Week6Assignment

Pradeepta Das

18 November 2020

Let's look at the data.

```
head(data)
```

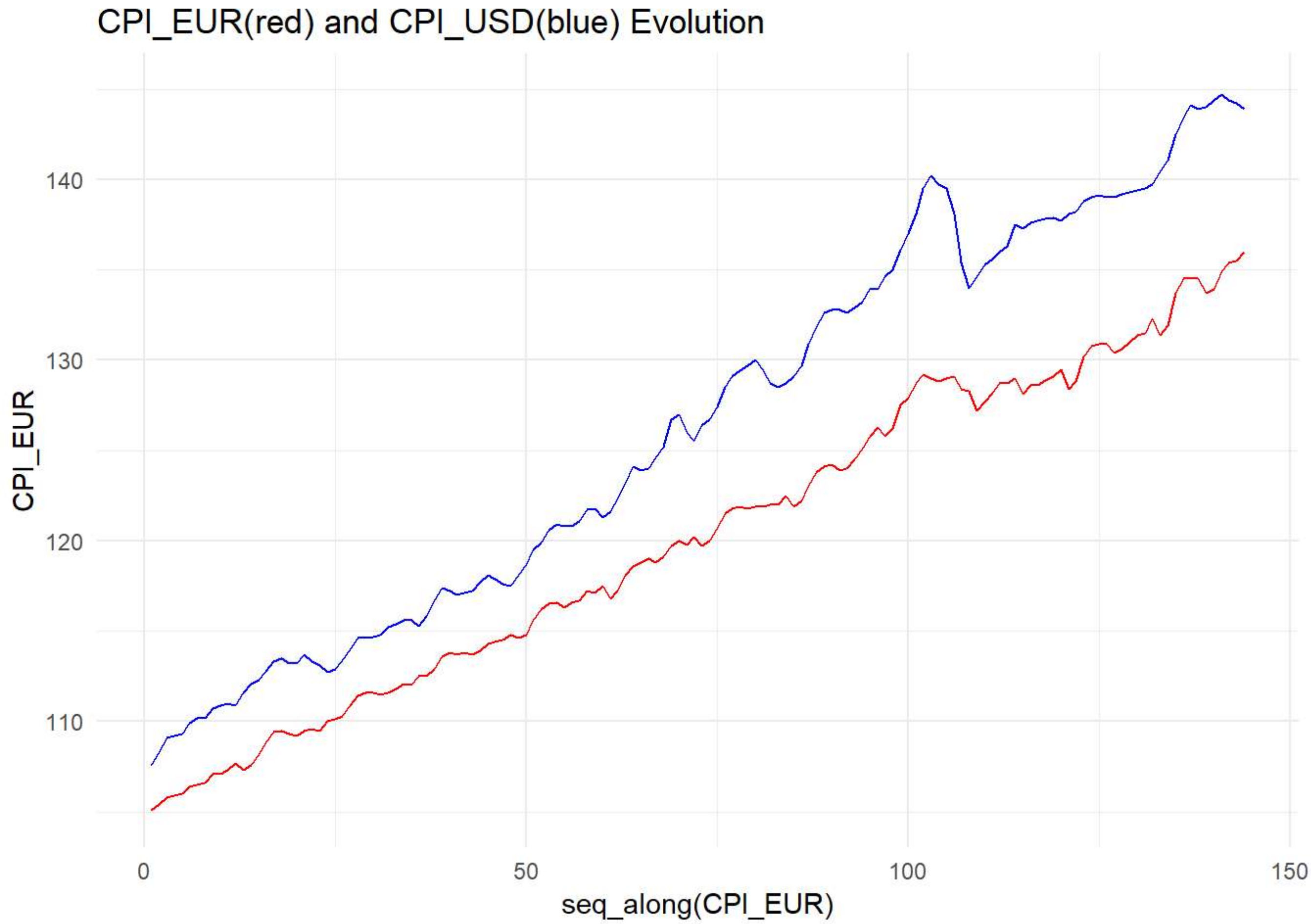
```
##   YYYY.MM TREND CPI_EUR CPI_USA  LOGPEUR  LOGPUSA    DPEUR    DPUSA
## 1 2000M01     1   105.1   107.6 4.654912 4.678421       NA       NA
## 2 2000M02     2   105.4   108.3 4.657763 4.684905 0.002850 0.006485
## 3 2000M03     3   105.8   109.1 4.661551 4.692265 0.003788 0.007360
## 4 2000M04     4   105.9   109.2 4.662495 4.693181 0.000945 0.000916
## 5 2000M05     5   106.0   109.3 4.663439 4.694096 0.000944 0.000915
## 6 2000M06     6   106.4   109.9 4.667206 4.699571 0.003766 0.005474
```

Questions This test exercise uses data that are available in the data file TestExer6. The question of interest is to model monthly inflation in the Euro area and to investigate whether inflation in the United States of America has predictive power for inflation in the Euro area. Monthly data on the consumer price index (CPI) for the Euro area and the USA are available from January 2000 until December 2011. The data for January 2000 until December 2010 are used for specification and estimation of models, and the data for 2011 are left out for forecast evaluation purposes.

