STA2202 - Time Series Analysis - Assignment 3 - PRACTICE

Luis Correia - Student No. 1006508566

June 09th 2020

Submission instructions:

Submit *two separate files* to A3 on Quercus - the deadline is 11:59PM on Monday, June 15.
- A PDF file with your Theory part answers.
- A PDF file with your Practice part report (w/ code in R Markdown chunks or in Appendix).

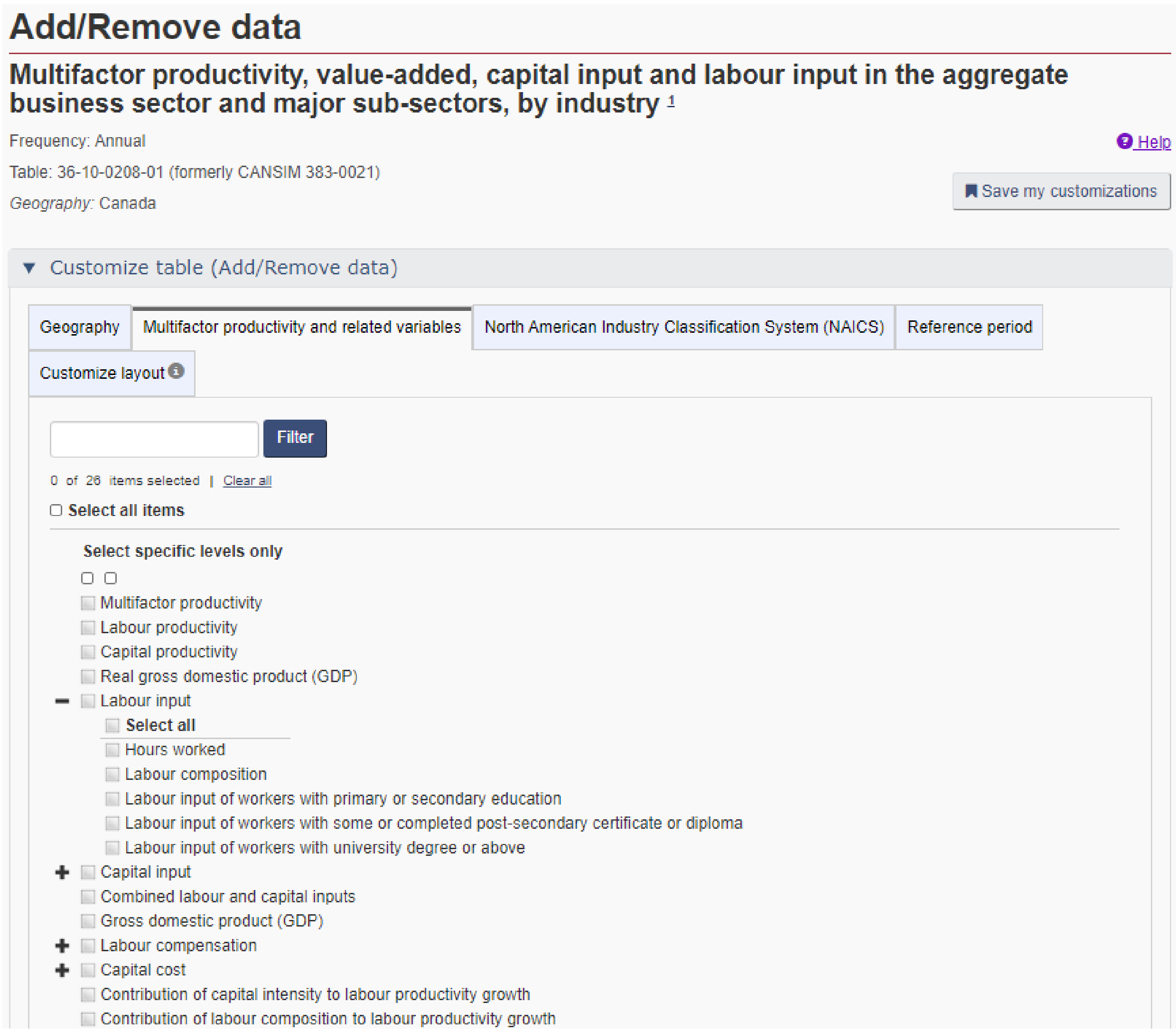
Practice

For this part you will work on this year'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; this problem is referred to as *nowcasting*. Each student will work on one of 20 different industry/sector groups as follows:

