QUESTION 1 : DIRECT FORECASTS

Whenever we are not sure about the data generating process, the best we can do is to use the direct forecasting approach.

(a) We are in period T. We want to use the four last observations to make predictions for the next four future values of y.

$$Y_{T+4} = Y_{T+4} - Y_T$$

Lagged values of Y are useful for predicting future values of Y.

Direct forecast for period T+1

$$Y^1_{T+1} = \alpha^1_1 Y_T + \alpha^1_2 Y_{T-1} + \alpha^1_3 Y_{T-2} + \alpha^1_4 Y_{T-3} + U^1_{T+1}$$

$$Y^2_{T+2} = \alpha^2_1 Y_T + \alpha^2_2 Y_{T-1} + \alpha^2_3 Y_{T-2} + \alpha^2_4 Y_{T-3} + U^2_{T+2}$$

$$Y^3_{T+3} = \alpha^3_1 Y_T + \alpha^3_2 Y_{T-1} + \alpha^3_3 Y_{T-2} + \alpha^3_4 Y_{T-3} + U^3_{T+3}$$

$$Y^4_{T+4} = \alpha^4_1 Y_T + \alpha^4_2 Y_{T-1} + \alpha^4_3 Y_{T-2} + \alpha^4_4 Y_{T-3} + U^4_{T+4}$$

(b) We are in period T-1 trying to use the last four observations to make predictions on the next four future values of Y.

$$Y^1_T = \alpha^1_1 Y_{T-1} + \alpha^1_2 Y_{T-2} + \alpha^1_3 Y_{T-3} + \alpha^1_4 Y_{T-4} + U^1_T$$

$$Y^2_{T+1} = \alpha^2_1 Y_{T-1} + \alpha^2_2 Y_{T-2} + \alpha^2_3 Y_{T-3} + \alpha^2_4 Y_{T-4} + U^2_{T+1}$$

$$Y^3_{T+2} = \alpha^3_1 Y_{T-1} + \alpha^3_2 Y_{T-2} + \alpha^3_3 Y_{T-3} + \alpha^3_4 Y_{T-4} + U^3_{T+2}$$

$$Y^4_{T+3} = \alpha^4_1 Y_{T-1} + \alpha^4_2 Y_{T-2} + \alpha^4_3 Y_{T-3} + \alpha^4_4 Y_{T-4} + U_{T+3}$$

(c) We are in period T-2, making predictions for 4 periods ahead

$$Y^1_{T-1} = \alpha^1_1 Y_{T-2} + \alpha^1_2 Y_{T-3} + \alpha^1_3 Y_{T-4} + \alpha^1_4 Y_{T-5} + U^1_{T-1}$$
$$Y^2_{T} = \alpha^2_1 Y_{T-2} + \alpha^2_2 Y_{T-3} + \alpha^2_3 Y_{T-4} + \alpha^2_4 Y_{T-5} + U^2_{T}$$
$$Y^3_{T+1} = \alpha^3_1 Y_{T-2} + \alpha^3_2 Y_{T-3} + \alpha^3_3 Y_{T-4} + \alpha^3_4 Y_{T-5} + U^3_{T+1}$$
$$Y^4_{T+2} = \alpha^4_1 Y_{T-2} + \alpha^4_2 Y_{T-3} + \alpha^4_3 Y_{T-4} + \alpha^4_4 Y_{T-5} + U^4_{T+2}$$

(d) When the shock happens in period T+1, for instance, and given that we have $Y^h_{T+h} = a^h(Y_{T+h} - Y_T)$, the overlapping data Y_{T+1} is included in $Y^1_{T+1} = a^1(Y_{T+1} - Y_T)$, $Y^2_{T+2} = a^2(Y_{T+2} - Y_T)$, ...

This shock was not foreseen in period T when trying to make a forecast for period T+2. As such, the forecast for Y^2_{T+2} will be too low. Namely, $U^2_{T+2} > 0$.

Similarly, the shock was not known in period T-1 when making a forecast for T+1. Thus, the forecast for T+1 will be too low because it doesn't incorporate the unexpected event in T+1. That is $U^2_{T+1} > 0$

Hence, the error term in multiperiod regression is serially correlated.

$$Y_{T+4} = Y_{T+4} - Y_T$$
$$Y_{T+3} = Y_{T+3} - Y_T$$
$$Y_{T+2} = Y_{T+2} - Y_T$$
$$Y_{T+1} = Y_{T+1} - Y_T$$

By construction, U^h_{T+h} will be serially correlated. HAC standard errors will be necessary.

(e) $Y_t = \alpha(L) Y_t + \beta(L) X_t + \gamma(L) Z_t + U_t$

The expanded model if the $\alpha(L)$ polynomial has 3 lags, the $\beta(L)$ has 2 lags and $\gamma(L)$ has 1 lag:

Version 1:

$$Y_t = \alpha_0 + \alpha_{11} Y_t + \alpha_{12} Y_{t-1} + \alpha_{13} Y_{t-2} + \alpha_{14} Y_{t-2} +$$
$$+ \beta_0 + \beta_{21} X_t + \beta_{22} X_{t-1} + \beta_{23} X_{t-2} +$$
$$+ \gamma_0 + \gamma_{31} Z_t + \gamma_{32} Z_{t-1} + U_t$$