(a) Make time series plots of the CPI of the Euro area and the USA, and also of their logarithm log(CPI) and of the two monthly inflation series DP = Δlog(CPI). What conclusions do you draw from these plots?

```
g1<- ggplot(aes(x= seq_along(CPI_EUR), y= CPI_EUR), data = data)+ geom_line(color= "red")+
  geom_line(y= data$CPI_USA, color= "blue") +
  ylim(c(105,145))+
  ggtitle("CPI_EUR(red) and CPI_USD(blue) Evolution")

print(g1)
```



The two consumer price index (CPI) plots indicate that:

USA and EURO prices seem to be correlated.

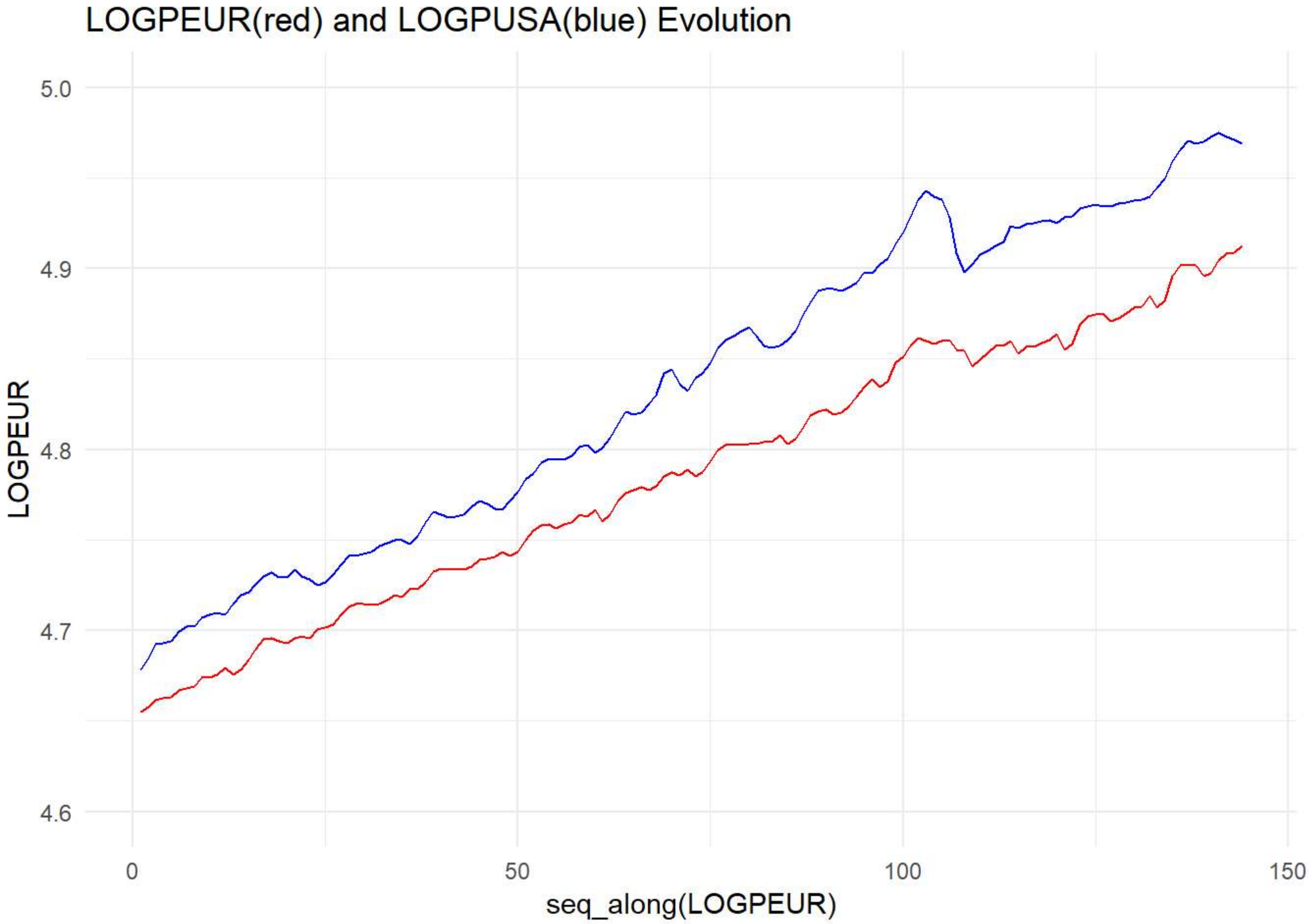
USA prices are typically higher than EURO prices, and the difference seems to slightly increase over time.

Both indexes are steadily increasing over time, with very few exceptions (i.e. year 2008).

