A close-up of a logo

AI-generated content may be incorrect.

|  |  |
| --- | --- |
| Assignment Cover Sheet | |
| Candidate Number | 740094051 |
| Module Code | BEEM012\_A\_2\_202425 |
| Module Name | Applied Econometrics 2 |
| Assignment Title | Problem Set 1 |

Within the Business School we support the responsible and ethical use of GenAI tools, and we seek to develop your ability to use these tools to help you study and learn. An important part of this process is being transparent about how you have used GenAI tools during the preparation of your assignments. Information about GenAI can be found [here](https://universityofexeteruk.sharepoint.com/sites/StudyZone/SitePages/Generative-Artificial-Intelligence-(AI)-guidance.aspx) and guidance on the responsible use of GenAI tools can be found [here](https://universityofexeteruk.sharepoint.com/sites/StudyZone/SitePages/Using-GenAI-to-enhance-your-learning.aspx).

The below declaration is intended to guide transparency in the use of GenAI tools, and to assist you in ensuring the appropriate citation of those tools within your work.

I have / have not [delete as appropriate] used GenAI tools in the production of this work.

The following GenAI tools have been used: [please specify] ……………………………………

I have used GenAI tools to assist with research or gathering information.

I have used GenAI tools to help me understand key theories and concepts.

I have used GenAI tools to help me analyse data.

I have used GenAI tools to create code.

I have used GenAI tools to suggest a plan or structure of my assessment.

I have used GenAI tools to give me feedback on a draft.

I have used GenAI tool to generate images, figures or diagrams.

I have used GenAI tools to generate creative content for my work.

I have used GenAI tools to proofread and correct grammar or spelling errors.

Other [please specify] troubleshoot a piece of code

  I declare that I have referenced use of GenAI tools and outputs within my assessment in line with the [University guidelines](https://libguides.exeter.ac.uk/referencing/generativeai) for referencing GenAI in academic work.

BEEM012 – Problem Set 1

Dimitrie Liviu Constantinescu

February 17, 2025

1 Descriptive Analysis

* 1. Data Description

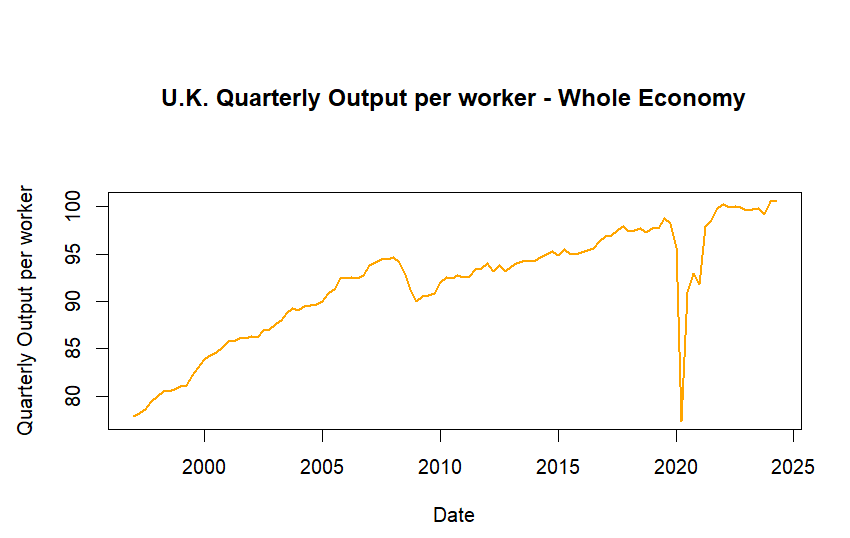
The outcome variable is the output per worker in the UK, a measure of productivity. The explanatory variable is the unemployment rates of NON-UK nationals.

This study intends to answer if the unemployment rate of NON-UK nationals has predictive power for the output per worker in the UK. A study by Parello found that the migration has positive effects on innovation and growth in the innovating economy (2022, p. 1163).