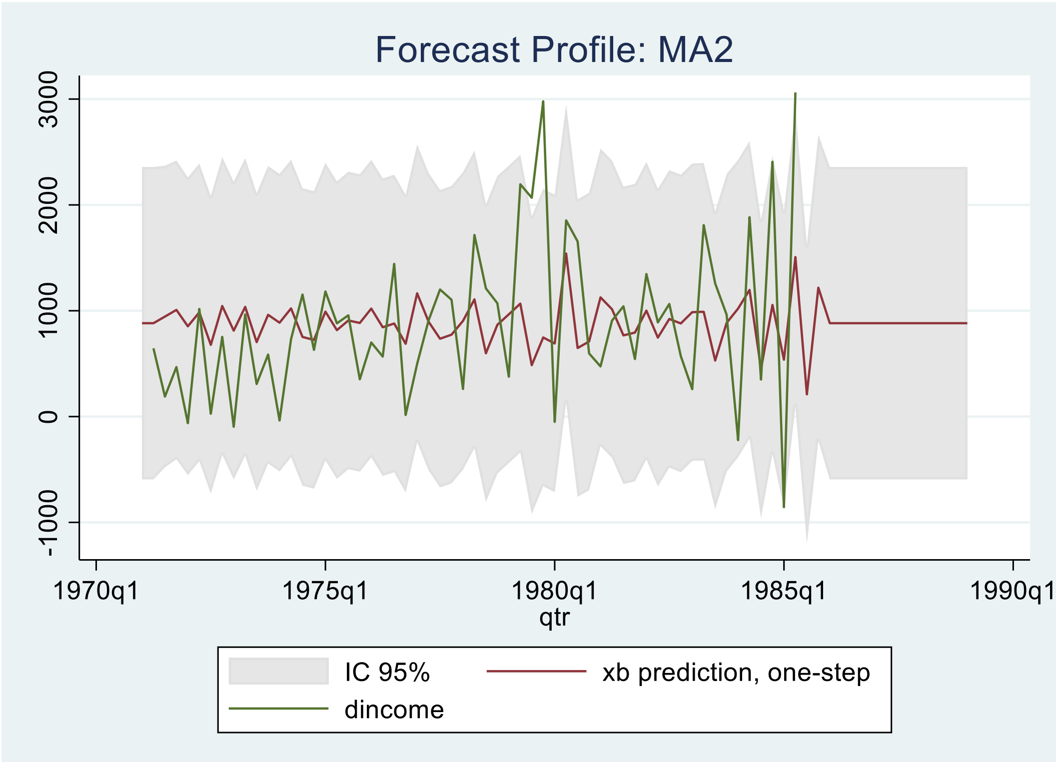
Version 2:

$$Y_t = \alpha_0 + \alpha_{11} Y_t + \alpha_{12} L Y_t + \alpha_{13} L^2 Y_t + \alpha_{14} L^3 Y_t +$$
$$+ \beta_0 + \beta_{21} X_t + \beta_{22} L X_t + \beta_{23} L^2 X_t +$$
$$+ \gamma_0 + \gamma_{31} Z_t + \gamma_{32} L Z_t + U_t$$