The indexes increasing trend seems to be rather logarithmic than linear.

```
g2<- ggplot(aes(x= seq_along(LOGPEUR), y= LOGPEUR), data = data)+ geom_line(color= "red")+
  geom_line(y= data$LOGPUSA, color= "blue") +
  ylim(c(4.6,5))+
  ggtitle("LOGPEUR(red) and LOGPUSA(blue) Evolution")

print(g2)
```



The CPI logarithm log(CPI) plots indicate exactly the same things, and the corresponding diagram looks very much similar to the previous one.

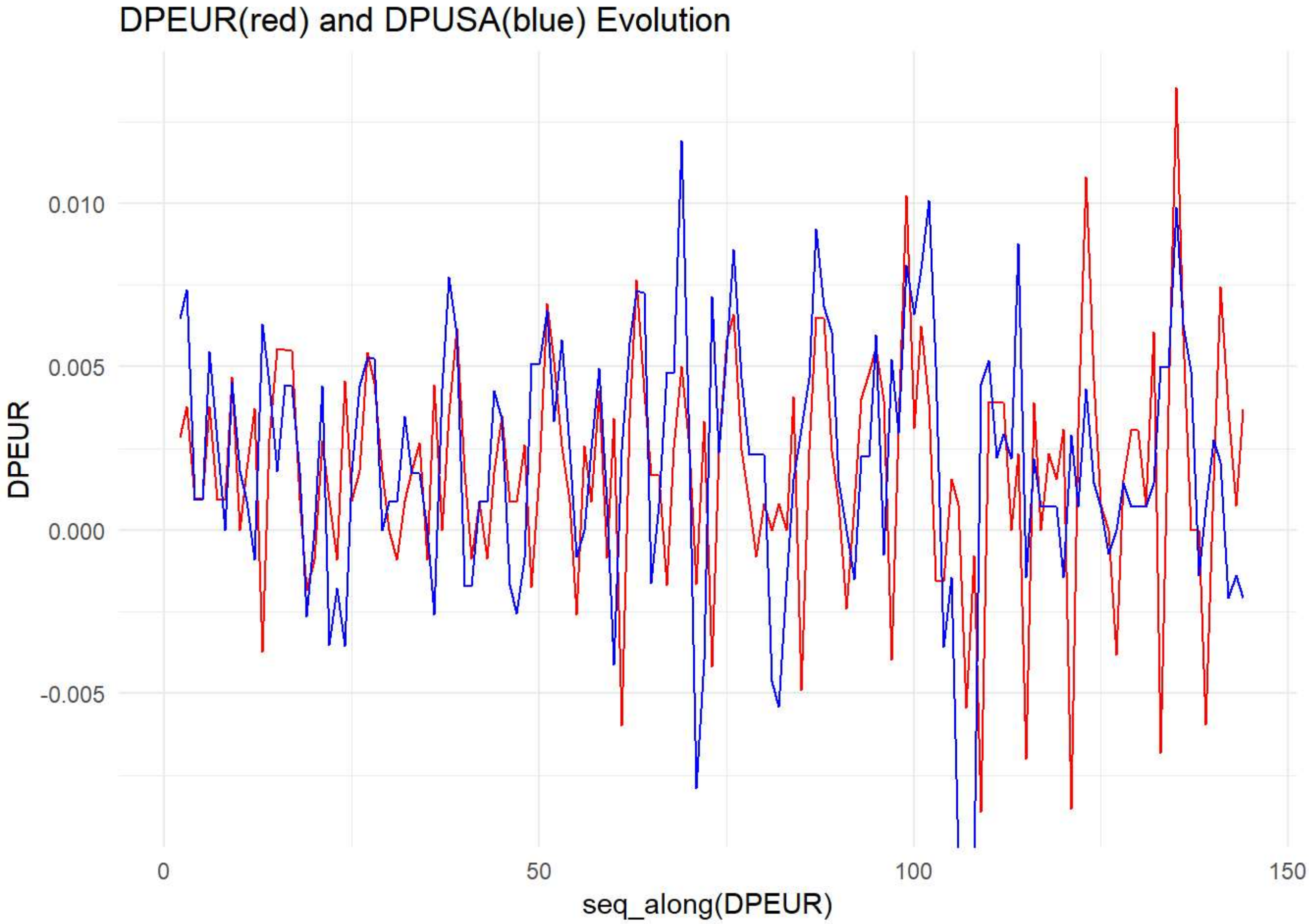
The apparently linear logarithmic increase confirms that the indexes increase over time is probably logarithmic rather than linear.

```
g3<- ggplot(aes(x= seq_along(DPEUR), y= DPEUR), data = data)+ geom_line(color= "red")+
  geom_line(y= data$DPUSA, color= "blue") +
  ggtitle("DPEUR(red) and DPUSA(blue) Evolution")

print(g3)
```

```
## Warning: Removed 1 rows containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_path).
```



