP series for `retail trade` sector:

```
library(tseries)
plot(retail_GDP, main= "Plot of (nomial) GDP series")
```




```r
adf.test(retail_GDP)
```

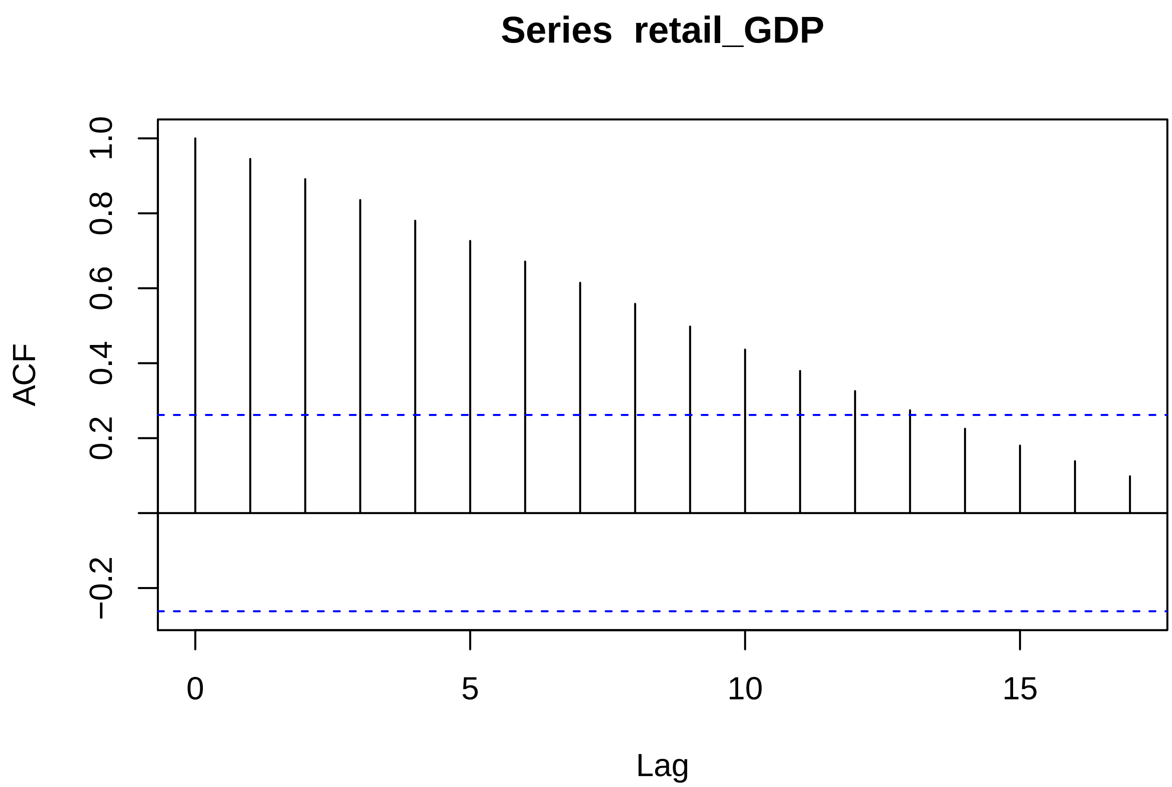
```
##
##  Augmented Dickey-Fuller Test
##
## data:  retail_GDP
## Dickey-Fuller = -1.0701, Lag order = 3, p-value = 0.9192
## alternative hypothesis: stationary
```

As we can see just from the plot of the original series above, there is evidence of a strong increasing trend. The Augmented Dickey-Fuller Test (ADF test) also shows a high p-value of 0.9192, which fails to reject the null hypothesis of the series being integrated at the 95% confidence level. In other words, we can conclude that the original series is most likely to be integrated and not stationary at the 95% confidence level. Below are the PACF and ACF plots to reinforce this conclusion:

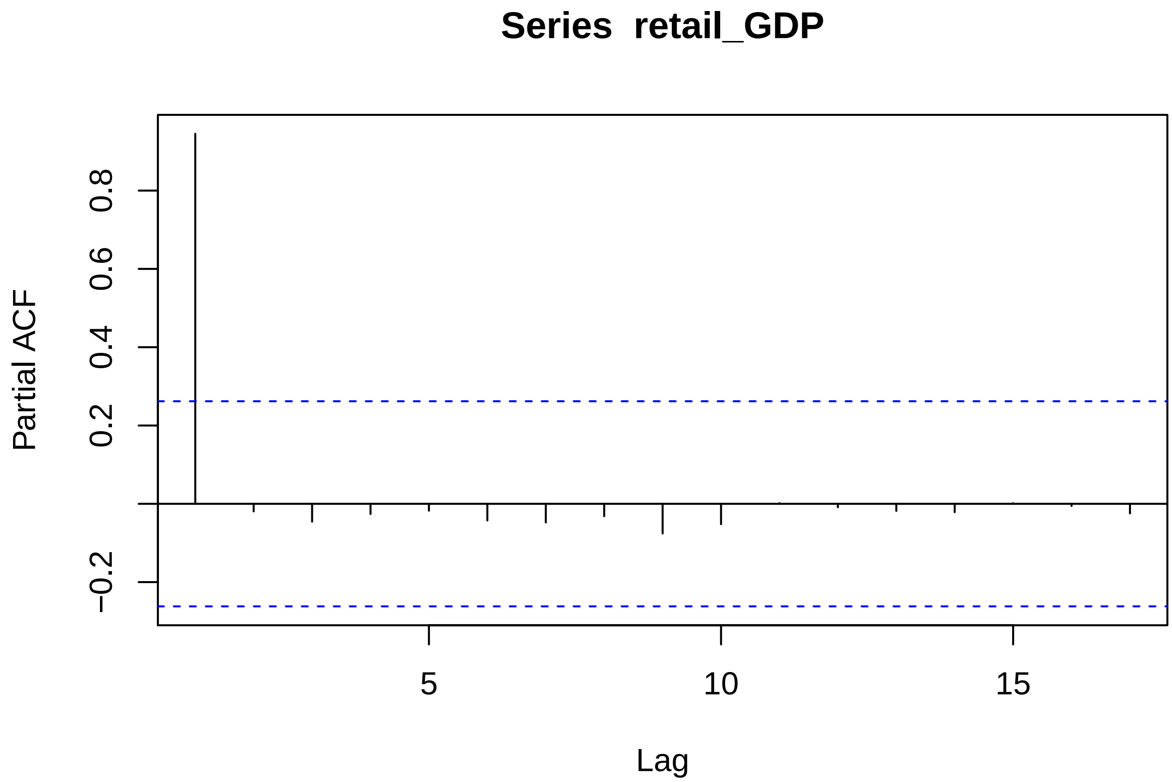
```r
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 3.6.3
```

```r
acf(retail_GDP)
```




```
pacf(retail_GDP)
```



We can see that the ACF plot tails off very slowly like that of a random walk process, indicating non-stationarity.