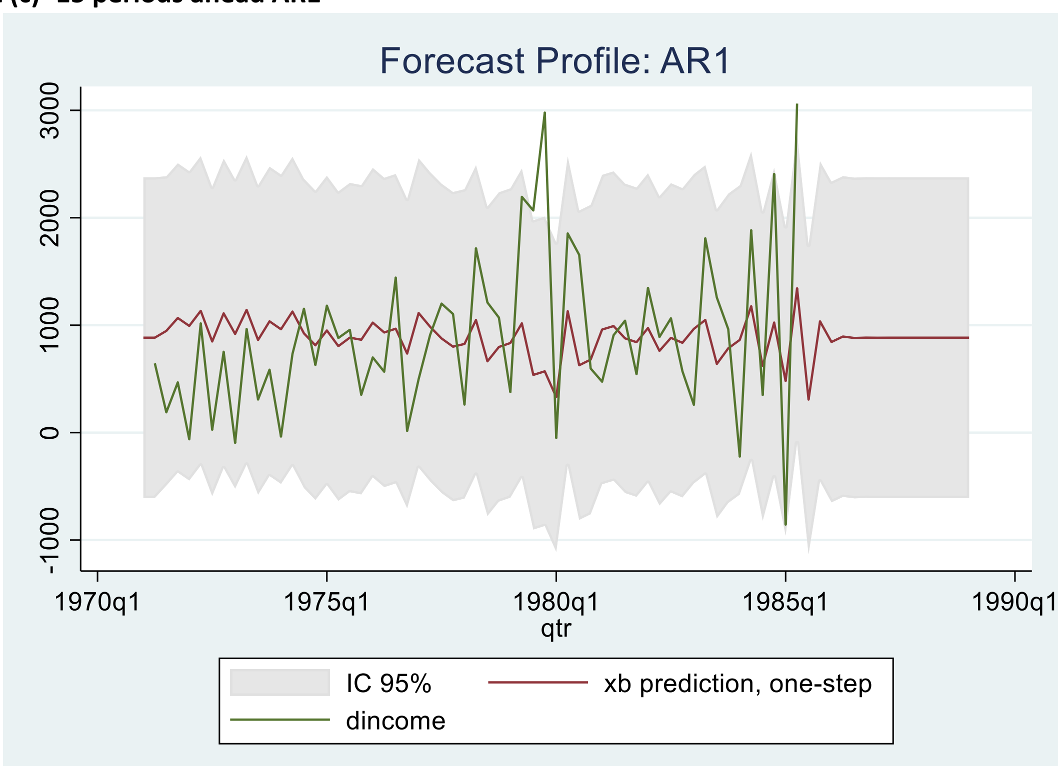
More concisely,

$$\alpha(L) Y_t = \alpha_{11} Y_t + \alpha_{12} L Y_t + \beta_{13} L^2 Y_t + \beta_{14} L^3 Y_t$$
$$\alpha(L) X_t = \beta_{21} X_t + \beta_{22} L X_t + \beta_{23} L^2 X_t$$
$$\alpha(L) Y_t = \gamma_{31} Z_t + \gamma_{32} L Z_t$$

