Econometrics III
Assignment Part 1 & 2
Tinbergen Insitute

Stanislav Avdeev
590050sa
stnavdeev@gmail.com

Bas Machielsen
590049bm
590049bm@eur.nl

March 12, 2021

Question 1

Part (a)

```r
# Part 1: Plot the Dutch GDP, ACF, and PACF
df <- readr::read_csv("./data/data_assign_p1.csv")

df <- ts(df$GDP_QGR,
         frequency = 4,
         start = c(1987, 2))
```

In the above figure, we observe the growth rates of Dutch GDP from 1987 to 2009.

```r
autoc <- acf(df,
    lag.max = 12, plot = F)

Box.test(df,
         lag = 12,
         type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  df
## X-squared = 12.106, df = 12, p-value = 0.4372
```

```r
pautoc <- pacf(df,
     lag.max = 12, plot = F)
```

In the above figure, we plot the ACF and PACF, which is the ACF controlled for the other lagged correlations. The ACF tells us that the correlation of GDP growth with its lags is very low - hinting at very little time-dependence in this time-series. More precisely, the estimated correlation coefficients are not higher than 0.2 (for the lag of 1 period). If the GDP is indeed generated by an AR(p) process, the estimates show that the process has low ϕ's (in absolute value), indicating a low time dependence.

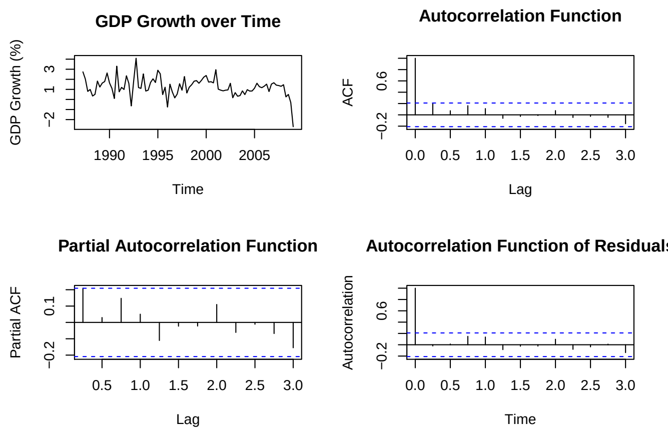
Part (b)

```r
ar4 <- dynlm(df ~ L(df, 1) + L(df, 2) + L(df, 3) + L(df, 4))
ar3 <- dynlm(df ~ L(df, 1) + L(df, 3) + L(df, 4))
ar2 <- dynlm(df ~ L(df, 1) + L(df, 3))
ar1 <- dynlm(df ~ L(df, 1))

ar1_m <- forecast::Arima(df, c(1,0,0), method = "CSS")

stargazer(ar4, ar3, ar2, ar1,
          type = "latex",
          header = FALSE,
          font.size = "small")
```

Table 1:

	Dependent variable:			
	df			
	(1)	(2)	(3)	(4)
L(df, 1)	0.232*	0.240*	0.257**	0.267**
	(0.124)	(0.122)	(0.119)	(0.117)
L(df, 2)	0.055			
	(0.126)			
L(df, 3)	0.203	0.210*	0.210*	
	(0.126)	(0.124)	(0.120)	
L(df, 4)	0.094	0.092		
	(0.125)	(0.124)		
Constant	0.479	0.533*	0.631***	0.896***
	(0.299)	(0.271)	(0.238)	(0.181)
Observations	84	84	85	87
R^2	0.099	0.097	0.089	0.058
Adjusted R^2	0.054	0.063	0.067	0.047
Residual Std. Error	0.902 (df = 79)	0.898 (df = 80)	0.891 (df = 82)	0.895 (df = 85)
F Statistic	2.181* (df = 4; 79)	2.873** (df = 3; 80)	4.019** (df = 2; 82)	5.229** (df = 1; 85)
Note:				*p<0.1; **p<0.05; ***p<0.01

The table above shows us the coefficients of the $AR(p)$ models with 1 to 4 lags included. Using a general-to-specific approach, we eliminate insignificant lags ($\alpha = 0.05$ level) at each step. Column (1) shows that only the first lag is significant at the 0.1 level. The second lag has the lowest ψ coefficient and the highest s.e., thus, we eliminate it first. Doing this iteratively, we are left with the first lag in the model which is significant at the 0.05 level. Investigating the coefficient, we observe there is a 1-period persistence of the time series on its lagged value. The *degree* of persistence, however, is not large, as the magnitude of the coefficient is fairly small (0.26).



Part c

```r
# Part 3: Plot ACF of residuals
autocred <- acf(ar1_m$resid, 12,
                plot = FALSE)

durbinWatsonTest(ar1)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1     -0.02081405       1.85759   0.548
##  Alternative hypothesis: rho != 0
```

In the fourth plot above, we show the autocorrelation of the residuals. We observe that none of the coefficients attain significance. Hence, we think that the model is well-specified. Formally, we compute the Durbin-Watson test statistic, of which the output is shown above. The null hypothesis of no autocorrelation is not rejected. In an unreported plot, we also investigate the *partial* autocorrelation of residuals, in which there is also no sign of autocorrelation.