(b) Perform the Augmented Dickey-Fuller (ADF) test for the two log(CPI) series. In the ADF test equation, include a constant (α), a deterministic trend term (β_t), three lags of DP = Δlog(CPI) and, of course, the variable of interest log(CPI_{t-1}). Report the coefficient of log(CPI_{t-1}) and its standard error and t-value, and draw your conclusion.

```
adf1 <- adf.test(data$LOGPEUR,'stationary',k=3)
adf1
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data$LOGPEUR
## Dickey-Fuller = -2.8263, Lag order = 3, p-value = 0.2324
## alternative hypothesis: stationary
```

This adf.test always includes constant term and linear trend parameter.

```
adf1 <- adf.test(data$LOGPUSA,'stationary',k=3)
adf1
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data$LOGPUSA
## Dickey-Fuller = -2.7345, Lag order = 3, p-value = 0.2706
## alternative hypothesis: stationary
```

Both the series are not stationary! The ADF statistic is greater than the critical value of −3.5.

(c) As the two series of log(CPI) are not cointegrated (you need not check this), we continue by modelling the monthly inflation series DPEUR=Δlog(CPIEUR) for the Euro area. Determine the sample autocorrelations and the sample partial autocorrelations of this series to motivate the use of the following AR model:

$$DPEUR_t = \alpha + \beta_1 DPEUR_{t-6} + \beta_2 DPEUR_{t-12} + \epsilon_t$$

. Estimate the parameters of this model (sample Jan 2000 - Dec 2010).

First divide the data:

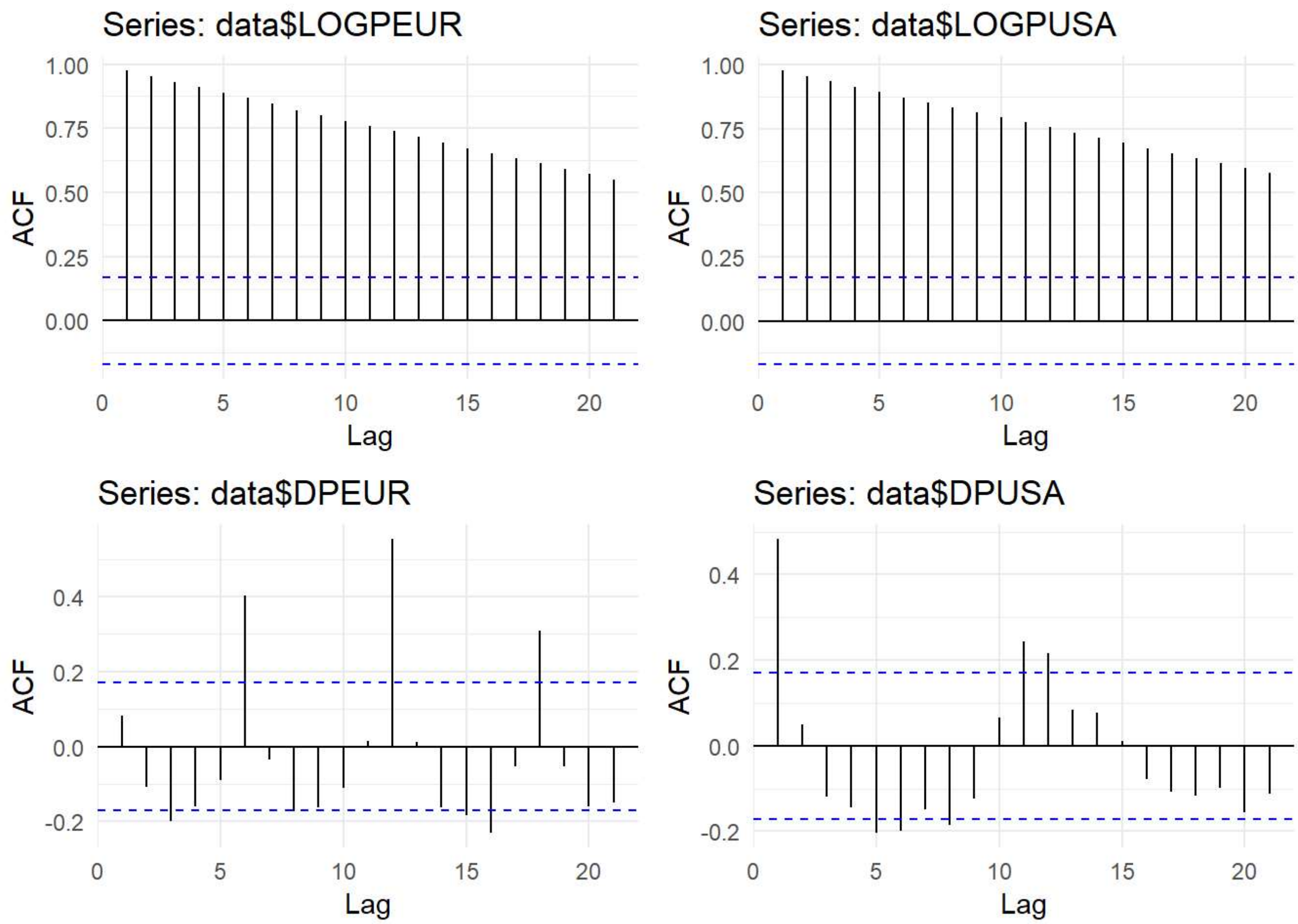
```
data_orig<- data.frame(data)
data[which(data$YYYY.MM == "2010M12"), ]
```

```
##     YYYY.MM TREND CPI_EUR CPI_USA  LOGPEUR  LOGPUSA    DPEUR    DPUSA
## 132 2010M12   132   132.3   139.7 4.885072 4.939497 0.006065 0.001433
```

```
data <- data[1:132,]
f_data <- data_orig[133:nrow(data_orig),]
```