Question 2 (b)- 15 Periods ahead MA2 forecasting

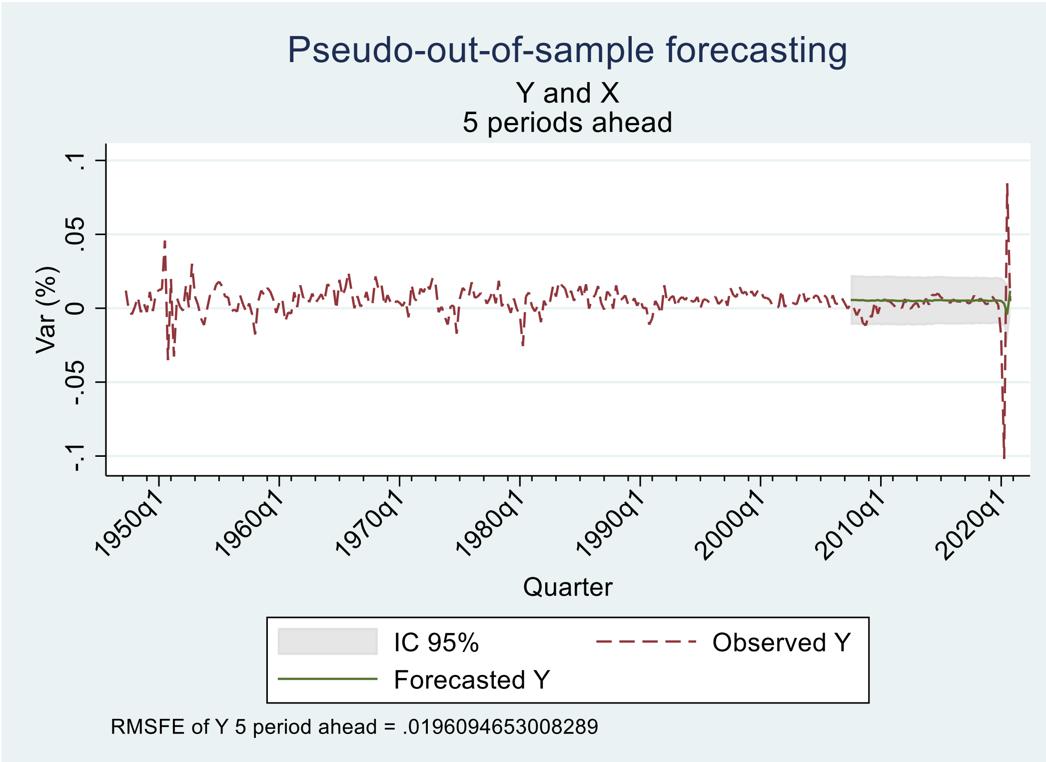


Question 2 (c)- 15 periods ahead AR1

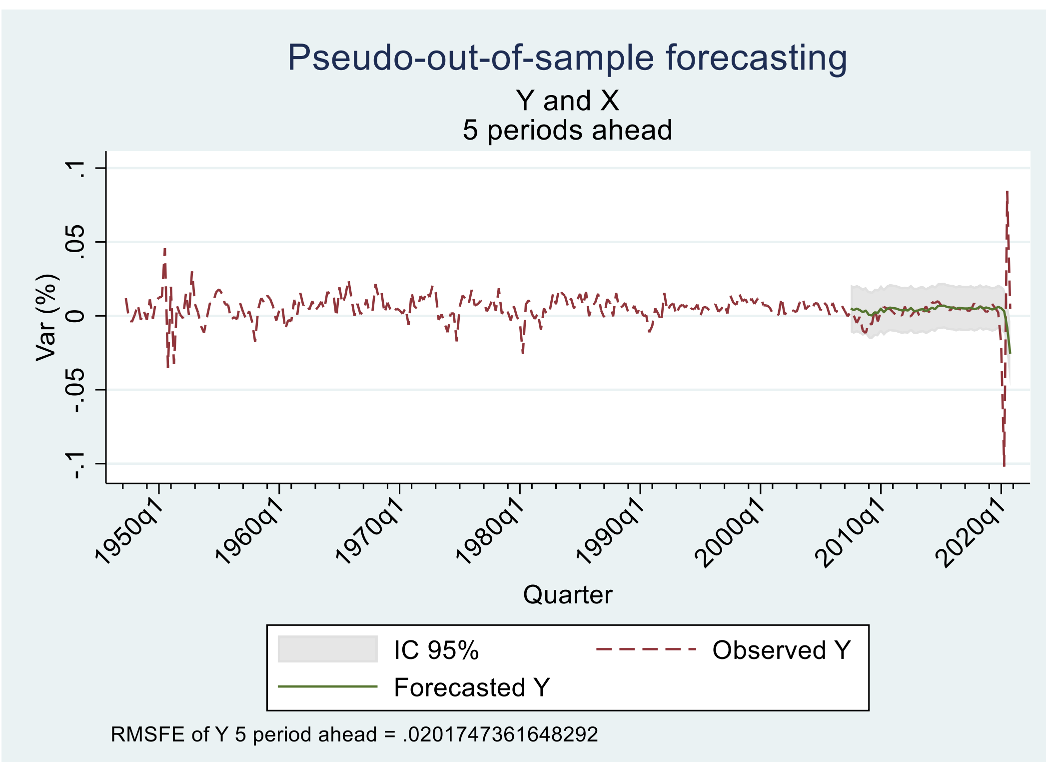


forecasting

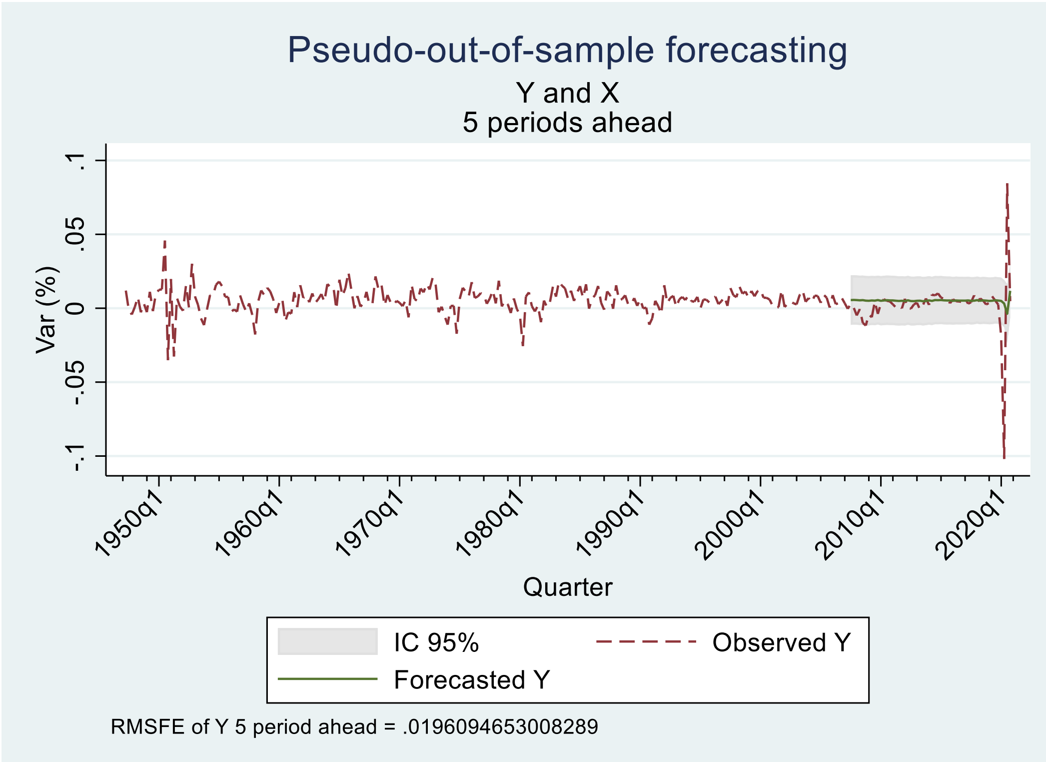
Question 3 (c)- ADL(1,1) forecast graph



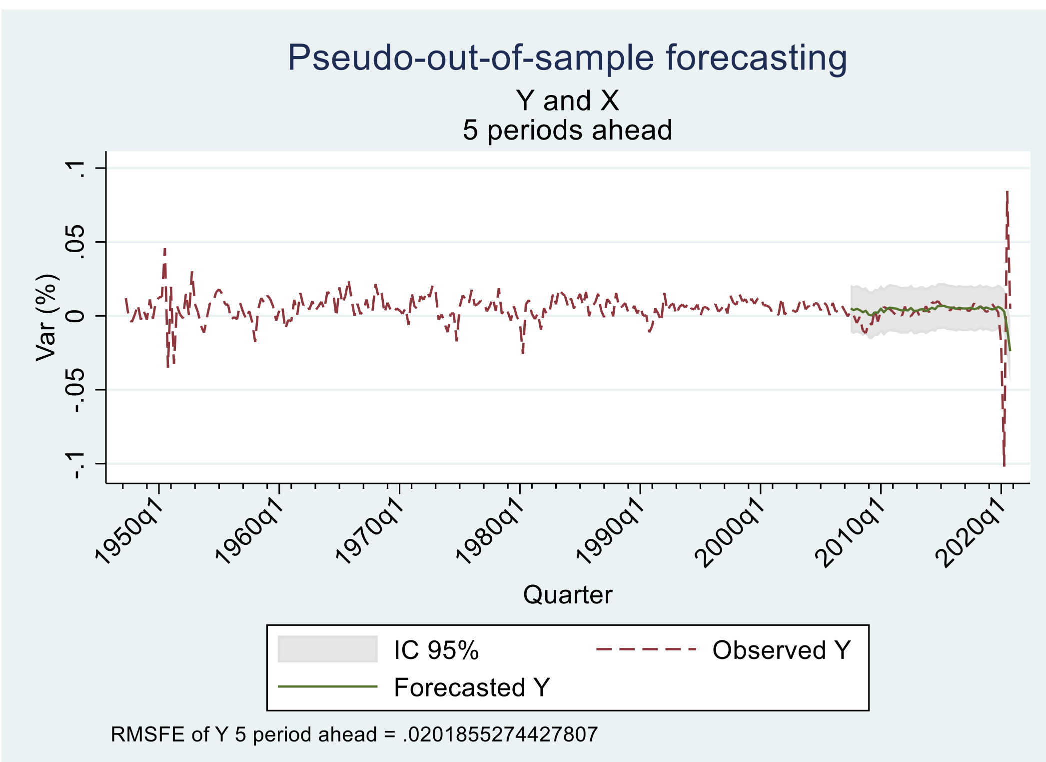
Question 3 (c)- ADL(3,1) forecast graph



Question 3 (g)- AR(1) forecast graph



Question 3 (g)- AR(2) forecast graph



```
clear
**setting directories
cd "C:\Users\Fjolla\Desktop\Time Series\HW3"

global data "C:\Users\Fjolla\Desktop\Time Series\HW3"

global workfiles "C:\Users\Fjolla\Desktop\Time Series\HW3"

**ANSWERS TO QUESTION 2
log using A3_Q2.log, replace
**loading data
use "income.dta"

gen year= .
replace year = 1971 in 1
forvalues i = 5(4)58 {
replace year = year[`i'-4] +1 in `i'
}
replace year = year[_n-1] if year == .
gen qtr = yq(year, time_01)
format qtr %tq
tsset qtr
gen dincome = income - income[_n-1]

************* a ****************
*** estimate model: AR1 and MA2
arima dincome, arima(1,0,0) vce(robust) nolog
predict pre1, xb
predict u1, resid
gen u1_2 = u1^2
summarize u1_2
scalar rmse_ar1 = sqrt(r(mean))
estat ic

scalar ar1_AIC = r(S)[1,5]
scalar ar1_BIC = r(S)[1,6]

arima dincome, arima(0,0,2) vce(robust) nolog
predict pre2, xb
predict u2, resid
gen u2_2 = u2^2
summarize u2_2
scalar rmse_ma2 = sqrt(r(mean))
estat ic
scalar ma2_AIC = r(S)[1,5]
scalar ma2_BIC = r(S)[1,6]

*** compare RMSE
di rmse_ar1 - rmse_ma2

*The MA2 model has a lower RMSE than the AR1 model by 17.415578.

*** Diebold-Mariano forecast comparison test

dmariano dincome pre1 pre2

**The null hypothesis of the Diebold-Mariano test is that the two forecasts have the same level
of accuracy.The alternative hypothesis is that the two forecasts have different levels of
accuracy. With a p-value of 0.2216 we fail to reject the null hypothesis that the two model
forecasts have the same accuracy level at any of the standard significance levels.

*** RMSE test
** simple one: rmse_ar1/rmse_ma2
di rmse_ar1/rmse_ma2

** rigorous one: under the assumption that the residuals from AR(1) and MA(2) are independent and
no autocorrelation in residuals of AR1 and MA2, the MSFE_AR1 - MSFE_MA2 is asymptotically
normally distributed as N(0, (v1+v2)/N), under the H0 that MSFE_AR1 = MSFE_MA2. s1 is the
```

```
variance of the squared residual from AR1 and s2 is the variance of the squared residual from MA2
scalar msfe_ar1 = rmse_ar1^2
scalar msfe_ma2 = rmse_ma2^2
qui summarize u1_2
scalar v1 = r(Var)
qui summarize u2_2
scalar v2 = r(Var)
scalar t = (msfe_ar1 - msfe_ma2)/sqrt((v1+v2)/_N)
di 1 - normal(t)

*Again, when we perform an RMSE test, both the simple and the more rigorous approach lead us to
the same conclusion. That is, althought the MA2 model has a smaller RMSEF, we fail to reject the
null hypothesis that the MSFE for the two models is equal according to the rigorous approach of
the rmsfe test. Our p-value of the test is .42544852. Hence, we cannot reject H0 regardless of
any standard significance levels alpha.

** I pick MA2 as the model to do forecast.
di ar1_AIC - ma2_AIC
di ar1_BIC - ma2_BIC

*Regardless of the fact that in the Diebold-Mariano forecast comparison test as well as RMSE test
I do not reject the H0 that the two models are equal, for forecasting purposes I pick the MA2
model. This model has a smaller AIC but a larger BIC than the AR1 model. Moreover, the MA2 model
