Econometrics III
Assignment Part 1 & 2
Tinbergen Insitute

Stanislav Avdeev
student no
stnavdeev@gmail.com

Bas Machielsen
590049bm
590049bm@eur.nl

March 9, 2021

Instructions

I created a file .Renviron, where the python distribution is located on my system. You can find that by opening a terminal, and entering `$ which -a python python3`. Then, in RStudio, use `usethis::edit_r_environ()`, and add `RETICULATE_PYTHON="/Users/basmachielsen/opt/anaconda3/bin/python"` (or your directory) on a new line to the file. In this way, we can seamlessly interchange R and Python code chunks. Restart RStudio, and then everything is ready to go:

Question 1

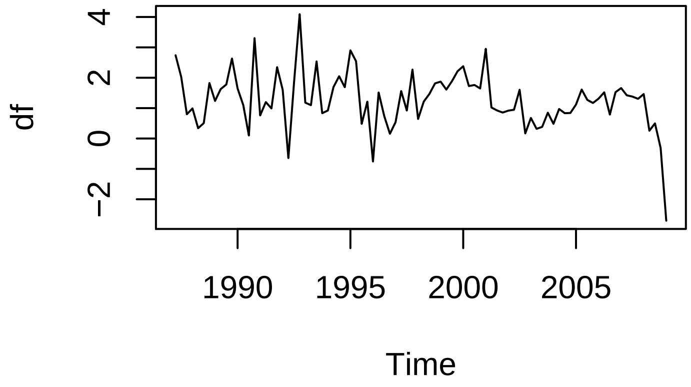
Part (a)

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

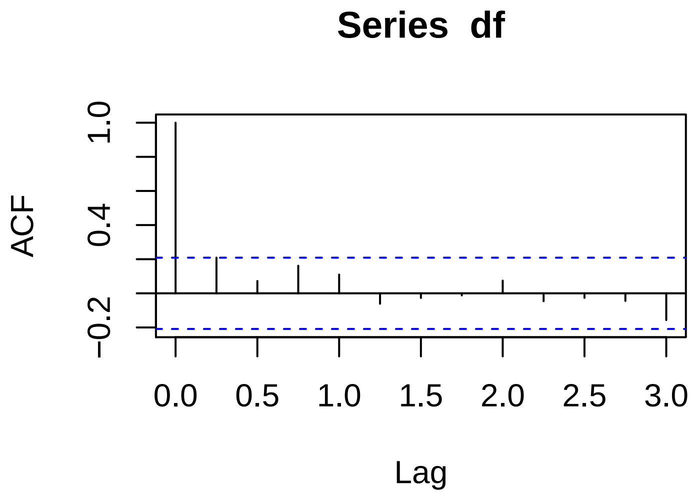
```
## Parsed with column specification:
## cols(
##   obs = col_character(),
##   GDP_QGR = col_double()
## )
```

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

plot(df)
```



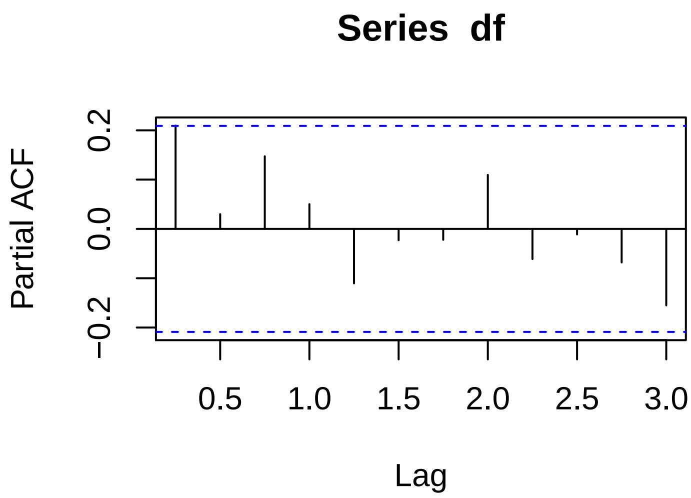
```r
acf(df, lag.max = 12)
```



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

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

```r
pacf(df, lag.max = 12)
```



Part (b)

```
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))
stargazer(ar4, ar3, ar2, ar1, type = "latex")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Tue, Mar 09, 2021 - 17:53:57

```
ar1_m <- arima(df, c(1, 0, 0))
```

Part c

```
# Part 3: Plot ACF of residuals
acf(ar1_m$resid, 12)
```

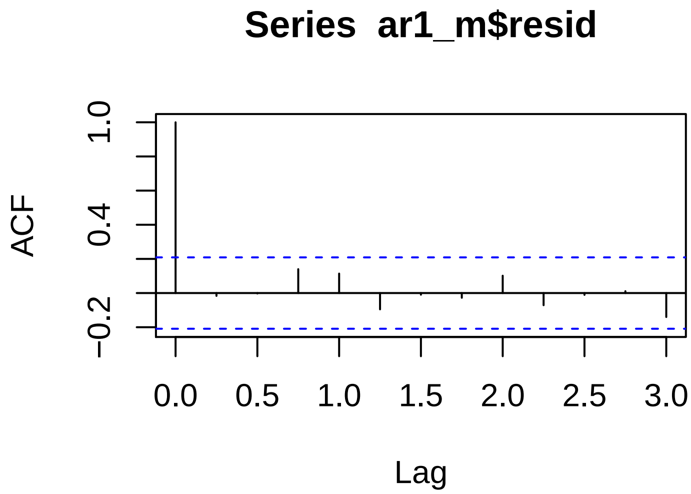


Table 1:

	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

Part d

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

Part e

```
# 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

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

```
##
##  Jarque-Bera test for normality
##
## data:  ar1_m$resid
## JB = 26.298, p-value = 0.002
```

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

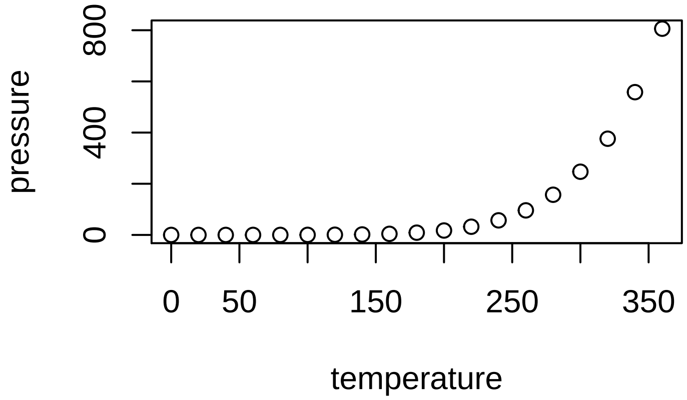
```
import pandas as pd
import numpy as np

r.mtcars.sum()
```

```
## mpg      642.900
## cyl      198.000
## disp    7383.100
## hp      4694.000
## drat     115.090
## wt       102.952
## qsec     571.160
## vs        14.000
## am        13.000
## gear     118.000
## carb      90.000
## dtype: float64
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Including Matplotlib

```
hoi = np.arange(0,10)

hoi
```

```
## array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])
```

```
#py$hoi
#gert::git_branch_checkout("attempt_bas")
#gert::git_add(c("*"))
#gert::git_commit(message = "Automatic commit")
#gert::git_push()
```