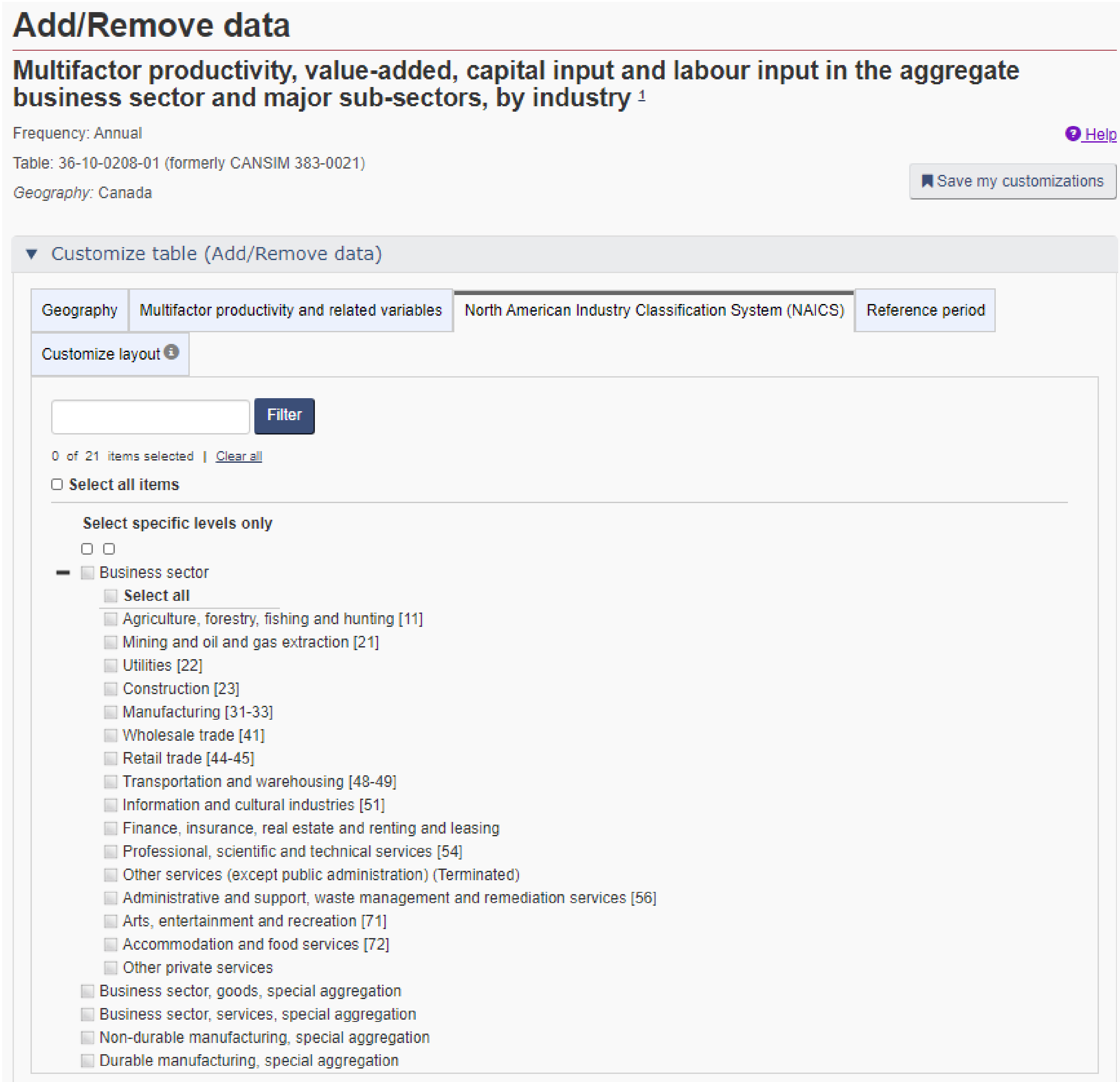
Sector/Industry Group	Last 2 digits of student #
Agriculture, forestry, fishing and hunting	00-04
Mining and oil and gas extraction	05-09
Utilities	10-14
Construction	15-19
Manufacturing	20-24
Wholesale trade	25-29
Retail trade	30-34
Transportation and warehousing	35-39
Information and cultural industries	40-44
Finance, insurance, real estate and renting and leasing	45-49
Professional, scientific and technical services	50-54
Other services (except public administration) (Terminated)	55-59
Administrative and support, waste management and remediation services	60-64
Arts, entertainment and recreation	65-69
Accommodation and food services	70-74
Other private services	75-79
Business sector, goods, special aggregation	80-84
Business sector, services, special aggregation	85-89
Non-durable manufacturing, special aggregation	90-94
Durable manufacturing, special aggregation	95-99

Data

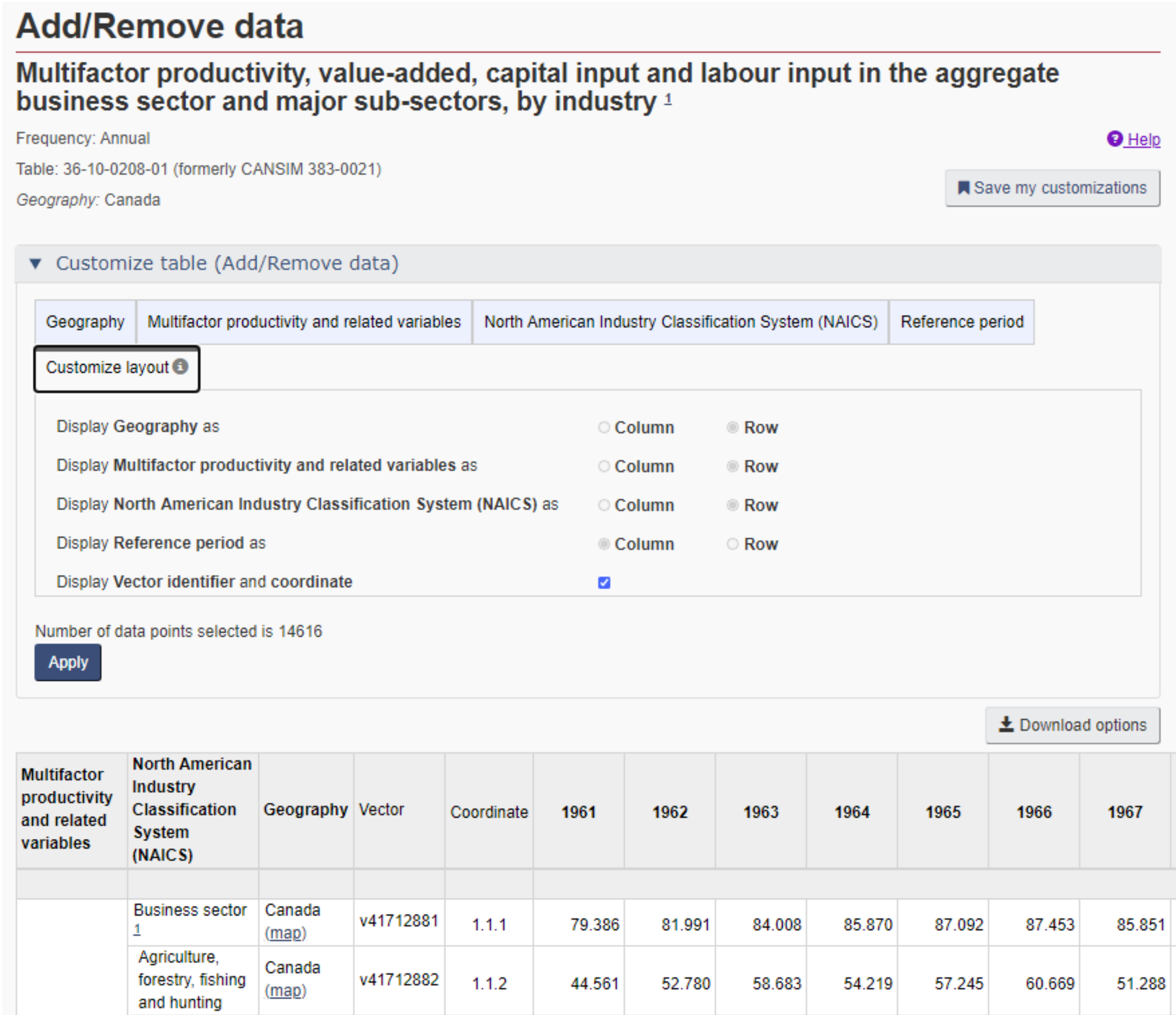
The data are given in StatCan Table: 36-10-0208-01 called *"Multifactor productivity, value-added, capital input and labour input in the aggregate business sector and major sub-sectors, by industry"*. This table contains annual data from 1961-2018 for a range of economic variables listed under the *Add/Remove data* option , as shown below:



You only need to work with data from your own Business Sector/Industry group, selected under the *North American Industry Classification System (NAICS)* tab:



You will notice that the range of values for the variable *Gross Domestic Product (GDP)* is two years **shorter** (ends in 2016) than the other variables (end in 2018). You can extract table data using R's `cansim` library, as in Assignment 1; you can find each series' *vector* identifier using the *Customize Layout* tab:



Add/Remove data

Multifactor productivity, value-added, capital input and labour input in the aggregate business sector and major sub-sectors, by industry [1]

Frequency: Annual

Table: 36-10-0208-01 (formerly CANSIM 383-0021)

Geography: Canada

Help

Save my customizations

Customize table (Add/Remove data)

Geography | Multifactor productivity and related variables | North American Industry Classification System (NAICS) | Reference period

Customize layout

Display **Geography** as	Column	Row
Display **Multifactor productivity and related variables** as	Column	Row
Display **North American Industry Classification System (NAICS)** as	Column	Row
Display **Reference period** as	Column	Row
Display **Vector identifier** and **coordinate**		

Number of data points selected is 14616

Apply

Download options

Multifactor productivity and related variables	North American Industry Classification System (NAICS)	Geography	Vector	Coordinate	1961	1962	1963	1964	1965	1966	1967
	Business sector 1	Canada (map)	v41712881	1.1.1	79.386	81.991	84.008	85.870	87.092	87.453	85.851
	Agriculture, forestry, fishing and hunting	Canada (map)	v41712882	1.1.2	44.561	52.780	58.683	54.219	57.245	60.669	51.288

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

```r
# Vector - v62458851 - Real gross domestic product/Arts, entertainment and recreation
# Real gross domestic product (GDP) (or real value-added) is a chained Fisher
# quantity index of gross domestic product (GDP) at basic prices.

X = get_cansim_vector( "v62458851", start_time = "1961-01-01", end_time = "2018-12-31") %>%
  pull(VALUE) %>% ts( start = 1961, 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.
```

```
# Vector - v62458803 - Gross Domestic Product/Arts, entertainment and recreation
# Gross domestic product (GDP) is valued at basic prices. It is calculated as gross output
# at basic prices minus intermediate inputs at purchaser prices. Data on gross domestic
# product (GDP) are available up to the most current year of the input-output table.

Y = get_cansim_vector( "v62458803", start_time = "1961-01-01", end_time = "2018-12-31") %>%
  pull(VALUE) %>% ts( start = 1961, frequency = 1)
```

1. [2 marks] 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).

{*Solution.*}

```
autoplot(Y, ylab="GDP (in million dollars)")
```

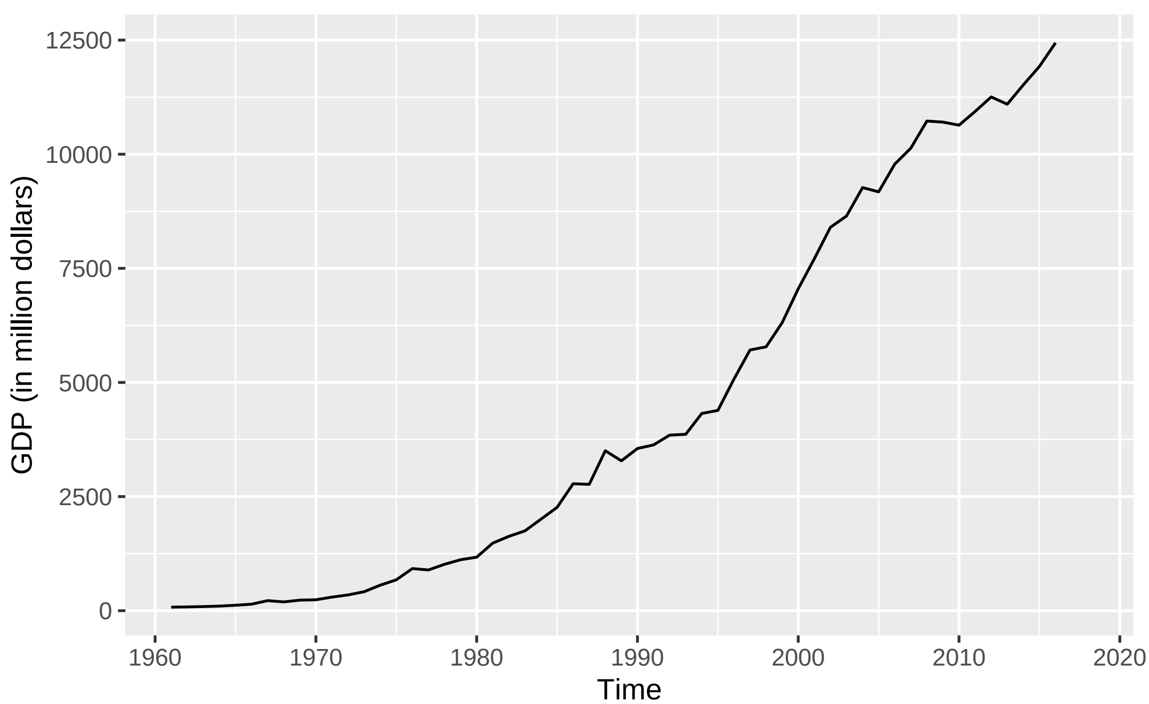


Figure 1: Nominal GDP

```
adf.test(diff(head(Y,-2)))
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(head(Y, -2))
## Dickey-Fuller = -2.4154, Lag order = 3, p-value = 0.4072
## alternative hypothesis: stationary
```

```
adf.test(head(Y,-2))
```

```
##
```

```
##  Augmented Dickey-Fuller Test
##
## data:  head(Y, -2)
## Dickey-Fuller = -1.903, Lag order = 3, p-value = 0.6137
## alternative hypothesis: stationary
```

Performing the `adf.test()` we obtain *p-value*= 0.4072 which results in the acceptance of $H_o : \phi = 1$, i.e., the GDP may be represented as a *Random-Walk* and, therefore, is **integrated**.