is also my first choice in terms of fit. That is, the second lag of the MA2 model is significant
at the significance level of 5% with a p-value of 0.042. The coefficient of the AR2 model, on the
other hand, does not prove to be statistically significant.

************* b *****************
** for the MA2 model

tsappend, add(15)
qui arima dincome, arima(0,0,2) vce(robust)
predict yhat, xb
predict fvar, mse
gen upper=yhat + 1.96*sqrt(fvar)
gen lower=yhat - 1.96*sqrt(fvar)
twoway (rarea lower upper qtr, bcolor(gs14) legend(lab(1 "IC 95%"))) (line yhat qtr) (line dincome
 qtr), title(Forecast Profile: MA2)

graph export forecast_ma2_profile.pdf, replace


*(b) The first problem I can observe is the fact that the predicted forecast falls outside of the
bands of prediction error in the first quarter of 1980 and the first quarter of 1985. This
decreases my confidence slightly that this is a good prediction. The second problem is that our
forecast is a straight line, and fails to give us any useful insight for the 15 periods ahead.

************* c *****************

** for the AR1 model
tsappend, add(15)
qui arima dincome, arima(1,0,0) vce(robust)
predict yhat1, xb
predict fvar1, mse
gen upper1=yhat1 + 1.96*sqrt(fvar1)
gen lower1=yhat1 - 1.96*sqrt(fvar1)
twoway (rarea lower1 upper1 qtr, bcolor(gs14) legend(lab(1 "IC 95%"))) (line yhat1 qtr) (line
dincome qtr), title(Forecast Profile: AR1)

graph export forecast_ar1_profile.pdf, replace


**(c) For the AR(1) model I get a very similar graph to that of the MA(2) model. This is not
exactly what I would expect to see. AR(1) is generally speaking a better model for predicting
income data than an MA(2) model. The fact that my results do not show any significant differences
makes me skeptical that the data are possibly simulated data instead of real income data.
```

```
*Answers to question 3

drop _all
clear all

cd "C:\Users\Fjolla\Desktop\Time Series\HW3"

global data "C:\Users\Fjolla\Desktop\Time Series\HW3"

global workfiles "C:\Users\Fjolla\Desktop\Time Series\HW3"

**loading data
use "consump.dta"

*** STATA code for pseudo-out-of-sample forcasting ***

describe //see the description of your data

di _N

// Generate variables

forvalues i = 1/20 {
forvalues j = 1/20 {
*di `i' `j'
qui gen f_Y_X1`i'_`j' = .       //generate space for forecasts
qui gen stdf_Y_X1`i'_`j' = .    //generate space for forecasts sd
qui gen sqdiff1`i'_`j' = .      //generate space for sum of square differences

***pseudo-out-of-sample (POOS) model to estimate

// First estimate the model with data you want to use to generate the model parameters

forvalues p = `=tq(2006q2)'/`=tq(2020q4)' {
di `p'
qui regress Y L(1/`i').Y L(1/`j').X if quarter<`p'
*What does Temp_T_hat do?

qui predict Temp_Y_hat, xb  // save predicted values of the regression
qui predict Temp_Y_rmsfe, stdf  // save standard deviation of the error of the regression

qui replace f_Y_X1`i'_`j' = Temp_Y_hat if quarter==`p'+1 // use the predicted values as in-sample forecast

qui replace stdf_Y_X1`i'_`j' = Temp_Y_rmsfe if quarter==`p'+1 // use the s.d. of error of the previous regression, as in-sample s.d.

qui replace sqdiff1`i'_`j' = (Y - Temp_Y_hat)^2 if quarter==`p'+1 // compute squared forecast errors

qui drop Temp_Y_hat Temp_Y_rmsfe
}

***compute the information criteria for the i lag
qui estat ic  //Akaike’s and Schwarz’s Bayesian information criteria
mat IC = r(S) //save results
scalar BIC`i'_`j' = el(IC, 1, 6)  //define a matrix for BIC