Fitting a VAR(1) model using `VARselect()`:

```
library(vars)
Y = cbind(retail_GDP, retail_real_GDP)
Y_intersect = ts.intersect(retail_GDP, retail_real_GDP) #combining two ts

VARselect(Y_intersect) #choose which order of VAR(p)
```

```
## $selection
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
##
## $criteria
##                   1            2            3            4            5
## AIC(n) 1.492097e+01 1.502429e+01 1.511890e+01 1.520904e+01 1.524201e+01
## HQ(n)  1.501032e+01 1.517321e+01 1.532738e+01 1.547710e+01 1.556963e+01
## SC(n)  1.515949e+01 1.542182e+01 1.567544e+01 1.592460e+01 1.611658e+01
## FPE(n) 3.021724e+06 3.355172e+06 3.699329e+06 4.070458e+06 4.243501e+06
##                   6            7            8            9           10
## AIC(n) 1.532086e+01 1.543103e+01 1.541833e+01 1.547651e+01 1.558634e+01
## HQ(n)  1.570804e+01 1.587779e+01 1.592465e+01 1.604239e+01 1.621179e+01
## SC(n)  1.635444e+01 1.662363e+01 1.676993e+01 1.698712e+01 1.725596e+01
## FPE(n) 4.650492e+06 5.285407e+06 5.345402e+06 5.846963e+06 6.796253e+06
```

```
VAR1_model = VAR(type = c("both"),Y_intersect, p=1)
#type = c("const", "trend", "both", "none")
VAR1_model
```

```
##
## VAR Estimation Results:
## =======================
##
## Estimated coefficients for equation retail_GDP:
## ===============================================
## Call:
## retail_GDP = retail_GDP.l1 + retail_real_GDP.l1 + const + trend
##
##      retail_GDP.l1 retail_real_GDP.l1              const              trend
##          0.8310143        192.0102401      -2565.4532857          5.0373739
##
##
## Estimated coefficients for equation retail_real_GDP:
## ====================================================
## Call:
## retail_real_GDP = retail_GDP.l1 + retail_real_GDP.l1 + const + trend
##
##      retail_GDP.l1 retail_real_GDP.l1              const              trend
##      -3.006489e-05       9.937737e-01       2.853467e-01       9.916676e-02
```

```
coeff = Bcoef(VAR1_model)
coeff #coefficients matrix of VAR1_model
```

```
##                 retail_GDP.l1 retail_real_GDP.l1         const      trend
## retail_GDP       8.310143e-01        192.0102401 -2565.4532857 5.03737393
## retail_real_GDP -3.006489e-05          0.9937737     0.2853467 0.09916676
```

```r
squared_matrix = coeff[1:2, 1:2] #removing constant and trend
squared_matrix
```

```
##                  retail_GDP.l1 retail_real_GDP.l1
## retail_GDP        8.310143e-01        192.0102401
## retail_real_GDP -3.006489e-05          0.9937737
```

```r
eigen = eigen(squared_matrix)
eigen_values = eigen$values
eigen_values
```

```
## [1] 0.941547 0.883241
```

```r
mod_eigen = Mod(eigen_values) #mod of eigen values
mod_eigen
```

```
## [1] 0.941547 0.883241
```

```r
#Using VARS:roots() function to check eigen values again
roots = roots(VAR1_model)
roots #eigen value all <= |1|
```

```
## [1] 0.941547 0.883241
```

Now, I have fitted a bivariate VAR(1) model on both `(nominal) GDP` and `Real GDP`, without any transformation on the series, and includes both a constant and trend term in this model. Based on the coefficient matrix and its corresponding eigen values, we can see that both eigenvalues (0.941547, 0.883241), are all less than 1, so this `VAR(1)` model is casual/stationary.

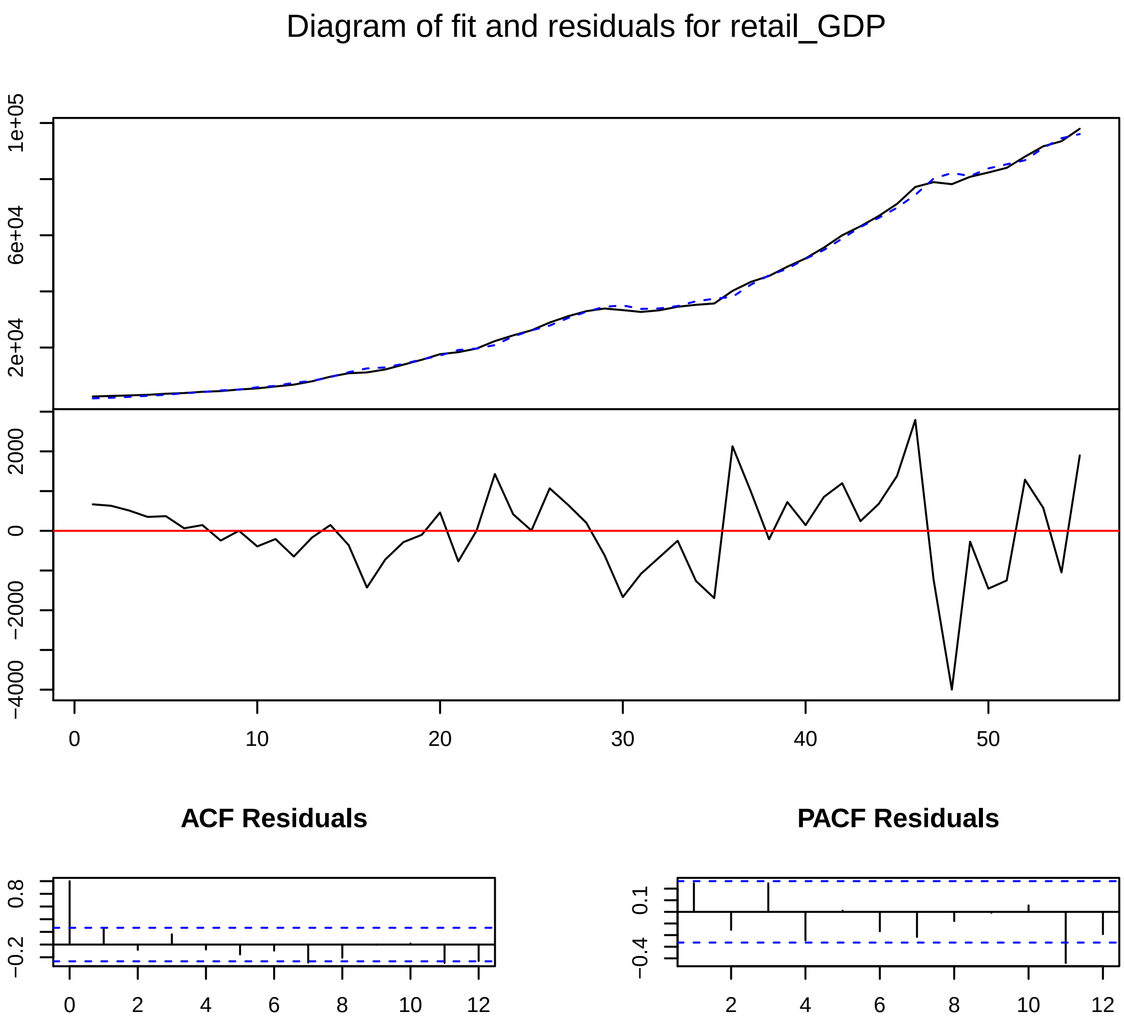
Mathematically, the $VAR(1)$ model fitted could be defined as follows:

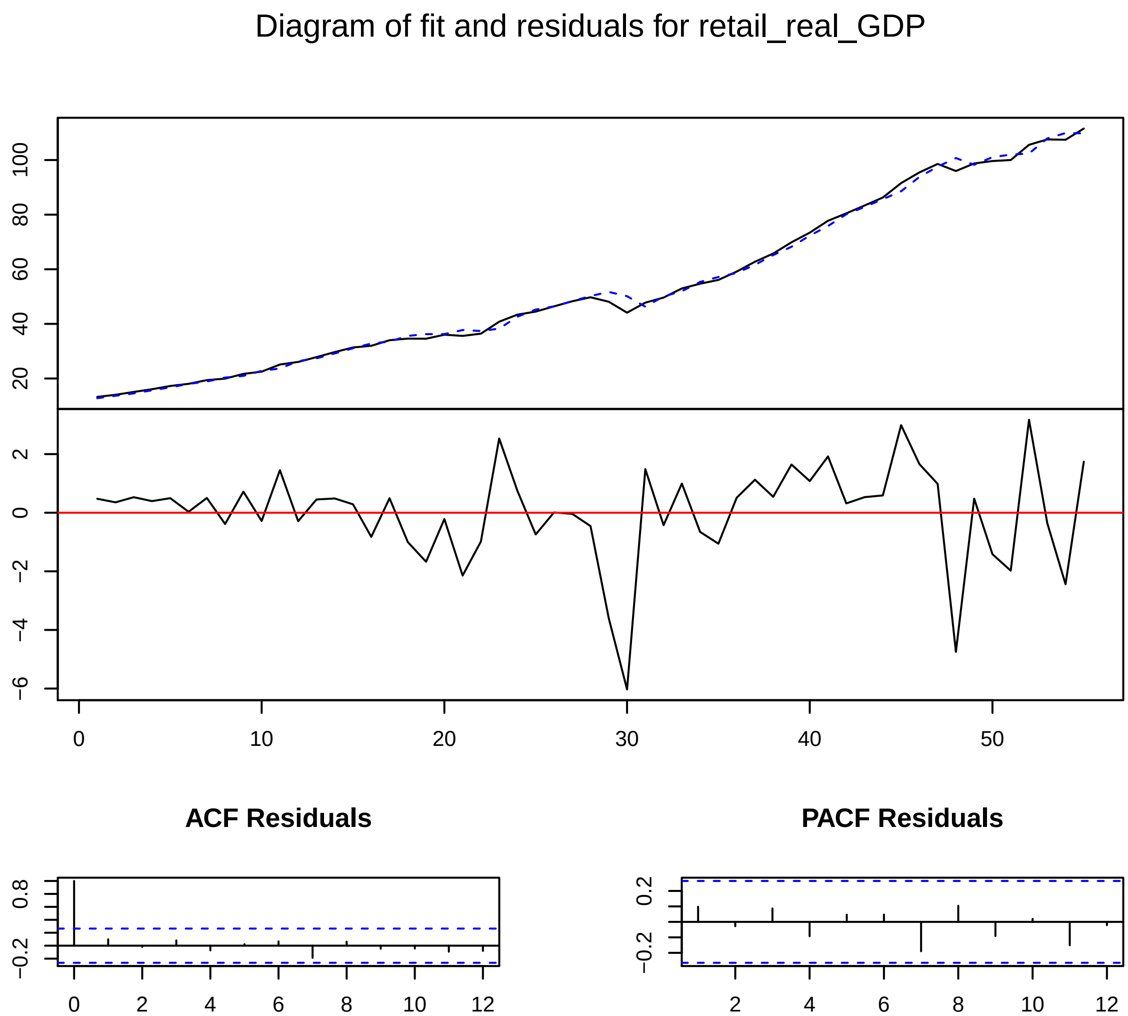
$$\begin{bmatrix} retail\_GDP_t \\ real\_GDP_t \end{bmatrix} = \begin{bmatrix} X_{1,t} \\ X_{2,t} \end{bmatrix} = \begin{bmatrix} 8.310143e-01 & 192.0102401 \\ -3.006489e-05 & 0.9937737 \end{bmatrix} \begin{bmatrix} X_{1,t-1} \\ X_{2,t-1} \end{bmatrix} + \begin{bmatrix} U_t \\ V_t \end{bmatrix}$$

, where U_t, V_t are `WNs`.