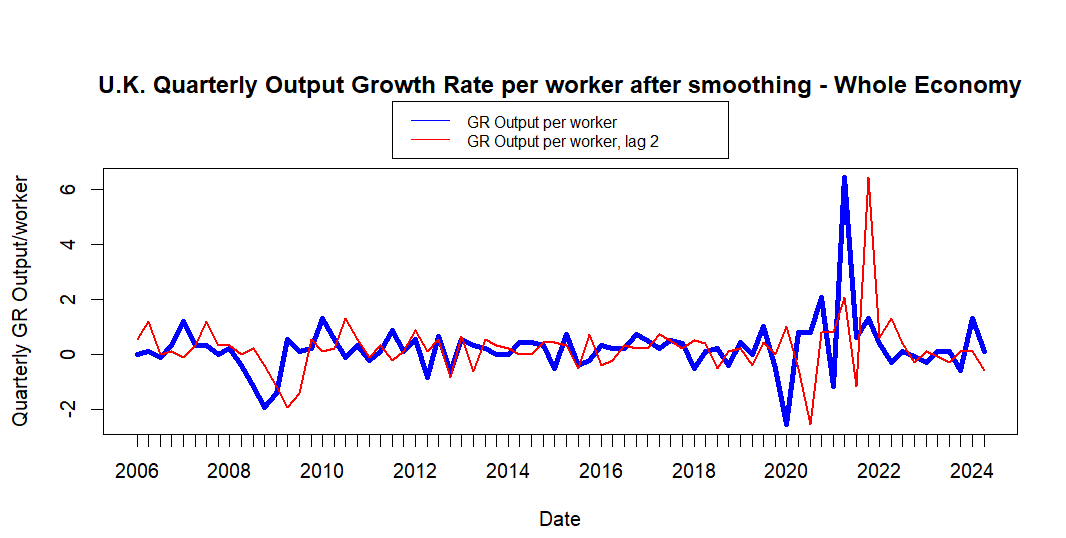
The economic intuition is that low unemployment rates would cause an increase in the output per worker in the next period and vice versa. Therefore, we expect a negative relationship between *Xt* and *Yt*.

Null and alternative hypothesis:

* 1. Time Series Plots



A trend is visible. It has two shocks, in 2008 and 2020, which have impacted the trend downwardly. There are large outliers in Quarter 2 and 3 2020. They will be smoothed out by replacing them with the average of the Growth rate one year prior and after the outliers.



There are big fluctuations between 2020 and 2022. A brake may be detected there.

2. Autoregression Analysis of a Time Series

2.1 Estimate an Autoregression Model:

AR (1) Output Growth model:

(Dyer, 2021, p. 3)

Select the correct p lag with Bayes Information Criterion.

BIC values with their related p:

|  |  |
| --- | --- |
| p | BIC |
| 1 | -0.1776 |
| 2 | -0.1878 |
| 3 | -0.1426 |
| 4 | -0.0899 |

The AR (2) model has the lowest BIC. The lowest value of BIC is used because it gives the estimate of the true p.

AR (2) Output Growth model:

Test for Violations of key Time Series Assumptions:

Test for a unit root process:

Theoretically, unit root processes are not intrinsic to most macroeconomic time series and variations are stationary around a deterministic trend function (Perron, 1989, p. 1361).

This means that the mean and variance should be constant over time. This means the model is stationary. If the model performs better when adding a trend, it means that it is stationary around a trend.

It is expected that the null hypothesis will be rejected.

Augmented Dickey Fuller test: the time series may have a trend, beta 0 will be included.

(Dyer, 2021, p. 3)

Null and alternative hypothesis:

*H0: = = 0*

*H1: = < 1*

The ADF test statistic is -4.98, lower than tau2 critical values in Table 2. It is significant, and the null hypothesis is rejected at 1% significance level. The time series is stationary. A trend test will not be conducted.

|  |  |  |  |
| --- | --- | --- | --- |
| **ADF test without trend** | | | |
|  | | | |
|  | 1pct | 5pct | 10pct |
|  | | | |
| tau2 | -3.460 | -2.880 | -2.570 |

Test for a break:

Breaks violate stationarity. Failure to address breaks will result in applying the same parameters to the whole time series, estimating misleading coefficients.

Chow test model with QLR Max break point:

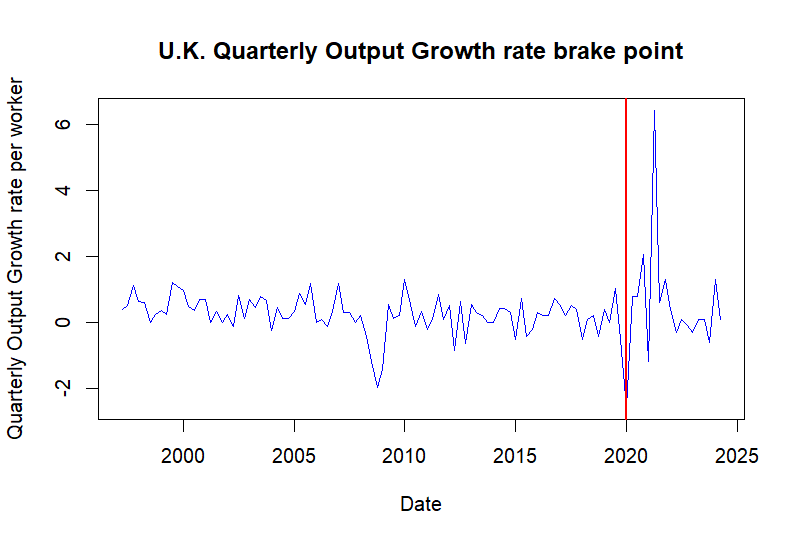
(Dyer, 2021, p. 3)

(Dyer, 2021, p. 3)

Null and alternative hypothesis:

*H0: = = 0*

*H1: = ≠ 0*



The break point is in Quarter 1, 2020. The F-test with two restrictions is 6.21. This is higher than 5.86 critical value at 5% significance level. The null hypothesis is rejected. There is enough evidence to conclude that a brake occurred.

Estimate a model with the break to absorb the effects it has on the target estimator. The brake is very likely due to the Covid-19 negative impact on the UK economy (ONS, 2021, part 1).

Report estimated coefficients:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AR (1), AR (2), AR (2) with drift and AR (2) with brake Coefficient Table** | | | | |
|  | | | | |
|  | *Dependent variable:* | | | |
|  |  | | | |
|  | Output | | diff(Output) | Output |
|  | (1) | (2) | (3) | (4) |
|  | | | | |
| Output lag 1 (1, 3, 4) | -0.004 |  | -0.770\*\*\* | 0.298\* |
|  | (0.097) |  | (0.155) | (0.163) |
|  |  |  |  |  |
| Output lag 1 (2) |  | -0.004 |  |  |
|  |  | (0.095) |  |  |
|  |  |  |  |  |
| Output lag 2 (2) |  | 0.249\*\* |  |  |
|  |  | (0.096) |  |  |
|  |  |  |  |  |
| Diff(Output) lag 1 (3) |  |  | -0.231\* |  |
|  |  |  | (0.138) |  |
|  |  |  |  |  |
| Diff(Output) lag 2 (3) |  |  | 0.017 |  |
|  |  |  | (0.100) |  |
|  |  |  |  |  |
| D (4) |  |  |  | 0.607\*\* |
|  |  |  |  | (0.234) |
|  |  |  |  |  |
| Output lag 2 (4) |  |  |  | 0.192\*\* |
|  |  |  |  | (0.094) |
|  |  |  |  |  |
| Output lag 1 \* D | (4) |  |  | -0.482\*\* |
|  |  |  |  | (0.200) |
|  |  |  |  |  |
| Constant (1, 2, 3, 4) | 0.296\*\*\* | 0.222\*\* | 0.219\*\* | 0.093 |
|  | (0.090) | (0.092) | (0.096) | (0.099) |
|  |  |  |  |  |
|  | | | | |
| Observations | 108 | 107 | 106 | 107 |
| R2 | 0.00002 | 0.061 | 0.536 | 0.142 |
| Adjusted R2 | -0.009 | 0.043 | 0.522 | 0.108 |
| Residual Std. Error | 0.884 (df = 106) | 0.865 (df = 104) | 0.869 (df = 102) | 0.835 (df = 102) |
| F Statistic | 0.002 (df = 1; 106) | 3.396\*\* (df = 2; 104) | 39.235\*\*\* (df = 3; 102) | 4.220\*\*\* (df = 4; 102) |
|  | | | | |
| *Note:* | \*p\*\*p\*\*\*p<0.01 | | | |

AR (1):

Non-significant. The smoothing process of outliers decreased autocorrelation.

AR (2):

The second lag coefficient is significant.

Lower standard error as the additional data in the second lag enhanced model fitness.

AR (1) with drift:

Stationarity present before taking the first differences: = + – 1 (-0.770 – 0.231 – 1 ≈ 0).

This means UK economy is resilient as shocks do not persist.

AR (2) with brake:

All coefficients are significant except the constant.

This model has the lowest standard errors. The model has good fit. It may also be caused by too many variables.

Conclusion:

The most reliable model is AR (2) with brake. This provides robustness in terms of significance, errors size and fitness.