scalar AIC`i'_`j' = el(IC, 1, 5)  //define a matrix for AIC

***compute the RMSFE for the i lag
qui egen meansqdiff`i'_`j' = mean(sqdiff1`i'_`j')
qui scalar RMSFE`i'_`j' = sqrt(meansqdiff`i'_`j')

qui drop meansqdiff`i'_`j'
}
}
```

```

***find the best criteria between all lags
// generate space for the minimum values of the criteria
scalar minBIC = .
scalar minAIC = .
scalar minRMSFE = .

// find the minimum values of the criteria
forvalues i = 1/20 {
forvalues j =1/20 {
if BIC`i'_`j' < minBIC {
scalar minBIC = BIC`i'_`j'
local minBICij ",`i' , `j'"
}
if AIC`i'_`j' < minAIC {
scalar minAIC = AIC`i'_`j'
local minAICij ",`i' , `j'"
}
if RMSFE`i'_`j' < minRMSFE {
scalar minRMSFE = RMSFE`i'_`j'
local minRMSFEij ",`i' , `j'"
}
}
}

// display your results
di "min BIC lag"`minBICij' " : " minBIC
di "min AIC lag"`minAICij' " : " minAIC
di "min RMSFE lag"`minRMSFEij' " : " minRMSFE

//do a relative MSFE test for ADL(1,1) and ADL(3,1) models
local T1 = tq(2006q2)
local T2 = tq(2020q4)
local h = 1
qui total(sqdiff11_1) if quarter <= `T2' & quarter >= `T1' + `h'
local MSFE1_1 = e(b)[1,1]
qui total(sqdiff13_1) if quarter <= `T2' & quarter >= `T1' + `h'
local MSFE3_1 = e(b)[1,1]
di (1/(`T2' - `T1' - `h' + 1)*`MSFE3_1')/(1/(`T2' - `T1' - `h' + 1)*`MSFE1_1')

** (a)The first part of the code takes approximately 4 minutes and 16 seconds to complete. The reason why the first part of the code takes so much longer is because the code loops over the 1 to 20 lags of the distributed lag model and the Autoregressive distributed lag model while computing all the criteria specified in the code. On the other hand, the second part of the code involves no such forloop. The lines of code that take the longest to compute are lines where we generate variables and space for forecasts. This is because these lines tell stata to compute all the lags of the models and repeat the process of computing all the criteria specified in the forloop for multiple times, without having to specify each one of them individually.  In general, with larger p and q values it takes more time to calculate the maximum likelihood of the ARIMA model.


** (b) The AR model without x is not evaluated in the set of models we consider in the first set of the code. To change this we need to change our forloop. Namely, we can simply take out the j loop out from the code.

*** First run the code until here and see which models are your "best" ones, given the criteria you used
*****
*** Now estimate the POOS model with the lags of the models that have the minimum critiria values found above.
*****
***`p'+1 is for one period ahead forecast and you can add more periods changing the "+1" to "+h"
*****

*ADL(1,1)

qui gen f_Y_X1 = .
```

```
qui gen stdf_Y_X1 = .
qui gen sqdiff1 = .
forvalues p = `=tq(2006q2)'/`=tq(2020q4)' {

//qui regress Y L(1/1).Y L(1/1).X if quarter<`p'
qui regress Y L(1/1).Y if quarter<`p'

qui predict Temp_Y_hat, xb
qui predict Temp_Y_rmsfe, stdf

qui replace f_Y_X1 = Temp_Y_hat if quarter==`p'+5
qui replace stdf_Y_X1 = Temp_Y_rmsfe if quarter==`p'+5

qui replace sqdiff1 = (Y - Temp_Y_hat)^2 if quarter==`p'+5

qui drop Temp_Y_hat Temp_Y_rmsfe
}

***Compute the RMSFE for the POOS estimated model
qui egen meansqdiff1 = mean(sqdiff1)
qui scalar RMSFE_Y_X1 = sqrt(meansqdiff1)
di "RMSFE of Y and X 1 periods ahead =" %9.4f scalar(RMSFE_Y_X1)

***Compute the confidence intervals
qui gen fcastHigh = f_Y_X1 + 1.96*stdf_Y_X1
qui gen fcastLow = f_Y_X1 - 1.96*stdf_Y_X1