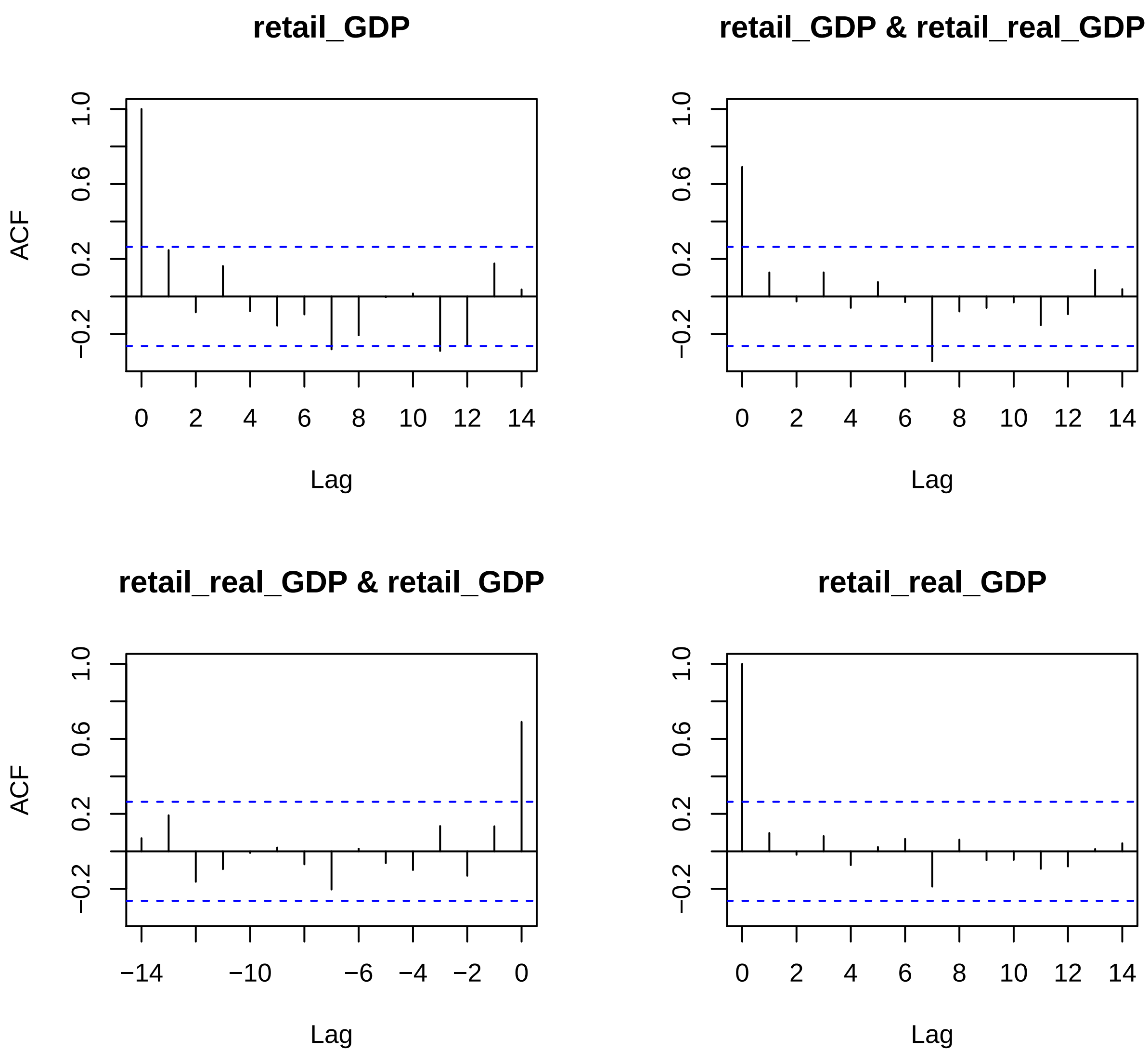
Plot of residuals and their ACF/CCF

```r
plot(VAR1_model)
```






```
#ACF/CCF plot of residuals(VAR1_model)
acf(resid(VAR1_model))
```




```
fitted_values = fitted(VAR1_model)
fitted_values_nominal= fitted_values[,1] #predicted/fitted values for nominal

library(Metrics)
length(retail_GDP) #56
```

```
## [1] 56
```

```
length(fitted_values_nominal) #55
```

```
## [1] 55
```

```
#residual MAPE => mape(actual, predicted)
#length differ, removed last value of original retail_GDP
mape(retail_GDP[1:55], fitted_values_nominal) #residual MAPE
```

```
## [1] 0.08149088
```

From the plot of both the (nominal) retail GDP and real GDP, we can see that the VAR(1) model has made pretty good predictions since the blue dashed line (fitted values/predictions) more or less overlap with the black lines (original observations). The residuals for both plot also has a mean centered at 0, and the variance of the residuals is also more or less constant.

The ACF/CCF plots of the residuals are well-behaved with White Noise-like behaviour (no strong auto-correlation after lag 0 in ACF plots of residuals), and there is only a significant spike in cross correlation at lag 0 as well, suggesting that the VAR(1) model is a good fit. There is also no evidence of partial auto correlations from the PACF plot of both GDP and real GDP.

Mathematically, we can define the model as follows since the series are simultaneously correlated White Noise Processes:

(nominal) GDP as Y_t and simultaneous Real GDP as X_t, where

$$X_t = W_t, Y_t = V_t$$
$$\mathrm{Cov}(W_t, V_t) = \mathbf{\Sigma}_t = \begin{bmatrix} \sigma_1 & \sigma_{1,2} \\ \sigma_{2,1} & \sigma_2 \end{bmatrix}$$

, where $\sigma_{1,2} = \sigma_{2,1} \neq 0$.

The `summary` table is as follows:

```r
summary(VAR1_model)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: retail_GDP, retail_real_GDP
## Deterministic variables: both
## Sample size: 55
## Log Likelihood: -550.043
## Roots of the characteristic polynomial:
## 0.9415 0.8832
## Call:
## VAR(y = Y_intersect, p = 1, type = c("both"))
##
##
## Estimation results for equation retail_GDP:
## ===========================================
## retail_GDP = retail_GDP.l1 + retail_real_GDP.l1 + const + trend
##
##                      Estimate Std. Error t value Pr(>|t|)
## retail_GDP.l1       8.310e-01  5.686e-02  14.615  < 2e-16 ***
## retail_real_GDP.l1  1.920e+02  6.718e+01   2.858  0.00616 **
## const              -2.565e+03  9.279e+02  -2.765  0.00791 **
## trend               5.037e+00  4.747e+01   0.106  0.91591
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1131 on 51 degrees of freedom
## Multiple R-Squared: 0.9986,  Adjusted R-squared: 0.9985
## F-statistic: 1.222e+04 on 3 and 51 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation retail_real_GDP:
## ================================================
```

```
## retail_real_GDP = retail_GDP.l1 + retail_real_GDP.l1 + const + trend
##
##                      Estimate Std. Error t value Pr(>|t|)
## retail_GDP.l1      -3.006e-05  8.555e-05  -0.351    0.727
## retail_real_GDP.l1  9.938e-01  1.011e-01   9.832 2.29e-13 ***
## const               2.853e-01  1.396e+00   0.204    0.839
## trend               9.917e-02  7.143e-02   1.388    0.171
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.702 on 51 degrees of freedom
## Multiple R-Squared: 0.997,   Adjusted R-squared: 0.9968
## F-statistic:  5592 on 3 and 51 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                 retail_GDP retail_real_GDP
## retail_GDP         1279457        1329.785
## retail_real_GDP       1330           2.896
##
## Correlation matrix of residuals:
##                 retail_GDP retail_real_GDP
## retail_GDP          1.0000          0.6908
## retail_real_GDP     0.6908          1.0000
```