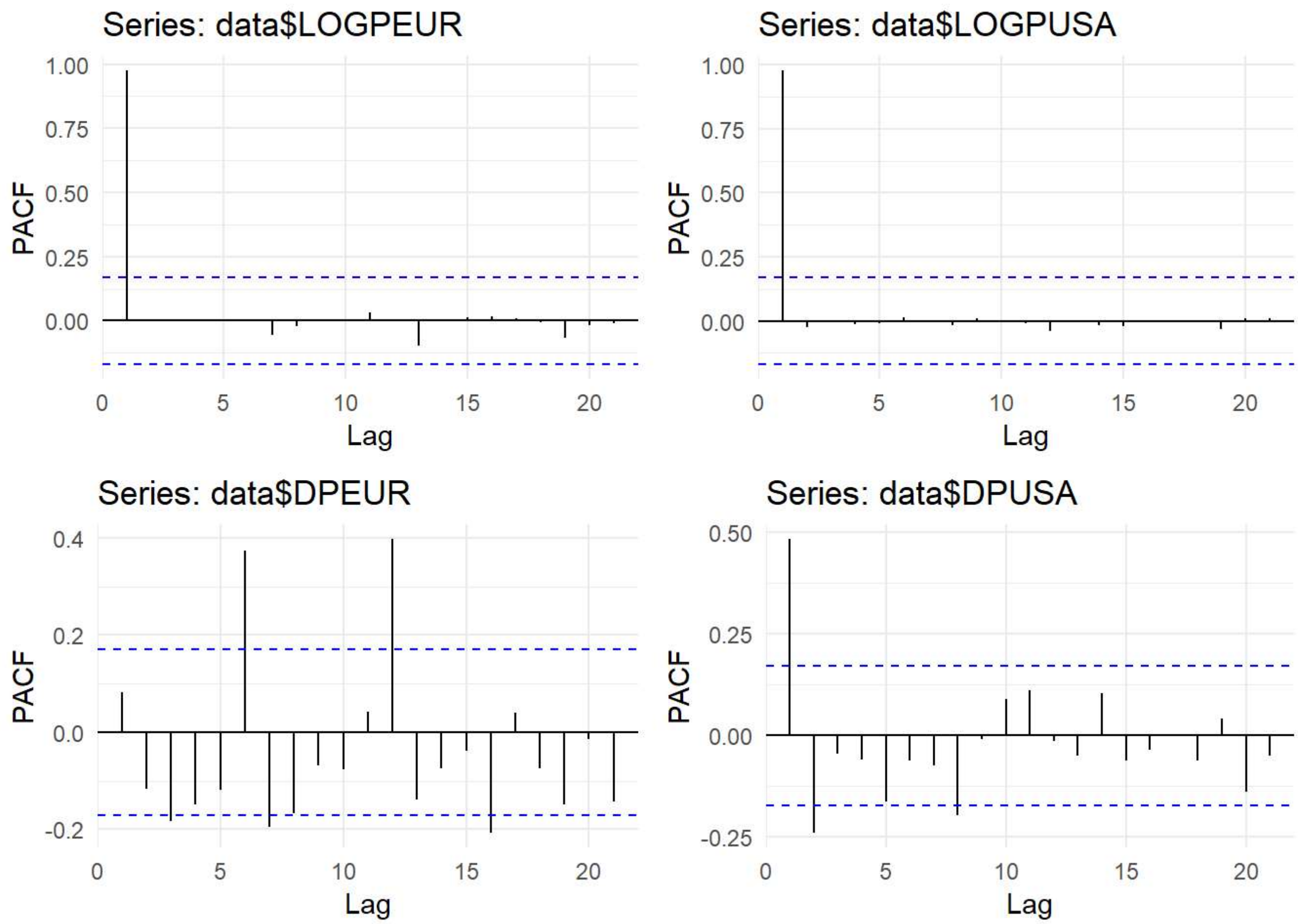
AutoCorrelations:

```
g1<-ggAcf(data$CPI_EUR, type="correlation")
g2<-ggAcf(data$CPI_USA, type="correlation")
g3<-ggAcf(data$LOGPEUR, type="correlation")
g4<-ggAcf(data$LOGPUSA, type="correlation")
g5<-ggAcf(data$DPEUR, type="correlation")
g6<-ggAcf(data$DPUSA, type="correlation")
grid.arrange(g3,g4,g5,g6)
```