2.2 Estimate an Autoregressive Distributed Lag Model

Estimate ADL (1, 1) with break:

Select the correct ADL (p, p) with BIC.

|  |  |
| --- | --- |
| p | BIC |
| 1 | -0.2069 |
| 2 | -0.1582 |
| 3 | -0.1297 |
| 4 | -0.0805 |

The AR (1, 1) model has the lowest BIC.

Granger Causality test:

ADL (1, 1) F-test statistic with two restrictions is 4.7906. The null hypothesis is rejected at 1% significance level that the coefficient on the UneRate is significant.

ADL (2, 2) F-test statistic with three restrictions is 4.3499. The null hypothesis is rejected that the model with lag 1 and 2 of UneRate and the Brake interaction with UneRate is significant.

|  |  |  |
| --- | --- | --- |
| **ADL (1,1) and ADL (2,2) Coefficient Table** | | |
|  | | |
|  | *Dependent variable:* | |
|  |  | |
|  | Output | |
|  | (1) | (2) |
|  | | |
| Output lag 1 (1, 2) | 0.289\* | 0.287\* |
|  | (0.162) | (0.161) |
|  |  |  |
| Output lag 2 (2) |  | 0.108 |
|  |  | (0.094) |
|  |  |  |
| D (1,2) | -2.872\* | -3.244\*\* |
|  | (1.622) | (1.622) |
|  |  |  |
| Une\_rate lag 1 (1, 2) | 0.082 | -0.185 |
|  | (0.053) | (0.132) |
|  |  |  |
| Une\_rate lag 2 (2) |  | 0.286\*\* |
|  |  | (0.132) |
|  |  |  |
| Output lag 1\* D (1, 2) | -0.563\*\*\* | -0.586\*\*\* |
|  | (0.197) | (0.201) |
|  |  |  |
| Une\_rate lag 1 \* D (1, 2) | 0.631\*\* | 0.711\*\* |
|  | (0.271) | (0.274) |
|  |  |  |
| Constant (1, 2) | -0.500 | -0.695 |
|  | (0.420) | (0.425) |
|  |  |  |
|  | | |
| Observations | 108 | 107 |
| R2 | 0.184 | 0.242 |
| Adjusted R2 | 0.143 | 0.188 |
| Residual Std. Error | 0.815 (df = 102) | 0.797 (df = 99) |
| F Statistic | 4.585\*\*\* (df = 5; 102) | 4.514\*\*\* (df = 7; 99) |
|  | | |
| *Note:* | \*p\*\*p\*\*\*p<0.01 | |

UneRate’s coefficients are jointly predictive of the Output per worker.

The second lags do not improve the model significantly, the effect of UneRate on Output stays approximately the same, but the second lag is statistically significant. Adding more past information about the variable improves the model.

The interaction terms of the Brake with both variables did not change statistically.

Conclusion:

The Unemployment Rate of NON-UK nationals has a positive effect on the Output per worker predictions, as suggested by the coefficients in ADL (2,2) model. This contradicts the study by Pallero.

2.3 Check Out-of-Sample Forecast Performance:

The test has been conducted without the brake for the ADL (1,1) model mentioned above. Out-of-Sample test data: 2017 Q1 – 2024 Q2.

The Standard Error of the Regression is 0.534 and the Root Mean Squared Forecast Error is 1.476. The forecast errors are larger than the sample’s model errors. The t-test statistic is not significant. The null hypothesis is not rejected that the RMSFE is statistically different from SER. Forecasts are systematically wrong and can be inaccurate.

The forecasted period’s parameters have been affected by the shocks in 2020-2022 period. This is also illustrated in the graph below.

A graph with purple lines and numbers

AI-generated content may be incorrect.

References:

Dyer, J. (2021). BEEM012\_EmpiricalAssignment\_LatexTemplate [TeX Doc]. University of Exeter.

Exeter University. (n.d.). QLRtest.R. [R script]. University of Exeter.

Exeter University. (n.d.). Week1\_Exercises.R. [R script]. University of Exeter.

Exeter University. (n.d.). Week2\_Exercises.R. [R script]. University of Exeter.

Exeter University. (n.d.). Week4\_Exercises.R. [R script]. University of Exeter.