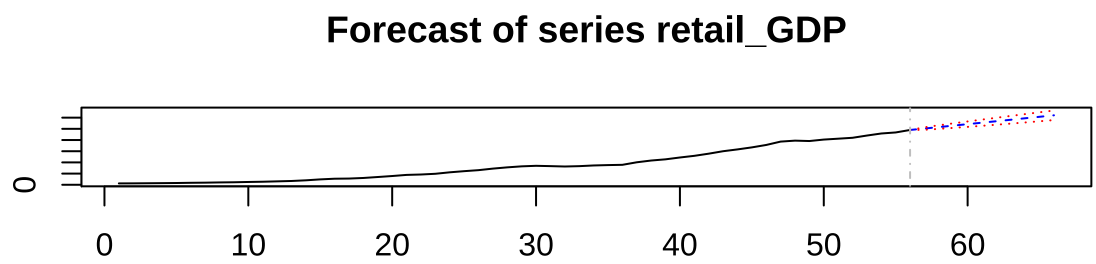
10-year-ahead predictions for both series

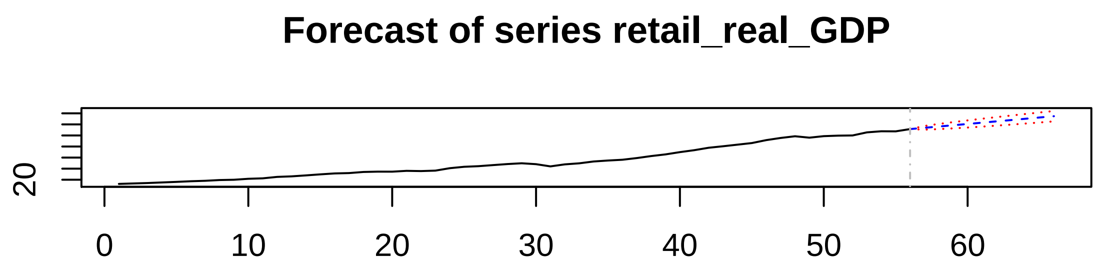
```
predict(VAR1_model,n.ahead=10, plot=T)
```

```
## $retail_GDP
##           fcst     lower    upper       CI
##  [1,] 100532.0  98315.02 102749.0 2216.976
##  [2,] 103124.8  99907.70 106341.9 3217.077
##  [3,] 105727.2 101668.65 109785.8 4058.552
##  [4,] 108339.0 103511.67 113166.2 4827.286
##  [5,] 110959.7 105412.29 116507.1 5547.427
##  [6,] 113589.2 107362.73 119815.6 6226.434
##  [7,] 116226.9 109360.56 123093.3 6866.387
##  [8,] 118872.7 111404.99 126340.5 7467.747
##  [9,] 121526.2 113495.48 129556.9 8030.713
## [10,] 124187.0 115631.30 132742.7 8555.703
##
## $retail_real_GDP
##           fcst    lower    upper       CI
##  [1,] 113.8197 110.4841 117.1552 3.335571
##  [2,] 116.1255 111.4551 120.7960 4.670454
##  [3,] 118.4383 112.7812 124.0953 5.657088
##  [4,] 120.7575 114.3035 127.2115 6.454003
##  [5,] 123.0830 115.9595 130.2064 7.123410
##  [6,] 125.4143 117.7164 133.1122 7.697897
##  [7,] 127.7512 119.5538 135.9487 8.197405
```

```
##  [8,] 130.0935 121.4581 138.7289 8.635421
##  [9,] 132.4408 123.4191 141.4625 9.021729
## [10,] 134.7929 125.4291 144.1567 9.363800
```

```r
plot(predict(VAR1_model,n.ahead=10, plot=T))
```