2. [3 marks] 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 inside the unit disk (use functions `eigen` and `Mod`).

{*Solution.*}

```r
# Include both variables in Matrix to fit VAR Model adjusting 'NA's for 2017 and 2018
YAdj <- head(Y,-2)
XAdj <- head(X,-2)
n <- length(YAdj)
M <- cbind(YAdj, XAdj)

# Fit VAR Model
fit1 <- VAR(M, p = 1, type = "both")

SY <- summary(fit1, equations = "YAdj")
SX <- summary(fit1, equations = "XAdj")

kableExtra::kable(SY$varresult$YAdj$coefficients, "latex",
                  booktabs = TRUE, caption = "VAR result for YAdj")
```

Table 2: VAR result for YAdj

	Estimate	Std. Error	t value	Pr(>|t|)
YAdj.l1	0.9484547	0.0274505	34.551493	0.0000000
XAdj.l1	-0.1116705	5.5516022	-0.020115	0.9840301
const	-130.1863159	100.4717476	-1.295751	0.2008954
trend	20.0978276	13.5389645	1.484444	0.1438461

```r
kableExtra::kable(SX$varresult$XAdj$coefficients, "latex",
                  booktabs = TRUE, caption = "VAR result for XAdj")
```

Table 3: VAR result for XAdj

	Estimate	Std. Error	t value	Pr(>|t|)
YAdj.l1	-0.0007877	0.0003988	-1.9751459	0.0536761
XAdj.l1	0.8191525	0.0806599	10.1556399	0.0000000
const	1.0871319	1.4597654	0.7447306	0.4598527
trend	0.5223699	0.1967092	2.6555444	0.0105394

Which leads to the following Coefficient Matrix:

```r
# Extracting the coefficient matrix
CF <- Bcoef(fit1)
kableExtra::kable(CF[1:2,1:2], "latex", caption = "Coefficient Matrix")
```

Table 4: Coefficient Matrix

	YAdj.l1	XAdj.l1
YAdj	0.9484547	-0.1116705
XAdj	-0.0007877	0.8191525

```r
# Calculating the eigenvalues of the coefficient matrix
kableExtra::kable(eigen(CF[1:2,1:2])$values, "latex", col.names = "Eigen-Values",
                  booktabs = TRUE, caption = "Eigenvalues of Coefficient Matrix")
```