ONS. (2021). Coronavirus and the impact on output in the UK economy: January 2021. <https://www.ons.gov.uk/economy/grossdomesticproductgdp/articles/coronavirusandtheimpactonoutputintheukeconomy/january2021>

OpenAI. (2023). ChatGPT (ChatGPT-4, March 14, 2023) [AI Language Model]. <https://chat.openai.com/auth/login>

Parello, C. P. (2022). Migration and growth in a Schumpeterian growth model with creative destruction. *Oxford Economic Papers*, 74(4), Pages 1139–1166. <https://doi.org/10.1093/oep/gpab065>

Perron, P. (1989). The Great Crash, the Oil Price Shock, and the Unit Root Hypothesis. *Econometrica*, *57*(6), 1361–1401. https://doi.org/10.2307/1913712

Appendix – Code:

#Load Libraries

library(AER)

library(readxl)

library(xts)

library(zoo)

library(dynlm)

library(stargazer)

library(urca)

#1.2 Time Series Plots

#Load the UK Output per worker data:

UK\_output <- read\_excel(path = "UKOutputperworker.xls")

Une\_rateee <- read\_excel(path = "NON-UKnational-unemployment.xls")

#set the datetime format of the data frame

UK\_output$Date <- as.yearqtr(UK\_output$Date, format = "%Y Q%q")

Une\_rateee$Date <- as.yearqtr(Une\_rateee$Date, format = "%Y Q%q")

#convert the data to xts to make it compatible with time series analysis tools

UK\_output\_xts <- xts(UK\_output$Output, UK\_output$Date)

Une\_rate <- xts(Une\_rateee$Rate, Une\_rateee$Date)[-1]

#transform the series in a growth rate

UKGR\_Output <- xts(100 \* log(UK\_output\_xts/lag(UK\_output\_xts)))[-1]

#plot the data to understand the patterns

#plot the output

plot(as.zoo(UK\_output\_xts), col = "orange",

lwd = 2,

ylab = "Quarterly Output per worker",

xlab = "Date",

main = "U.K. Quarterly Output per worker - Whole Economy")

#large outliers, may violate the 3rd assumption of fourth moments.

#smooth out the shocks in the year 2020

print(UKGR\_Output["2019::2021"])

#calculate the average growth rate for 1 year prior and after the shocks.

average <- mean(c(

coredata(window(UKGR\_Output, start = as.yearqtr("2019 Q2"), end = as.yearqtr("2020 Q1"))),

coredata(window(UKGR\_Output, start = as.yearqtr("2020 Q4"), end = as.yearqtr("2021 Q4")))

)) #(OpenAI, 2023)

average

#now we want to replace the 2 quarters with this average

UKGR\_Output[as.yearqtr("2020 Q2")] <- average #(OpenAI, 2023)

UKGR\_Output[as.yearqtr("2020 Q3")] <- average

#plot the Output per worker growth rate and the second order lag

UKGR\_Output\_lag1 <- lag(UKGR\_Output, 2) #(Exeter, n.d., line 86-105)

par(mar=c(4.3, 4.3, 7, 2), xpd=TRUE)

plot(as.zoo(UKGR\_Output), col = "blue",

lwd = 4,

ylab = "Quarterly GR Output/worker",

xlab = "Date",

mgp = c(3, 1, 0),

mar= c(1,1,1,1),

width=5,

height=5,

main = "U.K. Quarterly Output Growth Rate per worker after smoothing - Whole Economy")

lines(as.zoo(UKGR\_Output\_lag1),

type = "l", lwd = 2,

col = "red")

legend("top",

c("GR Output per worker", "GR Output per worker, lag 2"),

col=c("blue", "red"),

lty=c(1,1),

inset=c(0,-0.25),

cex=0.8)

#Autoregression Analysis of the time series

#2.1 Estimate an autoregression model

#run an AR(1) of the Output per worker

UK\_output\_AR1 <- lm(UKGR\_Output ~ lag(UKGR\_Output,1))

coeftest(UK\_output\_AR1)

#get stargazer table

stargazer(UK\_output\_AR1,

digits = 3,

header = F,

type = "html",

title = "AR(1) Regression of Output Growth",

out = "AR(1)Regression")

#Run a Bayes Information Criterion of lag 4.

#define a function that computes the BIC for each lag.

BIC <- function(model) {

ssr <- sum(model$residuals^2) # sum of squared residuals

t <- length(model$residuals) # the length of residuals needed, this is the same as the number of observations

p <- length(model$coef) - 1 # substract 1 from the list of coefficients to exclude the intercept.

return(

round(c("p" = p,

"BIC" = log(ssr/t) + ((p+1)\*log(t)/t)),

4)

)

}

# loop through the function for different lags

for (p in 1:4) {

print(BIC(lm(UKGR\_Output ~ lag(UKGR\_Output,1:p))))

} #(Exeter, n.d., line 103-134)

#estimate the model with the lowest BIC

#estimate AR(2) as well for comparison.