Partial AutoCorrelations:

```
g1<-ggAcf(data$CPI_EUR, type="partial")
g2<-ggAcf(data$CPI_USA, type="partial")
g3<-ggAcf(data$LOGPEUR, type="partial")
g4<-ggAcf(data$LOGPUSA, type="partial")
g5<-ggAcf(data$DPEUR, type="partial")
g6<-ggAcf(data$DPUSA, type="partial")
grid.arrange(g3,g4,g5,g6)
```



The lags with the largest values found from above gaps, are 6 and 12:

Thus, the findings motivate using lags 6 and 12 with the AR model. And as this is not an MA model (residuals are not included), AR model could be estimated using the OLS method (instead of the MLE method).

Estimating the AR model for lags 6 and 12 and using OLS, produces the following Autoregressive Fit Model:

```
library(dynlm)
```

```
## Warning: package 'dynlm' was built under R version 3.6.3
```

```
ts_logeur <- ts(data$DPEUR, start =2)
ar.model <- dynlm(ts_logeur ~ L(ts_logeur, 6) + L(ts_logeur, 12))
summary(ar.model)
```

```
##
## Time series regression with "ts" data:
## Start = 15, End = 133
##
## Call:
## dynlm(formula = ts_logeur ~ L(ts_logeur, 6) + L(ts_logeur, 12))
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0103343 -0.0017369 -0.0000475  0.0015322  0.0080903
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.0003838  0.0002811   1.365   0.1749
## L(ts_logeur, 6)  0.1887459  0.0772888   2.442   0.0161 *
## L(ts_logeur, 12) 0.5979841  0.0835544   7.157 8.05e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002569 on 116 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4232, Adjusted R-squared:  0.4132
## F-statistic: 42.55 on 2 and 116 DF,  p-value: 1.381e-14
```

(d) Extend the AR model of part (c) by adding lagged values of monthly inflation in the USA at lags 1, 6, and 12. Check that the coefficient at lag 6 is not significant, and estimate the ADL model

```
train <- ts.intersect(ts(data$DPEUR), ts(data$DPUSA))
colnames(train) <- c("dpEur", "dpUsa")
model <- dynlm(dpEur ~ L(dpEur, 6) + L(dpEur, 12) + L(dpUsa, 1) + L(dpUsa, 6) + L(dpUsa, 12), da
ta=train)
summary(model)
```

```
##
## Time series regression with "ts" data:
## Start = 14, End = 132
##
## Call:
## dynlm(formula = dpEur ~ L(dpEur, 6) + L(dpEur, 12) + L(dpUsa,
##     1) + L(dpUsa, 6) + L(dpUsa, 12), data = train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0065866 -0.0016535 -0.0000118  0.0012630  0.0082682
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0004407  0.0002853   1.545    0.125
## L(dpEur, 6)   0.2029891  0.0785520   2.584    0.011 *
## L(dpEur, 12)  0.6367464  0.0874766   7.279 4.78e-11 ***
## L(dpUsa, 1)   0.2264287  0.0511286   4.429 2.20e-05 ***
## L(dpUsa, 6)  -0.0560565  0.0547645  -1.024    0.308
## L(dpUsa, 12) -0.2300418  0.0541695  -4.247 4.47e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002272 on 113 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.5602, Adjusted R-squared:  0.5408
## F-statistic: 28.79 on 5 and 113 DF,  p-value: < 2.2e-16
```

The coefficient $DPUSA_{t-6}$ is indeed not-significant. (value=−0.056, t−statistic=−1.024, p−value=0.308).

Simplified ADL model estimation

```
adl.model <- dynlm(dpEur ~ L(dpEur, 6) + L(dpEur, 12) + L(dpUsa, 1) + L(dpUsa, 12), data=train)
summary(adl.model)
```

```
##
## Time series regression with "ts" data:
## Start = 14, End = 132
##
## Call:
## dynlm(formula = dpEur ~ L(dpEur, 6) + L(dpEur, 12) + L(dpUsa,
##     1) + L(dpUsa, 12), data = train)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0067809 -0.0016356  0.0000532  0.0013660  0.0082448
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0003391  0.0002676   1.267   0.2076
## L(dpEur, 6)   0.1687310  0.0710801   2.374   0.0193 *
## L(dpEur, 12)  0.6551529  0.0856263   7.651 6.93e-12 ***
## L(dpUsa, 1)   0.2326460  0.0507772   4.582 1.19e-05 ***
## L(dpUsa, 12) -0.2264880  0.0540694  -4.189 5.55e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.002273 on 114 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.5561, Adjusted R-squared:  0.5406
## F-statistic: 35.71 on 4 and 114 DF,  p-value: < 2.2e-16
```

(e) Use the models of parts (c) and (d) to make two series of 12 monthly inflation forecasts for 2011. At each month, you should use the data that are then available, for example, to forecast inflation for September 2011 you can use the data up to and including August 2011. However, do not re-estimate the model and use the coefficients as obtained in parts (c) and (d). For each of the two forecast series, compute the values of the root mean squared error (RMSE), mean absolute error (MAE), and the sum of the forecast errors (SUM). Finally, give your interpretation of the outcomes.