***graph
qui twoway (rarea fcastLow fcastHigh quarter, bcolor(gs14) legend(lab(1 "IC 95%"))) (tsline Y
f_Y_X1, lpattern(dash solid) legend(lab(2 "Observed Y") lab(3 "Forecasted Y"))), title(
"Pseudo-out-of-sample forecasting") subtitle("Y and X" "5 periods ahead") ytitle("Var (%)") xtitle
("Quarter") xlabel(#10, angle(45)) xmticks(#40) note("RMSFE of Y 5 period ahead =
`=scalar(RMSFE_Y_X1)'") name(f_YX1_graph, replace)


*ADL(3,1)


qui gen f_Y_X1 = .
qui gen stdf_Y_X1 = .
qui gen sqdiff1 = .
forvalues p = `=tq(2006q2)'/`=tq(2020q4)' {

//qui regress Y L(3/1).Y L(3/1).X if quarter<`p'
qui regress Y L(3/1).Y if quarter<`p'

qui predict Temp_Y_hat, xb
qui predict Temp_Y_rmsfe, stdf

qui replace f_Y_X1 = Temp_Y_hat if quarter==`p'+5
qui replace stdf_Y_X1 = Temp_Y_rmsfe if quarter==`p'+5

qui replace sqdiff1 = (Y - Temp_Y_hat)^2 if quarter==`p'+5

qui drop Temp_Y_hat Temp_Y_rmsfe
}

***Compute the RMSFE for the POOS estimated model
qui egen meansqdiff1 = mean(sqdiff1)
qui scalar RMSFE_Y_X1 = sqrt(meansqdiff1)
di "RMSFE of Y and X 1 periods ahead =" %9.4f scalar(RMSFE_Y_X1)

***Compute the confidence intervals
qui gen fcastHigh = f_Y_X1 + 1.96*stdf_Y_X1
qui gen fcastLow = f_Y_X1 - 1.96*stdf_Y_X1

***graph
qui twoway (rarea fcastLow fcastHigh quarter, bcolor(gs14) legend(lab(1 "IC 95%"))) (tsline Y
```

```
f_Y_X1, lpattern(dash solid) legend(lab(2 "Observed Y") lab(3 "Forecasted Y"))), title("Pseudo-out-of-sample forecasting") subtitle("Y and X" "5 periods ahead") ytitle("Var (%)") xtitle("Quarter") xlabel(#10, angle(45)) xmticks(#40) note("RMSFE of Y 5 period ahead = `=scalar(RMSFE_Y_X1)'") name(f_YX1_graph, replace)



**(c)

*The two models appear to be very close when it comes to their graphical representations. The shaded area in grey is the 95% confidence interval of our predicted forecast. The ADL(1,1) model appears to have a slightly narrower confidence band than the ADL(3,1) model. This is possibly indicative that the ADL(1,1) model gives us a slightly more concise predicted forecast.
*Given that var(%) appears to be highest at the very beginning of the time period (1950s) and then gradually decreasing as time goes by, I would conclude that the predicted forecast does follow the real data, with the var(%) being lowest by the end of the period around the forecasted area.
*As I was also expecting, only the ADL(1,1) model was able to predict 2020. However, none of the two models seem to improve as 2020 evolves.

**(d)

*The ADL(1,1) model has an RMSEF OF ~0.019609, while the ADL(3,1) model has an RMSEF of ~0.020175 for the period forecasted bewteen 2006 q2 and 2020 q4. The ADL(1,1) does slightly better than the ADL(3,1) model. However, this slight difference in RMSEF is what defines whether or not we are able to correctly forecast. Hence, this difference is not negligible.


//do a relative MSFE test for ADL(1,1) and ADL(3,1) models
local T1 = tq(2006q2)
local T2 = tq(2020q4)
local h = 1
qui total(sqdiff11_1) if quarter <= `T2' & quarter >= `T1' + `h'
local MSFE1_1 = e(b)[1,1]
qui total(sqdiff13_1) if quarter <= `T2' & quarter >= `T1' + `h'
local MSFE3_1 = e(b)[1,1]
di (1/(`T2' - `T1' - `h' + 1)*`MSFE3_1')/(1/(`T2' - `T1' - `h' + 1)*`MSFE1_1')


**(e)

*The local MSFE is bigger for the ADL(1,1) model than the ADL(3,1) model by roughly 0.9778. The better model is the one with a smaller local MSFE. Given that the ADL(1,1) model was performing better than the ADL(3,1) model in the previous parts of the problem, I did not expect this to be the case.