UK\_output\_AR2 <- lm(UKGR\_Output ~ lag(UKGR\_Output,1:2))

coeftest(UK\_output\_AR2)

#get stargazer for both models

stargazer(UK\_output\_AR1, UK\_output\_AR2,

digits = 3,

header = F,

type = "html",

title = "AR(1) and AR(2) Regression of Output Growth",

out = "AR(1,2)Regression")

#Test for Key Time Series Assumptions:

#Conduct an Augumented Dickey Fuller test for AR(2) to test for a unit root process:

UK\_output\_AR1\_URtest <- ur.df(UKGR\_Output, lags = 2, type = "drift") #include drift, the intercept

ADF\_AR1\_valuetest <- UK\_output\_AR1\_URtest@teststat[1]

ADF\_AR1\_critvalues <- UK\_output\_AR1\_URtest@cval[1,]

print(c("ADF test statistic:", round(ADF\_AR1\_valuetest,2)))

print("ADF Critical values:")

print(ADF\_AR1\_critvalues)

#add a trend to the model as indicated by economic theory to see if it embetters the target estimate.

UK\_output\_AR1\_URtest\_T <- ur.df(UKGR\_Output, lags = 2, type = "trend")

ADFT\_AR1\_valuetest <- UK\_output\_AR1\_URtest\_T@teststat[1]

ADFT\_AR1\_critvalues <- UK\_output\_AR1\_URtest\_T@cval[1,]

print(c("ADF with trend test statistic:", round(ADFT\_AR1\_valuetest,2)))

print("ADF with trend Critical values:")

print(ADFT\_AR1\_critvalues)

#QLR test

num\_periods <- length(UKGR\_Output)

tau\_zero = round(0.25\*num\_periods,digits=0)

tau\_one = round((1-0.05)\*num\_periods,digits=0)

n\_tests <- tau\_one - tau\_zero + 1

tau <- seq(tau\_zero, tau\_one) (Exter, line)

#set a binary variable and run a Chow test

D <- 1\*(time(UKGR\_Output) > time(UKGR\_Output)[tau[1]])

Chow\_test = lm(UKGR\_Output ~ lag(UKGR\_Output,1) + D + (D\*lag(UKGR\_Output,1)))

coeftest(Chow\_test)

#array of chow test stats and run a for loop to compute a chow test for each tau

chow\_test\_stat <- array(n\_tests)

for (i in 1:n\_tests) {

D <- 1\*(time(UKGR\_Output) > time(UKGR\_Output)[tau[i]])

chow\_test\_AR1 = lm(UKGR\_Output ~ lag(UKGR\_Output,1) + D + (D\*lag(UKGR\_Output,1)))

chow\_test\_hyp = linearHypothesis(chow\_test\_AR1, c("D=0", "lag(UKGR\_Output, 1):D=0"), test="F", white.adjust = FALSE)

chow\_test\_stat[i] = chow\_test\_hyp$F[2]

}

data.frame("Level" = tau, "F-stat" = chow\_test\_stat)

#Get the max F-stat

QLR\_test = max(chow\_test\_stat)

cat("QLR Test Statistic: ", QLR\_test)

tau\_est <- tau[which.max(chow\_test\_stat)]

UKGR\_Output[tau\_est]

D <- 1\*(time(UKGR\_Output) > time(UKGR\_Output)[tau\_est])

chow\_test\_AR = lm(UKGR\_Output ~ lag(UKGR\_Output,1) + D + (D\*lag(UKGR\_Output,1)) + lag(UKGR\_Output,2))

coeftest(chow\_test\_AR)

#(Exeter, n.d., line 39-91)

# Finally, we can see what this looks like by plotting our data with a line at

# break period to see if the results look like what we see on the plot

plot(as.zoo(UKGR\_Output), col = "blue",

lwd = 1,

ylab = "Quarterly Output Growth rate per worker",

xlab = "Date",

main = "U.K. Quarterly Output Growth rate brake point")

abline(v=as.yearqtr("2020 Q1"), col = "red", lwd = 2)

#get coef table for UR test with drift and trend

UKGR\_URtest <- lm(diff(UKGR\_Output) ~ lag(UKGR\_Output,1) + lag(diff(UKGR\_Output),1:2))

coeftest(UKGR\_URtest)