ADL Prediction

```
dpEur <- ts(f_data$DPEUR)
dpUsa <- ts(f_data$DPUSA)
test <- ts.intersect(dpEur, dpUsa)
dependentvarname <- "dpEur"
x1<- "dpUsa"

Ntrain <- nrow(train)
Ntest <- nrow(test)
df_train <-data.frame(train)
df_test <- data.frame(test)
# can't rbind ts's apparently, so convert to numeric first
df_train[,dependentvarname] <- as.numeric(df_train[,dependentvarname])
df_test[,dependentvarname] <- NA
df_testtraindata <- rbind( df_train, df_test )
df_testtraindata[,dependentvarname] <- ts( as.numeric( df_testtraindata[,dependentvarname] ) )

l1<-6
l2<- 12
l3 <- 1
l4 <- 12

# step through one by one
for( i in 1:Ntest ) {
  result <- sum(adl.model$coeff * cbind(1, df_testtraindata[Ntrain+i-l1, dependentvarname],
                                df_testtraindata[Ntrain+i-l2, dependentvarname],
                                df_testtraindata[Ntrain+i-l3, x1],
                                df_testtraindata[Ntrain+i-l4, x1]))
  df_testtraindata[Ntrain+i,dependentvarname] <- result
  cat("predicted i",i,"of",Ntest," :",result,"\n")
}
```

```
## predicted i 1 of 12  : -0.00621878
## predicted i 2 of 12  : 0.003634046
## predicted i 3 of 12  : 0.008113709
## predicted i 4 of 12  : 0.005836694
## predicted i 5 of 12  : 0.002269904
## predicted i 6 of 12  : 0.002658253
## predicted i 7 of 12  : -0.003540649
## predicted i 8 of 12  : 0.001792593
## predicted i 9 of 12  : 0.004194302
## predicted i 10 of 12  : 0.003641583
## predicted i 11 of 12  : 0.0005755105
## predicted i 12 of 12  : 0.004114096
```