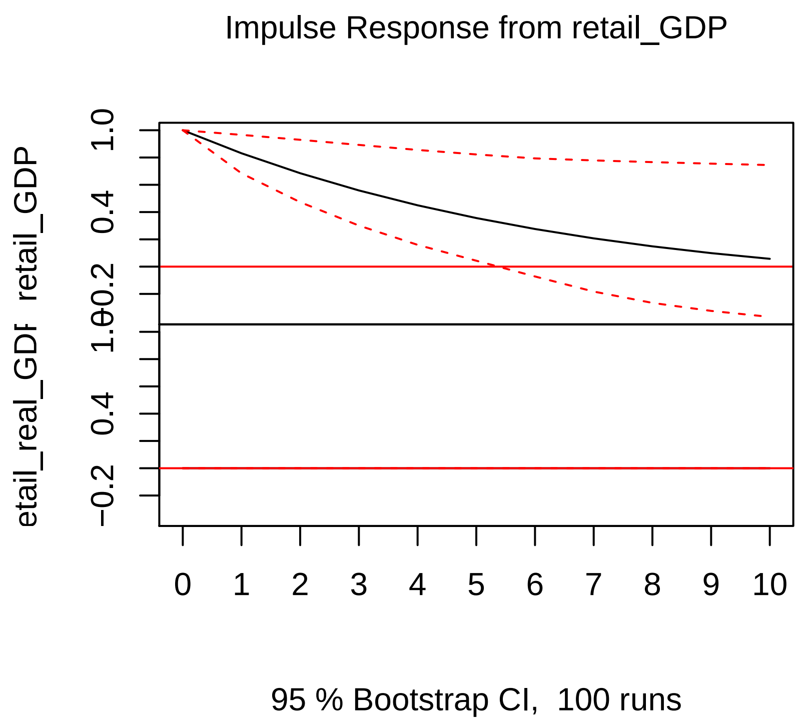


Granger-Casuality Tests

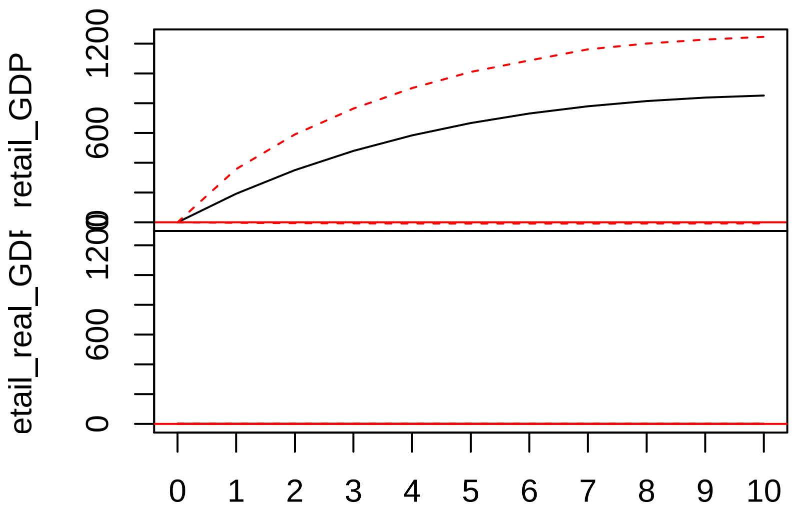
```r
causality(VAR1_model, cause='retail_real_GDP')
```

```
## $Granger
##
##  Granger causality H0: retail_real_GDP do not Granger-cause retail_GDP
##
## data:  VAR object VAR1_model
## F-Test = 8.1691, df1 = 1, df2 = 102, p-value = 0.005166
##
##
## $Instant
##
##  H0: No instantaneous causality between: retail_real_GDP and retail_GDP
##
## data:  VAR object VAR1_model
## Chi-squared = 17.767, df = 1, p-value = 2.497e-05
```

```
irf(VAR1_model, ortho=FALSE) %>% plot()
```



Impulse Response from retail_real_GDP



95 % Bootstrap CI, 100 runs

Rather than using only a bivariate/multivariate model to predict `retail GDP` (we used `real GDP` in this case), there might be better models that could account for the relationships between other (economic variables) and `retail GDP`. A simple Granger-Casuality hypothesis test is performed, and we realized that we could reject the null hypothesis under 90% confidence level that `real GDP` could have Granger-cause (nominal) `retail GDP`. In other words, using other (economic) variables and their corresponding time series data (as external regressors) could help us make better predictions than using only the past values of `retail GDP` alone. Although it is hard to see such pattern from the impulse response plots (i.e., it is hard to observe an strong casuality-like effect for `real GDP` on `retail GDP`), the hypothesis that using other external economic regressors to help better predict `retail GDP` is still valid. Hence, I have decided to try an ARMA-error regression model with various external regressor as follows.

Fitting an ARMA-error regression model for (nominal) GDP (Y_t) with simultaneous Real GDP (X_t) as the external regressor:

```
#since auto.arima()'s xreg() requires same length,
#we will only get the real GDP up until 2016
retail_real_GDP_2016 = get_cansim_vector( "v41712939", start_time = "1961-01-01",
                                          end_time = "2016-12-01") %>% pull(VALUE) %>%
  ts( start = 1961, end = 2016)
#start 1961, ends in 2016

ARMA_error_model = auto.arima(retail_GDP, xreg=retail_real_GDP_2016)
ARMA_error_model
```

```
## Series: retail_GDP
## Regression with ARIMA(4,0,0) errors
##
## Coefficients:
##          ar1     ar2     ar3      ar4  intercept      xreg
##       1.5142  -0.408  0.1421  -0.2523   27260.77  379.1507
## s.e.  0.1438   0.292  0.2930   0.1468   23246.10   89.4924
##
## sigma^2 estimated as 998570:  log likelihood=-467.19
## AIC=948.38   AICc=950.71   BIC=962.55
```

```
fitted_values_ARMA = fitted(ARMA_error_model) #predicted/fitted values for nominal
mape(retail_GDP, fitted_values_ARMA) #residual MAPE for ARMA errors
```

```
## [1] 0.02894104
```

We can see that the `auto.arima()` function returns a ARIMA(4,0,0) with AIC=948.38 and AICc=950.71. The MAPE for this model is 0.02894104.

Fitting an ARMA-error regression model with other variables:

The different external regressors/variables I have decided to fit the ARMA-error regression model for retail trade (nominal) GDP is as follows:

```
multifactor_productivity = get_cansim_vector( "v41712888", start_time = "1961-01-01",
                                    end_time = "2016-12-01") %>% pull(VALUE) %>%
  ts( start = 1961, end = 2016)

labour_productivity = get_cansim_vector( "v41712905", start_time = "1961-01-01",
                                    end_time = "2016-12-01") %>% pull(VALUE) %>%
  ts( start = 1961, end = 2016)

capital_productivity = get_cansim_vector( "v41712922", start_time = "1961-01-01",
                                    end_time = "2016-12-01") %>% pull(VALUE) %>%
  ts( start = 1961, end = 2016)
```

The ARMA-error regression model for each corresponding external regressors for fitting retail trade (nominal) GDP is as follows:

Using Labour productivity as external regressor:

```
ARMA_error_model_labour_productivity = auto.arima(retail_GDP, xreg=labour_productivity)
ARMA_error_model_labour_productivity
```

```
## Series: retail_GDP
## Regression with ARIMA(0,0,5) errors
##
## Coefficients:
##          ma1     ma2     ma3     ma4     ma5   intercept       xreg
##       1.3184  1.3215  1.0465  0.6605  0.2922  -39778.566  1148.1535
## s.e.  0.1507  0.2273  0.2495  0.2248  0.1302    5628.804    80.1389
##
## sigma^2 estimated as 5569354:  log likelihood=-511.82
## AIC=1039.64   AICc=1042.7   BIC=1055.84
```

```
fitted_labour_productivity = fitted(ARMA_error_model_labour_productivity)
#predicted/fitted values for nominal
mape(retail_GDP, fitted_labour_productivity) #residual MAPE for ARMA errors
```

```
## [1] 0.1265858
```