Table 5: Eigenvalues of Coefficient Matrix

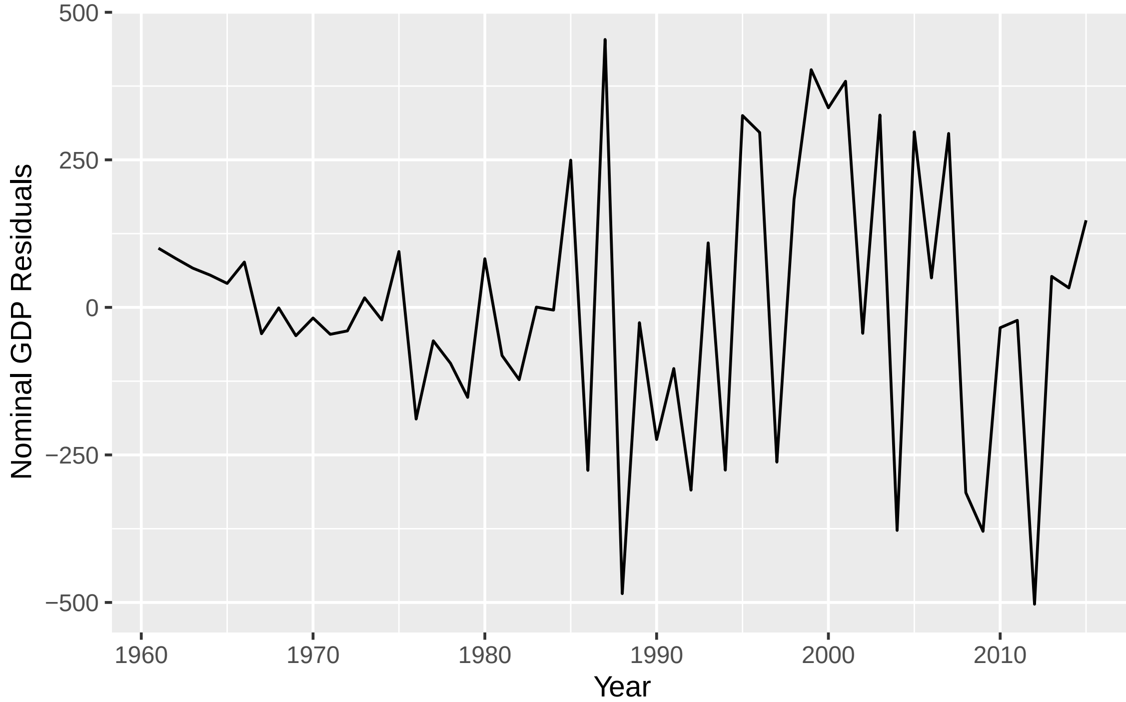
Eigen-Values
0.9491315
0.8184757

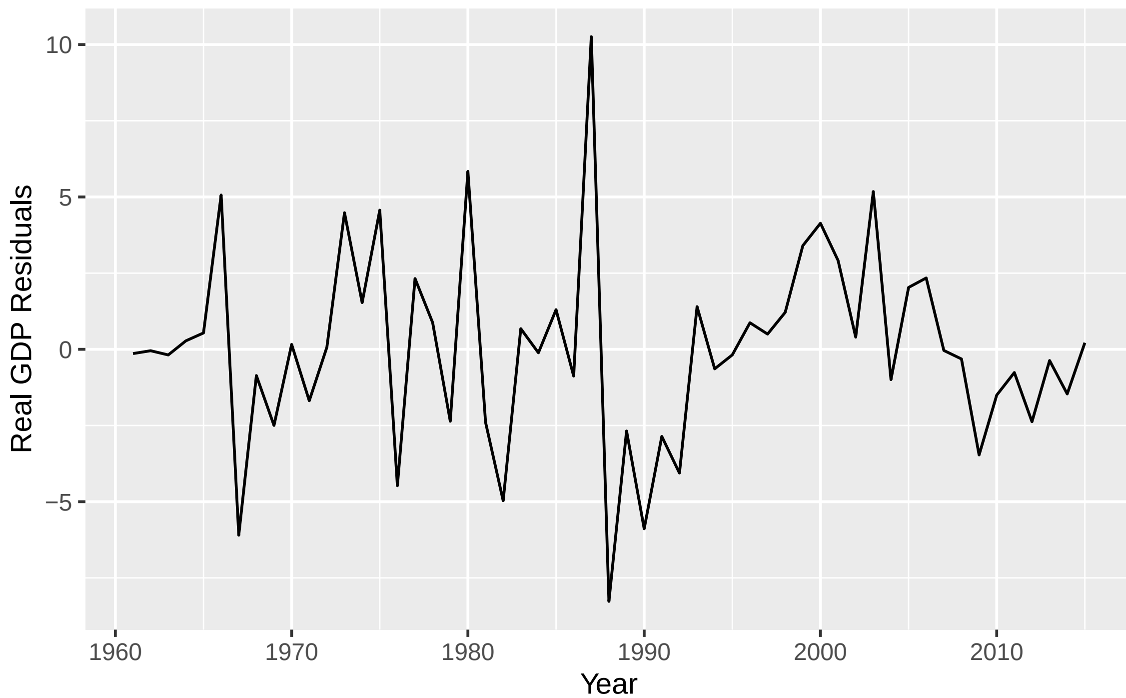
As all eigen-values are *in modulus* inside the unit disk, therefore the model is **stationary**.

3. [2 marks] 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.

{*Solution.*}

The residuals for the adjusted model can be visualized below, as well its ACF/CCF.






```r
# ACF Plot
ggAcf(Resid, lag.max = 24)
```

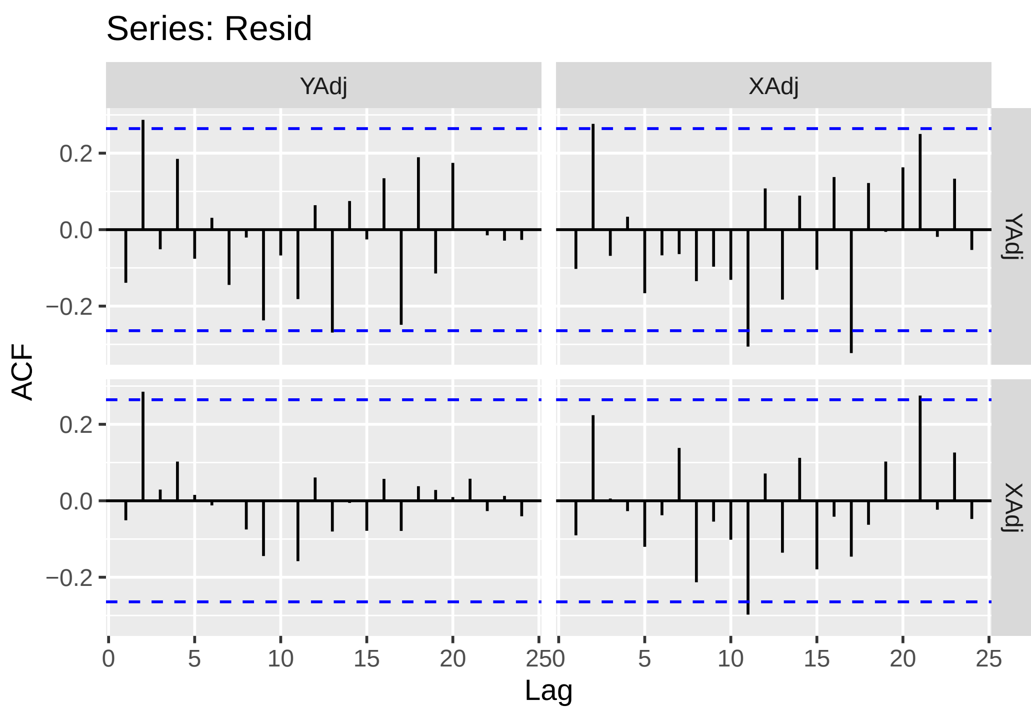


Figure 2: ACF - Nominal & Real GDP

```r
# CCF Plot
ggCcf(Resid[,"YAdj"],Resid[,"XAdj"], type = "correlation")
```