forvalues i = 1/20 {
*di `i'
qui gen f_Y_X1`i' = .      //generate space for forecasts
qui gen stdf_Y_X1`i' = .   //generate space for forecasts sd
qui gen sqdiff1`i' = .     //generate space for sum of square differences

***pseudo-out-of-sample (POOS) model to estimate

// First estimate the model with data you want to use to generate the model parameters

forvalues p = `=tq(2006q2)'/`=tq(2020q4)' {
di `p'
qui regress Y L(1/`i').Y if quarter<`p'

qui predict Temp_Y_hat, xb  // save predicted values of the regression
qui predict Temp_Y_rmsfe, stdf  // save standard deviation of the error of the regression

qui replace f_Y_X1`i' = Temp_Y_hat if quarter==`p'+1 // use the predicted values as in-sample forecast

qui replace stdf_Y_X1`i' = Temp_Y_rmsfe if quarter==`p'+1 // use the s.d. of error of the
```

```
previous regression, as in-sample s.d.

qui replace sqdiff1`i' = (Y - Temp_Y_hat)^2 if quarter==`p'+1 // compute squared forecast errors

qui drop Temp_Y_hat Temp_Y_rmsfe
}

***compute the information criteria for the i lag
qui estat ic  //Akaike’s and Schwarz’s Bayesian information criteria
mat IC = r(S) //save results
scalar BIC`i' = el(IC, 1, 6)  //define a matrix for BIC
scalar AIC`i' = el(IC, 1, 5)  //define a matrix for AIC

***compute the RMSFE for the i lag
qui egen meansqdiff`i' = mean(sqdiff1`i')
qui scalar RMSFE`i' = sqrt(meansqdiff`i')

qui drop meansqdiff`i'
}


***find the best criteria between all lags
// generate space for the minimum values of the criteria
scalar minBIC = .
scalar minAIC = .
scalar minRMSFE = .

// find the minimum values of the criteria
forvalues i = 1/20 {
if BIC`i'< minBIC {
scalar minBIC = BIC`i'
local minBICi ",`i'"
}
if AIC`i' < minAIC {
scalar minAIC = AIC`i'
local minAICi ",`i'"
}
if RMSFE`i' < minRMSFE {
scalar minRMSFE = RMSFE`i'
local minRMSFEi ",`i'"
}
}

// display your results
di "min BIC lag"`minBICi' " : " minBIC
di "min AIC lag"`minAICi' " : " minAIC
di "min RMSFE lag"`minRMSFEi' " : " minRMSFE


*(f)

*After including an AR model without x and its lags to our forvalues and re-running the first part of the code, the criteria suggests that we use an AR(1) model based on the BIC criteria, an AR(2) model based on the AIC criteria and an AR(1) model based on the RMSFE criteria.

*(g)

*There does not seem to be a huge improvement in the graphical representation of the data with the AR(1) and AR(2) models in comparison to the ADL(1,1) and ADL(3,1). The AR(1) model performs almost exactly the same as the ADL(1,1) model, except that the BIC value of the AR(1) model is slightly smaller. In a similar note, the new graphs also seem to be fairly close to the real data, but without any significant differences from those on the previous two models. The AR(1) model was able to predict 2020, while the AR(2) model was not. However, even these models do not seem to be able to improve as 2020 is evolving.


//relative MSFE test for the AR(1) and AR(2) models

```

```
local T1 = tq(2006q2)
local T2 = tq(2020q4)
local h = 1
qui total(sqdiff1_1) if quarter <= `T2' & quarter >= `T1' + `h'
local MSFE1_1 = e(b)[1,1]
qui total(sqdiff2_1) if quarter <= `T2' & quarter >= `T1' + `h'
local MSFE3_1 = e(b)[1,1]
di (1/(`T2' - `T1' - `h' + 1)*`MSFE3_1')/(1/(`T2' - `T1' - `h' + 1)*`MSFE1_1')

*(h)

*According to the relative MSFE I have performed, the models under (f) appear to be performing slighlty better.

```