Using Capital productivity as external regressor:

```
ARMA_error_model_capital_productivity = auto.arima(retail_GDP, xreg=capital_productivity)
ARMA_error_model_capital_productivity
```

```
## Series: retail_GDP
## Regression with ARIMA(0,1,4) errors
##
## Coefficients:
##          ma1     ma2     ma3     ma4      drift     xreg
##       0.6889  0.0786  0.5173  0.4741  1749.5207   1.8399
## s.e.  0.1307  0.1302  0.1549  0.1604   382.2251  25.9872
##
```

```
## sigma^2 estimated as 1248875:  log likelihood=-461.8
## AIC=937.59   AICc=939.98   BIC=951.64
```

```
fitted_capital_productivity = fitted(ARMA_error_model_capital_productivity)
#predicted/fitted values for nominal
mape(retail_GDP, fitted_capital_productivity) #residual MAPE for ARMA errors
```

```
## [1] 0.04965696
```

Using Multifactor productivity as external regressor:

```
ARMA_error_model_multifactor_productivity = auto.arima(retail_GDP, xreg=multifactor_productivity)
ARMA_error_model_multifactor_productivity
```

```
## Series: retail_GDP
## Regression with ARIMA(0,2,2) errors
##
## Coefficients:
##           ma1      ma2     xreg
##       -0.1714  -0.5653  84.1019
## s.e.   0.1406   0.1415  53.4711
##
## sigma^2 estimated as 1251574:  log likelihood=-454.63
## AIC=917.25   AICc=918.07   BIC=925.21
```

```
fitted_multifactor_productivity = fitted(ARMA_error_model_multifactor_productivity)
#predicted/fitted values for nominal
mape(retail_GDP, fitted_multifactor_productivity) #residual MAPE for ARMA errors
```

```
## [1] 0.02412925
```

```
summary(ARMA_error_model_multifactor_productivity)
```

```
## Series: retail_GDP
## Regression with ARIMA(0,2,2) errors
##
## Coefficients:
##           ma1      ma2     xreg
##       -0.1714  -0.5653  84.1019
## s.e.   0.1406   0.1415  53.4711
##
## sigma^2 estimated as 1251574:  log likelihood=-454.63
## AIC=917.25   AICc=918.07   BIC=925.21
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE       ACF1
## Training set 207.2331 1067.626 708.3383 1.209149 2.412925 0.3920515 -0.1239115
```

We can see that `multifactor productivity` has the smallest AIC/AICc values out of the 3 other external regressors (including variable `real GDP`) with AIC=917.25 and AICc=918.07. It also has the smallest MAPE of 0.02412925 or 2.41% . Hence, we will choose this regressor for further analysis and diagnositics.

```
library(astsa)
```

```
## Warning: package 'astsa' was built under R version 3.6.3
```

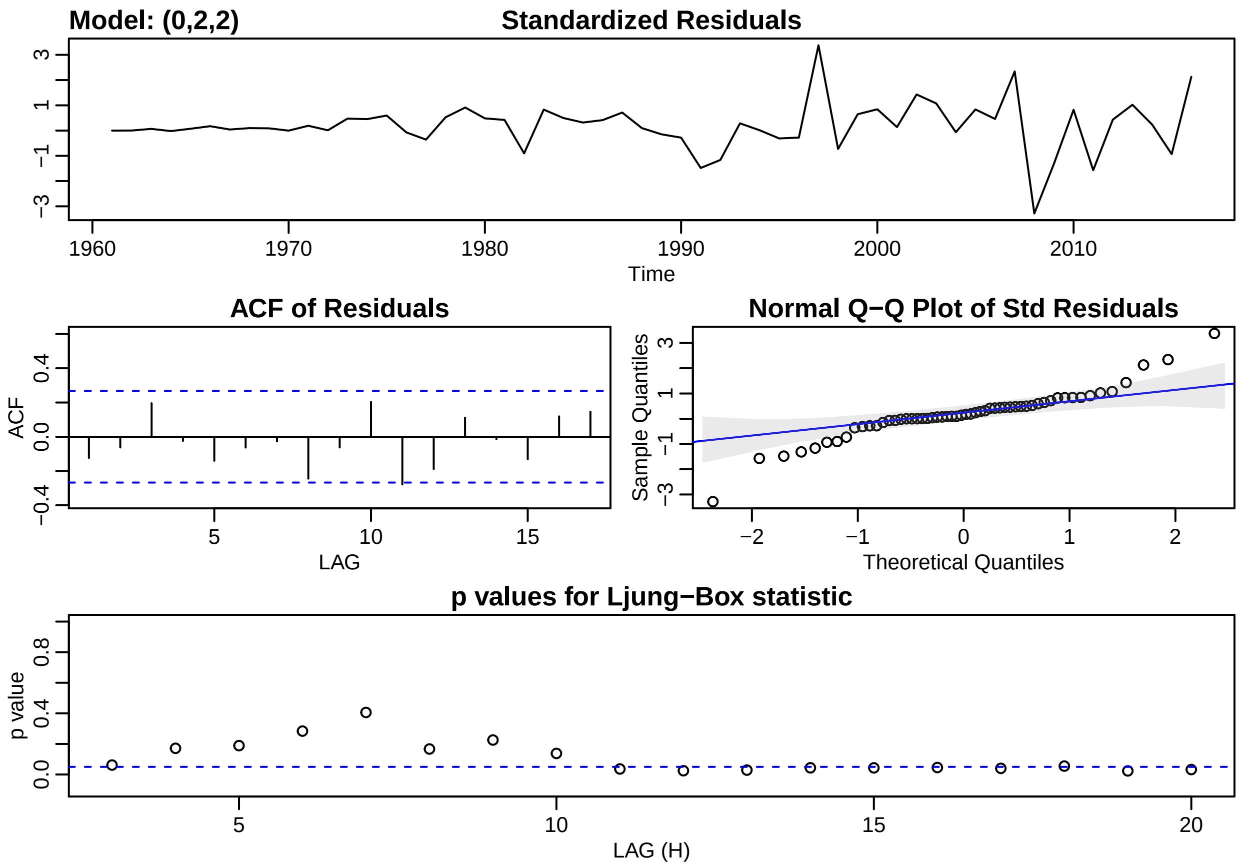
```
##
## Attaching package: 'astsa'
```

```
## The following object is masked from 'package:forecast':
##
##     gas
```

```r
best_model = arima(retail_GDP, xreg =multifactor_productivity , order = c(0,2,2))

#diagonistics
sarima(retail_GDP, xreg =multifactor_productivity ,0,2,2)
```

```
## initial  value 7.143537
## iter   2 value 7.007988
## iter   3 value 6.999962
## iter   4 value 6.996772
## iter   5 value 6.993952
## iter   6 value 6.992126
## iter   7 value 6.992024
## iter   8 value 6.992007
## iter   9 value 6.992006
## iter  10 value 6.992005
## iter  10 value 6.992005
## iter  10 value 6.992005
## final  value 6.992005
## converged
## initial  value 7.000309
## iter   2 value 7.000283
## iter   3 value 7.000079
## iter   4 value 7.000075
## iter   5 value 7.000072
## iter   6 value 7.000071
## iter   6 value 7.000071
## iter   6 value 7.000071
## final  value 7.000071
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.control = list(trace =
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1      ma2     xreg
##       -0.1714  -0.5653  84.1019
## s.e.   0.1406   0.1415  53.4711
##
## sigma^2 estimated as 1182042:  log likelihood = -454.63,  aic = 917.25
##
## $degrees_of_freedom
## [1] 51
##
## $ttable
##      Estimate      SE t.value p.value
## ma1   -0.1714  0.1406 -1.2194  0.2283
## ma2   -0.5653  0.1415 -3.9942  0.0002
## xreg  84.1019 53.4711  1.5728  0.1219
##
## $AIC
## [1] 16.98617
##
```

```
## $AICc
## [1] 16.99506
##
## $BIC
## [1] 17.1335
```

```r
Box.test(best_model$resid, lag = 24, type = c("Ljung-Box"), fitdf = 8)$p.value
```