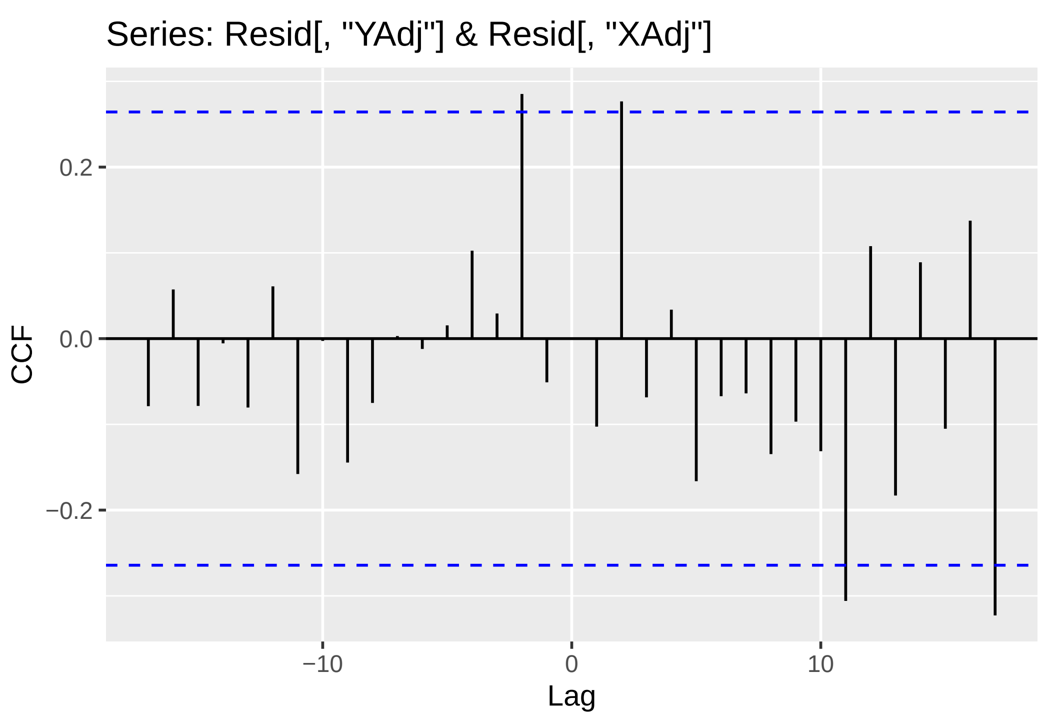


Figure 3: CCF - Nominal & Real GDP

Residuals plots shows ACF with no significant correlation of both variables which suggests a good model fit for our data. When checking the CCF output we observe similar output, showing that there is no significant cross-correlation between residuals of both variables used in our model.

```
# MAPE Function
MAPE <- function (At, Ft) {
  return(1/length(At)*sum(abs((At-Ft)/At)))
}
MAPE_Y <- MAPE(fit1$datamat$YAdj,fit1$varresult$YAdj$fitted.values)
MAPE_X <- MAPE(fit1$datamat$XAdj,fit1$varresult$XAdj$fitted.values)

kableExtra::kable(MAPE_Y, "latex", col.names = "MAPE",
                  booktabs = TRUE, caption = "MAPE for Nominal GDP")
```

Table 6: MAPE for Nominal GDP

MAPE
0.1236639

```
causality(fit1, cause = "XAdj")
```

```
## $Granger
##
##  Granger causality H0: XAdj do not Granger-cause YAdj
##
## data:  VAR object fit1
## F-Test = 0.00040461, df1 = 1, df2 = 102, p-value = 0.984
##
##
## $Instant
##
##  H0: No instantaneous causality between: XAdj and YAdj
##
## data:  VAR object fit1
## Chi-squared = 16.987, df = 1, p-value = 3.763e-05
```

4. [3 marks] 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.

{*Solution.*}

```
fit2 <- auto.arima(YAdj, xreg = XAdj,  max.p = 5, max.q = 5, max.d = 2)
summary(fit2)
```

```
## Series: YAdj
## Regression with ARIMA(0,2,1) errors
##
## Coefficients:
##           ma1     xreg
##       -0.8067  42.2256
## s.e.   0.0693   6.7301
##
## sigma^2 estimated as 34564:  log likelihood=-358.29
```

```
## AIC=722.59   AICc=723.07   BIC=728.56
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE     MASE       ACF1
## Training set 26.9703 179.1501 136.3827 -0.1567635 9.947425 0.550019 -0.1662939
```

```
MAPE_ARMA <- MAPE(fit2$x, fit2$fitted)
kableExtra::kable(MAPE_ARMA, "latex", col.names = "MAPE",
                  booktabs = TRUE, caption = "MAPE-ARMA for Nominal GDP")
```

Table 7: MAPE-ARMA for Nominal GDP

MAPE
0.0994743

The model obtained is an integrated MA(1) with external regressor as mentioned on `summary()` above. AIC obtained is 722.59 and MAPE 0.0994743, as reported in this summary and matches the one calculated by our routine.

5. [5 marks] 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.

{*Solution.*}

In order to find the best variable to be used as predictor, I will test all possible combinations adjusting ARMA models and find the minimum AIC possible.

The following code does the job by reading all data-sets (distinguished by their unique vector-ID), processes the `auto.arima()` function and checks if the current model is best than the previous.

At the end, we will get the model with smaller AIC.

```
# This database contains all data
my_data = read_csv("A3_Data/3610020801_databaseLoadingData.csv")

# Auxiliary table to process all models
TabVect <- my_data %>%
  group_by(VECTOR) %>%
  summarise()

# Processes 1st Vector-ID
NewX <- my_data %>%
  filter(VECTOR==as.character(TabVect[1,])) %>%
  pull(VALUE) %>%
  ts(start = 1961, frequency = 1)

TestedAIC <- rep(0,nrow(TabVect))

ft <- auto.arima(YAdj, xreg = NewX,  max.p = 5, max.q = 5, max.d = 2)

# Stores 1st Model as best AIC, including recspective Vector
TestedAIC[1] <- ft$aic
```

```r
AICMin <- TestedAIC[1]
VecMin <- as.character(TabVect[1,])
bestfit <- ft

Exceptions <- c(218) # There is an exception encountered in vector No.218
i <- 2

# Processes a bunch of models to discover the Best Series (smaller AIC)
while (i <= nrow(TabVect)){
  NewX <- my_data %>%
    filter(VECTOR==as.character(TabVect[i,])) %>%
    pull(VALUE) %>%
    ts(start = 1961, frequency = 1)
  if ((length(NewX)==length(YAdj)) && !(i %in% Exceptions)){

    ft <- auto.arima(YAdj, xreg = NewX,  max.p = 5, max.q = 5, max.d = 2)

    TestedAIC[i] <- ft$aic # Stores tested AIC

    if (TestedAIC[i] < AICMin) { # Discovered a best Regressor
      AICMin <- ft$aic
      VecMin <- as.character(TabVect[i,])
      XBest <- NewX
      bestfit <- ft
    }
  }
  i <- i + 1
}
```