UKGR\_URtestT <- lm(diff(UKGR\_Output) ~ time(UKGR\_Output) + lag(UKGR\_Output,1) + lag(diff(UKGR\_Output),1:2))

coeftest(UKGR\_URtestT)

#show the coefficient table

stargazer(UK\_output\_AR1, UK\_output\_AR2, UKGR\_URtest, chow\_test\_AR,

digits = 3,

header = F,

type = "html",

title = "AR (1), AR (2), AR (2) with drift, and AR (2) with brake Coefficient Table",

out = "AR1 Complete Coeff table")

stargazer(UKGR\_URtestT,

digits = 3,

header = F,

type = "html",

title = "ADF test with trend Coefficient Table",

out = "ADF test with trend coeftable")

#estimate an ADl(1,1) model

ADL\_OutRate <- lm(UKGR\_Output ~ lag(UKGR\_Output,1) + D + (D\*lag(UKGR\_Output,1)) + lag(Une\_rate,1) + (D\*lag(Une\_rate,1)))

coeftest(ADL\_OutRate)

#estimate ADL(2,2)

ADL2\_OutRate <- lm(UKGR\_Output ~ lag(UKGR\_Output,1) + lag(UKGR\_Output,2) + D + (D\*lag(UKGR\_Output,1)) + lag(Une\_rate,1) + lag(Une\_rate,2) + (D\*lag(Une\_rate,1)))

coeftest(ADL2\_OutRate)

#use BIC to select p length

#Run a Bayes Information Criterion of lag 4.

#define a function that computes the BIC for each lag.

BIC <- function(model) {

ssr <- sum(model$residuals^2) # sum of squared residuals

t <- length(model$residuals) # the length of residuals needed, this is the same as the number of observations

p <- length(model$coef) - 1 # substract 1 from the list of coefficients to exclude the intercept.

return(

round(c("p" = p,

"BIC" = log(ssr/t) + ((p+1)\*log(t)/t)),

4)

)

}

# loop through the function for different lags

for (p in 1:4) {

print(BIC(lm(UKGR\_Output ~ lag(UKGR\_Output,1:p) + D + (D\*lag(UKGR\_Output,1:p)) + lag(Une\_rate,1:p) + (D\*lag(Une\_rate,1:p)))))

}

#conduct Grange Caussality test for ADL(1,1)

linearHypothesis(ADL\_OutRate, c("lag(Une\_rate, 1)=0", "D:lag(Une\_rate, 1)=0"), test ="F", white.adjust=FALSE)

#conduct Grange Causality test for ADL(2,2)

linearHypothesis(ADL2\_OutRate, c("lag(Une\_rate, 1)=0", "lag(Une\_rate, 2)=0", "D:lag(Une\_rate, 1)=0"), test ="F", white.adjust=FALSE)

#ADL models coef table

stargazer(ADL\_OutRate, ADL2\_OutRate,

digits = 3,

header = F,

type = "html",

title = "ADL (1,1) and ADL (2,2) Coefficient Table",

out = "ADL coef table")

length(UKGR\_Output)\*0.25

#109 observations, 25% is 27.25 observations. Increase this to 30 observations for enough forecasts.

#4 quarters in a year. 30/4 = 7.5 years. Select the last 7.5 years from the sample

length(window(UKGR\_Output, start = as.yearqtr("2017 Q1"), end = as.yearqtr("2024 Q2")))

#out-of-sample forecast performance

sample\_end <- seq(2017.00, 2024.25, 0.25)

p <- length(sample\_end)

forecasts <- array(c(0),dim = c(p))

true\_outputs <- array(c(0),dim = c(p))

fore\_errors <- array(c(0),dim = c(p))

ser <- array(c(0),dim = c(p))

for (i in 1:p){

Sample\_back = as.yearqtr(sample\_end[i])

UKGR\_Output\_poos = UKGR\_Output[index(UKGR\_Output) < Sample\_back]

Une\_rate\_poos = Une\_rate[index(Une\_rate) < Sample\_back]

#model the data before 2017

ADL1\_model\_poos <- lm(UKGR\_Output\_poos ~ lag(UKGR\_Output\_poos,1) + lag(Une\_rate\_poos,1))