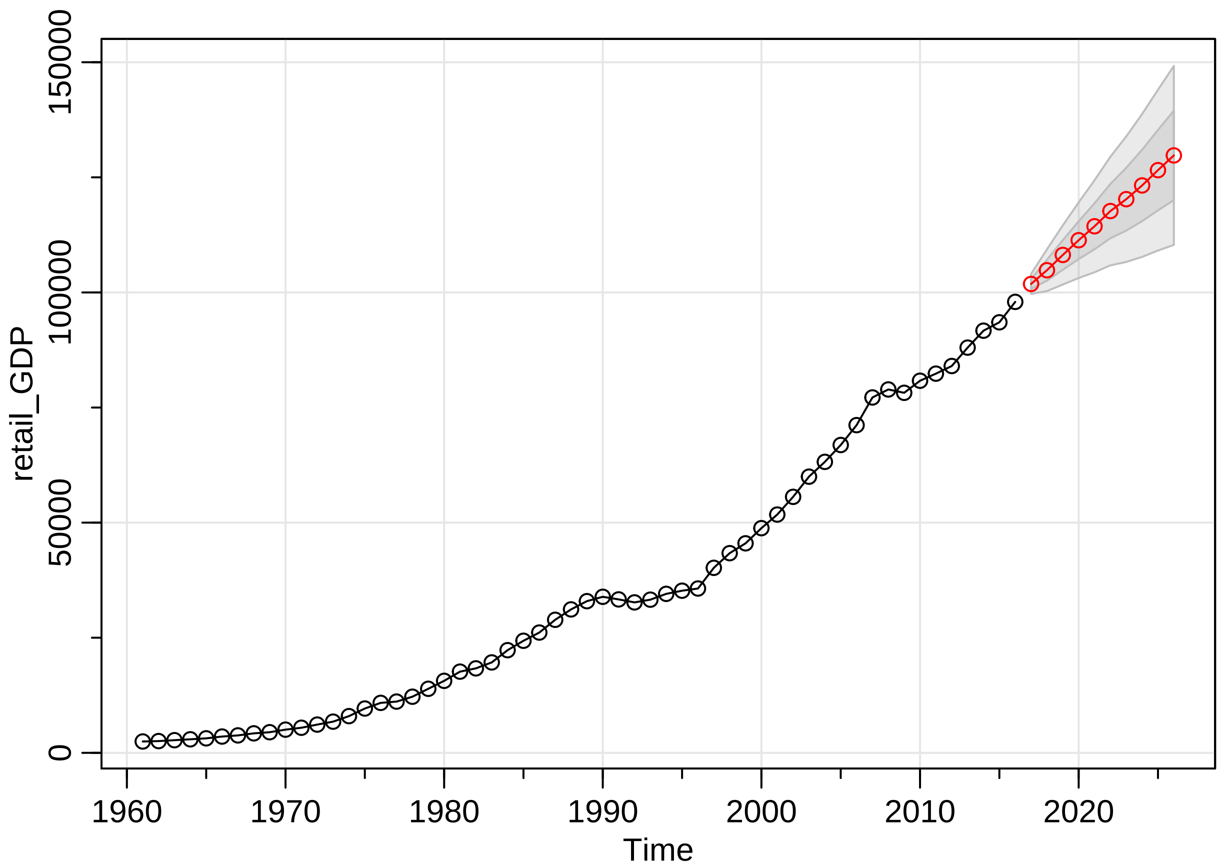
```
## [1] 0.01320815
```

First, we can see from the standardized residual plots that the residuals have a constant mean at around 0 and also has a more or less constant variance, suggesting stationarity. The ACF plot of the residuals are all within the 95% confidence intervals,indicating that there is no correlation between the residuals (suggesting a good fit of the model). The Normal Q-Q plot suggests that there are a few extreme outliers (on both end of the tails), making the normality of the residuals to be slightly violated (but the bulk of the residuals are still following a Normal distribution). This indicates that perhaps a transformation like the natural log-transformation could be applied to our time series. Most p-values of the Ljung-Box test are above the 5% blue dashed line, indicating that the model has no serial correlation with 95% confidence level (but there are some p-values right on the line itself). Hence, I have decided to use a Ljung-Box test to obtain the final p-value of 0.01320815. This suggests that we failed to reject the null-hypothesis that the data (residuals) are independently distributed, i.e., we have enough evidence to conclude that there is no serial correlations (of the residuals) for this model at the 95% confidence level. Hence the ARMA-error regression model for retail trade (nominal) GDP with `multifactor productivity` as its external regressor is the best model we have.

10-year-ahead predictions for retail_GDP series using multifactor productivity as external regressor

```r
sarima.for(retail_GDP, xreg =multifactor_productivity ,p=0,d=2,q=2, newxreg = tail(multifactor_producti
```

```
## Warning in z[[1L]] + xm: longer object length is not a multiple of shorter
## object length
```




```
## $pred
## Time Series:
## Start = 2017
## End = 2026
## Frequency = 1
##  [1] 101838.9 104820.1 108139.3 111343.5 114369.3 117675.9 120257.3 123238.5
##  [9] 126557.7 129761.9
##
## $se
## Time Series:
## Start = 2017
## End = 2026
## Frequency = 1
##  [1] 1087.217 2265.947 3210.481 4106.576 4996.873 5897.915 6817.468 7759.512
##  [9] 8726.155 9718.489
```