We tested 239 models whose AICs are distributed as the following histogram.

```r
cat("\nNo. of Models tested: ", length(which(TestedAIC>0)))
```

```
##
## No. of Models tested:  239
```

```r
Hst <- as.tibble(TestedAIC[which(TestedAIC>0)])
Hst %>%
  ggplot()+
  geom_histogram(aes(x=value), binwidth = 15)+
  labs(x = "AIC", y = "Frequency")
```

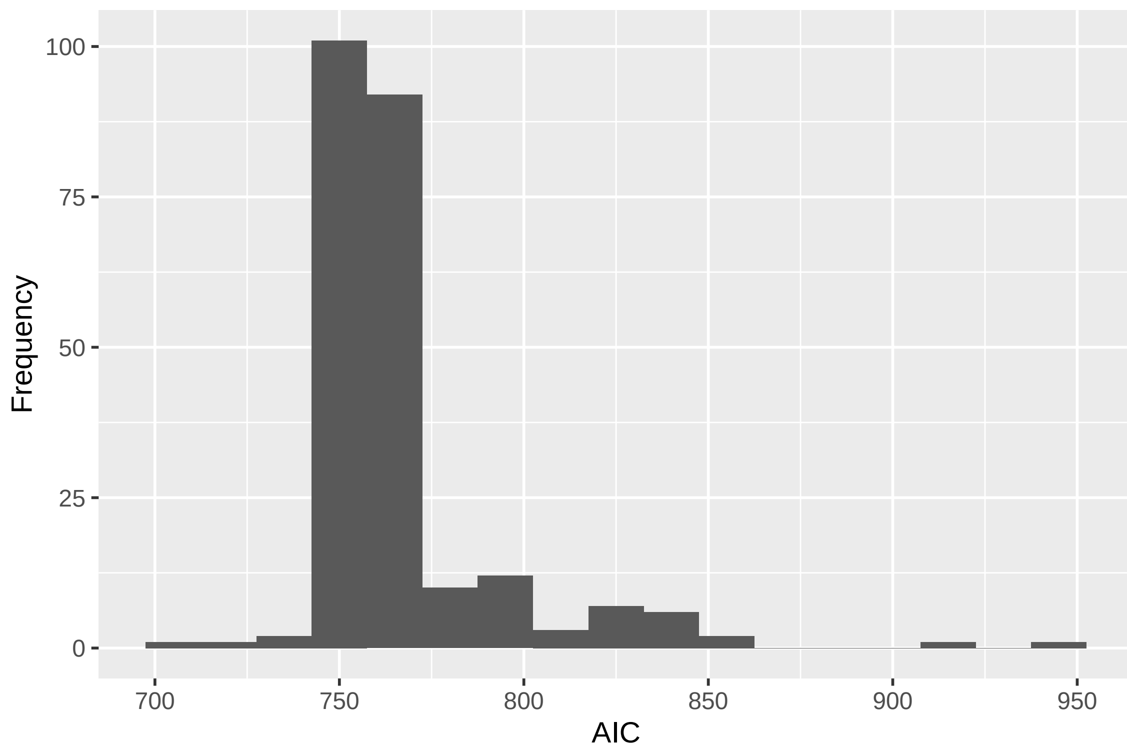


Figure 4: Histogram of AICs of models tested

The best model is $ARMA(1, 1, 2)$ and was obtained from vector $v62458807$ which corresponds to *Labour Compensation* in *Arts, entertainment and recreation [71]* sector.

The present model obtained an $AIC = 708.925$ which is smaller than $AIC = 722.59$ calculated in previous question.

The summary can be verified below.

```r
summary(bestfit)
```

```
## Series: YAdj
## Regression with ARIMA(1,1,2) errors
##
## Coefficients:
##          ar1      ma1     ma2    xreg
##       0.7194  -1.3561  0.6774  1.2154
## s.e.  0.1834   0.1849  0.1147  0.0859
##
## sigma^2 estimated as 20459:  log likelihood=-349.46
## AIC=708.92   AICc=710.15   BIC=718.96
##
## Training set error measures:
##                     ME     RMSE      MAE      MPE     MAPE      MASE        ACF1
## Training set 11.47096 136.4994 93.33773 1.848879 3.740181 0.3764226 -0.01100861
```

```r
BestMAPE_ARMA <- MAPE(bestfit$x, bestfit$fitted)
kableExtra::kable(BestMAPE_ARMA, "latex", col.names = "Best MAPE",
                  booktabs = TRUE, caption = "Best MAPE-ARMA for Nominal GDP")
```