#extract information

ser[i] <- summary(ADL1\_model\_poos)$sigma

beta0\_hat = ADL1\_model\_poos$coefficients[1]

beta1\_hat = ADL1\_model\_poos$coefficients[2]

delta1\_hat = ADL1\_model\_poos$coefficients[3]

true\_output <- UKGR\_Output[Sample\_back]

forecast <- (beta0\_hat + (beta1\_hat %\*% UKGR\_Output[Sample\_back - 0.25]) + (delta1\_hat %\*% Une\_rate[Sample\_back - 0.25]))

fore\_error <- true\_output - forecast

true\_outputs[i] <- true\_output

forecasts [i] <- forecast

fore\_errors [i] <- fore\_error

}

#Analysis of POOS:

ser\_withins <- ser[1]

estimated\_RMSFE <- sd(fore\_errors)

#print answer

cat("Within-Sample error:", ser\_withins, "\n")

cat ("Estimated Forecasted Error:", estimated\_RMSFE, "\n")

#test the significance

t.test(fore\_errors)

#plot the difference

true\_output\_plt <- xts(true\_outputs, as.yearqtr(sample\_end))

forecasts\_plt <- xts(forecasts, as.yearqtr(sample\_end))

plot(as.zoo(true\_output\_plt),

col = "purple",

lwd = 4,

ylab = "Percent",

main = "AR (1,1) Pseudo Out-of-Sample Forecats of Output per worker Growth Rate")

# add the series of pseudo-out-of-sample forecasts

lines(as.zoo(forecasts\_plt),

lwd = 4,

lty = 2)

# shade area between curves (the pseudo forecast error)

polygon(x= c(time(true\_output\_plt), rev(time(forecasts\_plt))),

y= c(true\_outputs, rev(forecasts)),

col = "grey85",

border = NA)

# add a legend

legend("bottomright",

lty = c(1, 2, 1),

lwd = c(2, 2, 10),

inset=c(0,0),

cex = 0.8,

col = c("purple", "black", "grey85"),

legend = c("Actual GDP growth rate",

"Forecasted GDP growth rate",

"Pseudo forecast Error"))

#(Exeter, n.d., line 51-147)

#draft for the model with brake, I could not find a way to iterate through it with a brake.

#this was my best try.

#the issue is cause by multicollinearity as Dd - the dummy variable

#makes the interaction coefficients multicollinear, producing NAs.

#troubleshooted with ChatGPT(2023).

sample\_end <- seq(2017.00, 2024.25, 0.25)

p <- length(sample\_end)

forecasts <- array(c(0),dim = c(p))

true\_outputs <- array(c(0),dim = c(p))

fore\_errors <- array(c(0),dim = c(p))

ser <- array(c(0),dim = c(p))

for (i in 1:p){

Sample\_back = as.yearqtr(sample\_end1[i])

UKGR\_Output\_poos = UKGR\_Output[index(UKGR\_Output) < Sample\_back]

Une\_rate\_poos = Une\_rate[index(Une\_rate) < Sample\_back]

Dd <- 1\*(index(UKGR\_Output\_poos) >= as.yearqtr("2020 Q1")) #(ChatGPT, 2023)

#model the data before 2017

ADL1\_model\_poos <- lm(UKGR\_Output\_poos ~ lag(UKGR\_Output\_poos,1) + lag(Une\_rate\_poos,1) + Dd + (Dd\*lag(UKGR\_Output\_poos,1)) + (Dd\*lag(Une\_rate\_poos,1)))

#extract information

ser[i] <- summary(ADL1\_model\_poos)$sigma

beta0\_hat = ADL1\_model\_poos$coefficients[1]

beta1\_hat = ADL1\_model\_poos$coefficients[2]

delta1\_hat = ADL1\_model\_poos$coefficients[3]

gamma0\_hat = ADL1\_model\_poos$coefficients[4]

gamma1\_hat = ADL1\_model\_poos$coefficients[5]

gamma2\_hat = ADL1\_model\_poos$coefficients[6]

true\_output <- UKGR\_Output[Sample\_back]

forecast <- (beta0\_hat + (beta1\_hat %\*% UKGR\_Output[Sample\_back - 0.25]) +

(delta1\_hat %\*% Une\_rate[Sample\_back - 0.25]) + gamma0\_hat + (gamma1\_hat %\*% (Dd \* UKGR\_Output[Sample\_back - 0.25])) +

(gamma2\_hat %\*% (D \* Une\_rate[Sample\_back - 0.25])))

fore\_error <- true\_output - forecast

true\_outputs[i] <- true\_output

forecasts [i] <- forecast

fore\_errors [i] <- fore\_error

}