Part d

```r
# Part 4: Forecast AR model for 2 years
df_pred <- predict(ar1_m, n.ahead = 8)$pred
```

We derived the forecasts by making use of the conditional expectation as the forecast minimizing the forecast error under quadratic loss. The conditional expectation of the forecast for $T+1$ is: $\mathbb{E}[X_{T+1}|X_1,\dots,X_T] = \mathbb{E}[\phi X_T + \alpha + \epsilon_{T+1}|X_1,\dots,X_T] = \phi X_T + \alpha$. Similarly, the forecast for $T+2$ equals $\phi\mathbb{E}[X_{T+1}+\alpha] = \phi(\alpha + \phi X_T) + \alpha$. Generalizing this pattern, we then end up with the following recursive forecasts (for $j > 1$):

$$\mathbb{E}[X_{T+j}] = \alpha + \alpha \cdot \left(\sum_{i=1}^{J-1} \phi^i\right) + \phi^j X^T$$

Part e

```r
# Part 5: Produce CI
df_ciu <- predict(ar1_m, n.ahead = 8)$pred + predict(ar1_m, n.ahead = 8)$se*1.96
df_cil <- predict(ar1_m, n.ahead = 8)$pred - predict(ar1_m, n.ahead = 8)$se*1.96
```

Part f

```r
# Part 6: Check normality
jb.norm.test(ar1_m$resid)
```

```
##
##  Jarque-Bera test for normality
##
## data:  ar1_m$resid
## JB = 31.722, p-value = 0.0015
```

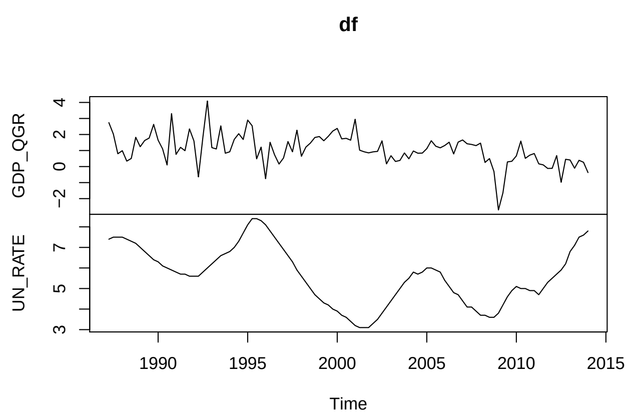
```r
# reject H0, innovations are not normally distributed
```

Question 2

```r
# Get the data
df <- readr::read_csv("./data/data_assign_p2.csv")

df <- ts(df[,2:3],
         frequency = 4,
         start = c(1987, 2))
```

Part (a)

```r
# Part a: Plot the Dutch GDP/unemployment, AR and ADL
plot(df)
```




```r
ar4 <- dynlm(df[,1] ~ L(df[,1], 1) + L(df[,1], 2) + L(df[,1], 3) + L(df[,1], 4))
ar3 <- dynlm(df[,1] ~ L(df[,1], 1) + L(df[,1], 3) + L(df[,1], 4))
ar2 <- dynlm(df[,1] ~ L(df[,1], 1) + L(df[,1], 3))

adl4 <- dynlm(df[,1] ~ L(df[,1], 1) + L(df[,1], 2) + L(df[,1], 3) + L(df[,1], 4) + L(df[,2], 1) + L(df[
adl3 <- dynlm(df[,1] ~ L(df[,1], 1) + L(df[,1], 3) + L(df[,1], 4))
adl2 <- dynlm(df[,1] ~ L(df[,1], 1) + L(df[,1], 3))

stargazer(adl4, adl3, adl2, type = "text", header = FALSE)
```

```
##
## ========================================================================================
##                                              Dependent variable:
##                        -----------------------------------------------------------------
##                                                    df[, 1]
##                               (1)                    (2)                     (3)
## ----------------------------------------------------------------------------------------
## L(df[, 1], 1)              0.305***               0.342***                0.356***
##                            (0.101)                (0.095)                 (0.091)
##
## L(df[, 1], 2)               0.078
##                            (0.105)
##
## L(df[, 1], 3)              0.218**                0.234**                 0.244***
##                            (0.104)                (0.099)                 (0.090)
##
## L(df[, 1], 4)               0.010                  0.050
##                            (0.102)                (0.099)
##
## L(df[, 2], 1)               0.620
##                            (0.713)
##
## L(df[, 2], 2)              -0.088
##                            (1.326)
##
## L(df[, 2], 3)              -1.931
##                            (1.335)
##
## L(df[, 2], 4)              1.502**
##                            (0.717)
##
## Constant                   -0.184                 0.376**                 0.401***
##                            (0.398)                (0.153)                 (0.144)
##
## ----------------------------------------------------------------------------------------
## Observations                 104                    104                     105
## R2                          0.282                  0.237                   0.233
## Adjusted R2                 0.222                  0.214                   0.218
## Residual Std. Error    0.859 (df = 95)       0.863 (df = 100)        0.857 (df = 102)
## F Statistic         4.674*** (df = 8; 95) 10.327*** (df = 3; 100) 15.508*** (df = 2; 102)
## ========================================================================================
## Note:                                                         *p<0.1; **p<0.05; ***p<0.01
```