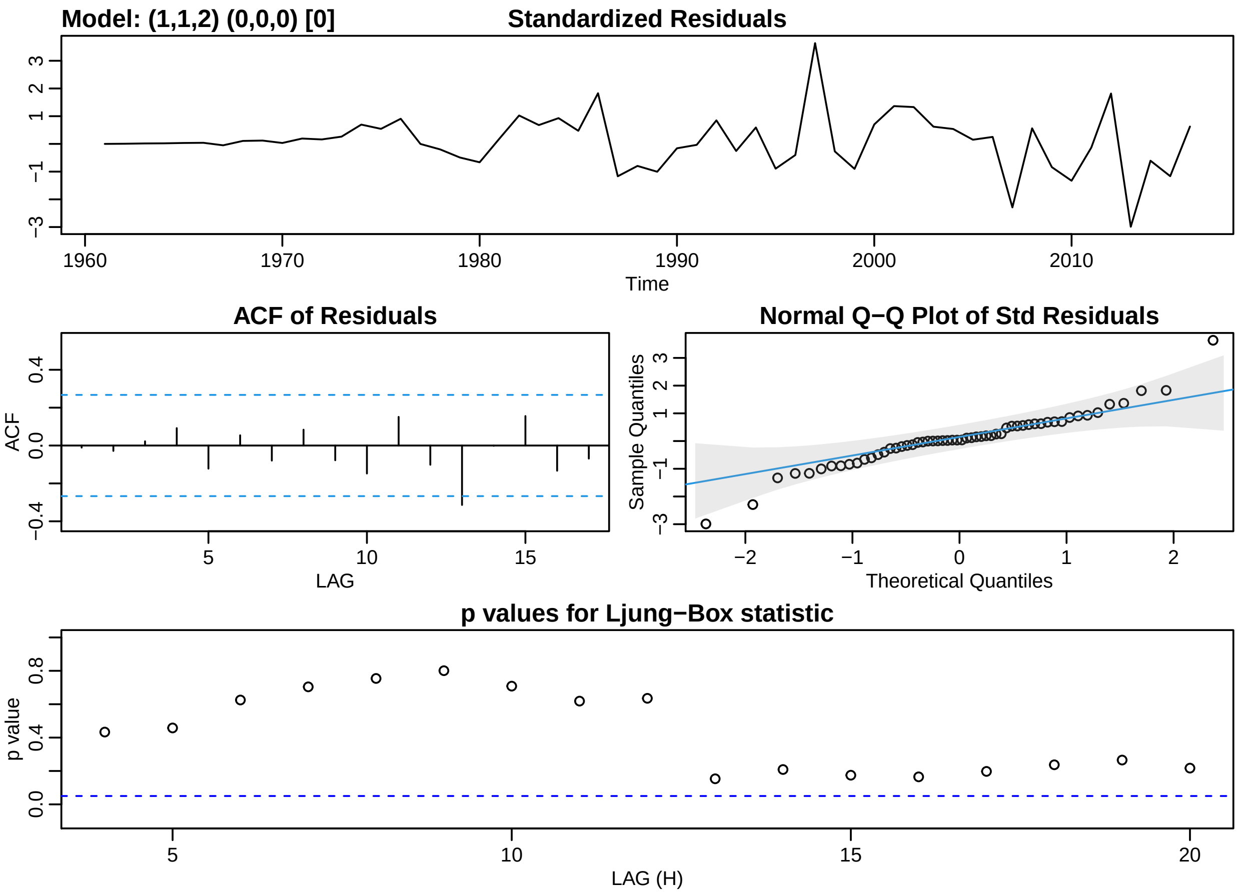
Table 8: Best MAPE-ARMA for Nominal GDP

Best MAPE
0.0374018

Now generating the diagnostics for the new model, we obtained the following:

```
MBest <- sarima(YAdj, 1, 1, 2, P=0, D=0, Q=0, S=0, xreg = XBest, no.constant=FALSE)
```

```
## initial  value 5.135434
## iter   2 value 5.077225
## iter   3 value 5.042140
## iter   4 value 5.027005
## iter   5 value 4.992972
## iter   6 value 4.991498
## iter   7 value 4.990458
## iter   8 value 4.979718
## iter   9 value 4.976558
## iter  10 value 4.970464
## iter  11 value 4.959493
## iter  12 value 4.958665
## iter  13 value 4.954070
## iter  14 value 4.948496
## iter  15 value 4.938089
## iter  16 value 4.935833
## iter  17 value 4.935071
## iter  18 value 4.934381
## iter  19 value 4.933876
## iter  20 value 4.933855
## iter  21 value 4.933820
## iter  22 value 4.933815
## iter  23 value 4.933814
## iter  24 value 4.933814
## iter  24 value 4.933814
## iter  24 value 4.933814
## final  value 4.933814
## converged
## initial  value 4.935715
## iter   2 value 4.935615
## iter   3 value 4.935438
## iter   4 value 4.935175
## iter   5 value 4.935100
## iter   6 value 4.934986
## iter   7 value 4.934932
## iter   8 value 4.934925
## iter   9 value 4.934925
## iter   9 value 4.934925
## iter   9 value 4.934925
## final  value 4.934925
## converged
```



The diagnostics of the best model shows residuals independently distributed and the ACF of residuals doesn't show significant correlation. The normality of standard residuals is preserved in the *Q-Q Plot*, except by 02-influential points at the borders. Finally, Ljung-Box statistic confirms the residuals for lags 1-20 are independent which fits our needs in this study.

6. [10 marks; **STA2202 (grad) students ONLY**] 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: $|Y_{n_i+1} - Y^{n_i}_{n_i+1}|/Y_{n_i+1}$
  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.)

{*Solution.*}

Calculating the predictions for the next 10 values for nominal GDP:

```
# Calculating the One-Step-Ahead prediction for 2007-2016
PctErr <- rep(0, 10)
OneStepAhd <- rep(0, 10)
for (St in 2007:2016) {
```

```r
  tFit <- arima(window(YAdj, start=1961, end=St-1),
                xreg = window(XBest, start=1961, end=(St-1)),  order = c(1, 1, 2))
  fcst <- predict(tFit, newxreg = window(XBest, start=St, end=St), n.ahead = 1)
  OneStepAhd[St-2006] <- fcst$pred
  PctErr[St-2006] <- as.numeric(abs(window(YAdj, start=St, end=St)-
                             OneStepAhd[St-2006])/window(YAdj, start=St, end=St))
}

prtTbl <- cbind(2007:2016,
                as.vector(window(YAdj, start=2007, end=2016)),
                OneStepAhd,
                PctErr)

kableExtra::kable(prtTbl, "latex", col.names = c("Year", "Actual GDP", "Predict GDP", "Pct Err"),
                  booktabs = TRUE, caption = "1-Step Ahead Predictor for Nominal GDP")
```

Table 9: 1-Step Ahead Predictor for Nominal GDP

Year	Actual GDP	Predict GDP	Pct Err
2007	10131.96	10458.50	0.0322288
2008	10727.30	10604.48	0.0114496
2009	10703.59	10878.60	0.0163500
2010	10635.67	10866.47	0.0217001
2011	10936.67	11024.04	0.0079890
2012	11254.75	11078.40	0.0156683
2013	11095.50	11708.83	0.0552773
2014	11520.06	11719.05	0.0172735
2015	11923.07	12119.60	0.0164834
2016	12439.91	12321.01	0.0095576

```r
kableExtra::kable(mean(PctErr), "latex", col.names = "MAPE",
                  booktabs = TRUE, caption = "MAPE of 1-Step Ahead Predictions for Nominal GDP")
```

Table 10: MAPE of 1-Step Ahead Predictions for Nominal GDP

MAPE
0.0203978

As we can see, the predictive performance using *out-of-sample* 1-step-ahead prediction reported smaller MAPE when compared with the one obtained in previous question, as expected.

This concludes the PRACTICE part of the assignment.