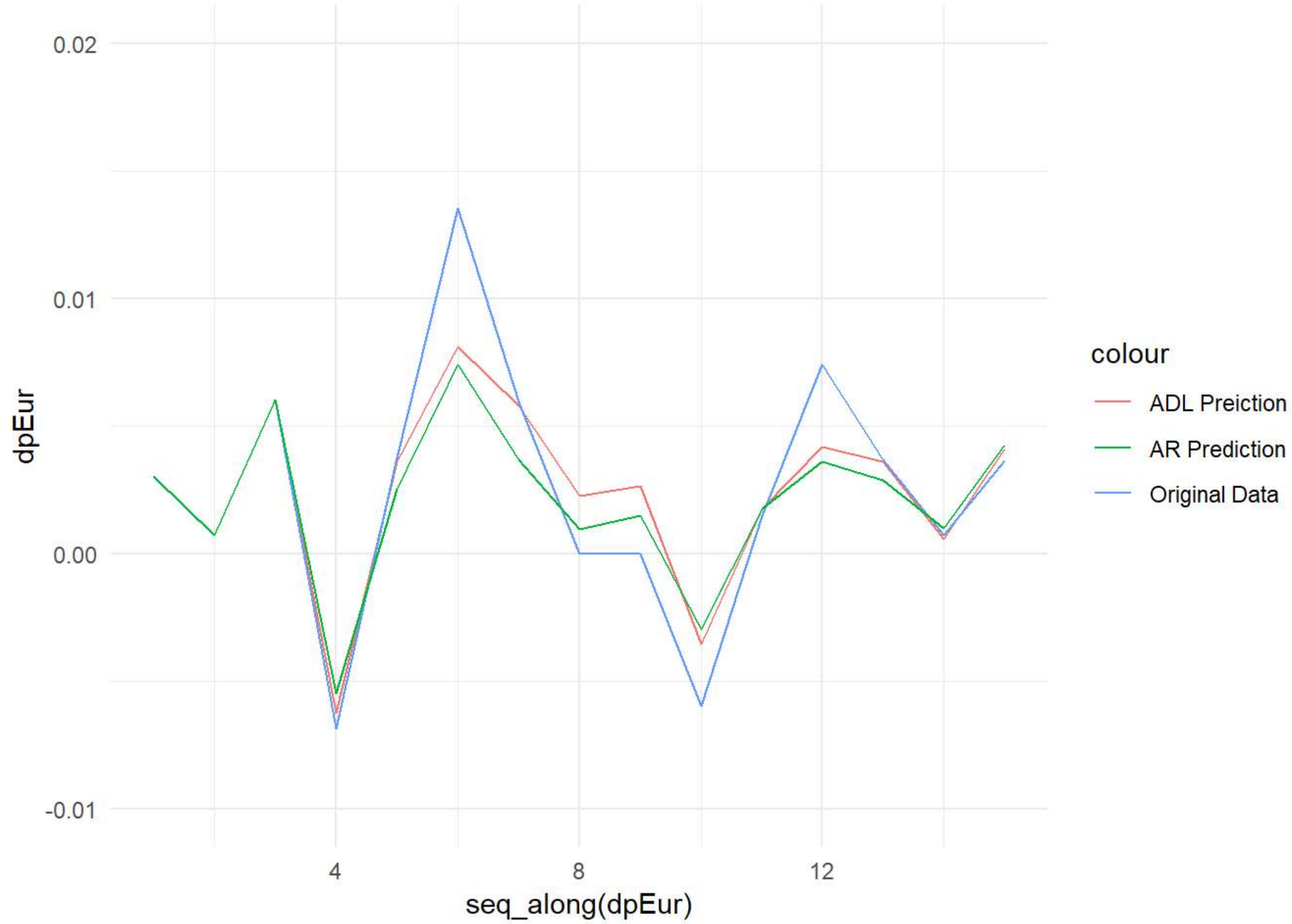
```
dpEur <- ts(f_data$DPEUR)
dpUsa <- ts(f_data$DPUSA)
test <- ts.intersect(dpEur, dpUsa)
dependentvarname <- "dpEur"

Ntrain <- nrow(train)
Ntest <- nrow(test)
df_train <-data.frame(train)
df_test <- data.frame(test)
# can't rbind ts's apparently, so convert to numeric first
df_train[,dependentvarname] <- as.numeric(df_train[,dependentvarname])
df_test[,dependentvarname] <- NA
ardf_testtraindata <- rbind( df_train, df_test )
ardf_testtraindata[,dependentvarname] <- ts( as.numeric( ardf_testtraindata[,dependentvarname] )
)

l1<-6
l2<- 12
#i=8
# step through one by one
for( i in 1:Ntest ) {
  result <- sum(ar.model$coeff * cbind(1, ardf_testtraindata[Ntrain+i-l1, dependentvarname],
                                    ardf_testtraindata[Ntrain+i-l2, dependentvarname]))
  ardf_testtraindata[Ntrain+i,dependentvarname] <- result
  cat("predicted i",i,"of",Ntest," :",result,"\n")
}
```

```
## predicted i 1 of 12  : -0.00543937
## predicted i 2 of 12  : 0.002532843
## predicted i 3 of 12  : 0.007425756
## predicted i 4 of 12  : 0.003708782
## predicted i 5 of 12  : 0.0009842607
## predicted i 6 of 12  : 0.001528509
## predicted i 7 of 12  : -0.002931379
## predicted i 8 of 12  : 0.001778539
## predicted i 9 of 12  : 0.003613982
## predicted i 10 of 12  : 0.002907036
## predicted i 11 of 12  : 0.001024606
## predicted i 12 of 12  : 0.004299039
```

```
df_testtraindata[130:nrow(df_testtraindata),] %>% ggplot(aes(x=seq_along(dpEur), y=dpEur,color=
"ADL Preiction"))+geom_line()+
  geom_line(y= data_orig[130:nrow(data_orig),]$DPEUR, aes(color= "Original Data"))+ylim(-0.01,
0.02)+
geom_line(y= ardf_testtraindata[130:nrow(ardf_testtraindata),]$dpEur, aes(color= "AR Prediction"
))
```




```r
RMSE = function(m, o){
  sqrt(mean((m - o)^2))
}
cat("RMSE For ADL: ",RMSE(df_testtraindata[133:nrow(df_testtraindata),]$dpEur,data_orig[133:nrow
(data_orig),]$DPEUR), "\n")
```

```
## RMSE For ADL:  0.002216519
```

```r
cat("RMSE For AR: ",RMSE(ardf_testtraindata[133:nrow(ardf_testtraindata),]$dpEur,data_orig[133:n
row(data_orig),]$DPEUR), "\n")
```

```
## RMSE For AR:  0.002491731
```

```r
MAE = function(m, o){
  mean(abs(m-o))
}
cat("MAE For ADL: ",MAE(df_testtraindata[133:nrow(df_testtraindata),]$dpEur,data_orig[133:nrow(d
ata_orig),]$DPEUR), "\n")
```

```
## MAE For ADL:  0.001490881
```

```
cat("MAE For AR: ",MAE(ardf_testtraindata[133:nrow(ardf_testtraindata),]$dpEur,data_orig[133:nro
w(data_orig),]$DPEUR), "\n")
```

```
## MAE For AR:  0.0018659
```

```
SUM = function(m, o){
  sum(abs(m - o))
}
cat("SUM For ADL: ",SUM(df_testtraindata[133:nrow(df_testtraindata),]$dpEur,data_orig[133:nrow(d
ata_orig),]$DPEUR), "\n")
```

```
## SUM For ADL:  0.01789057
```

```
cat("SUM For AR: ",SUM(ardf_testtraindata[133:nrow(ardf_testtraindata),]$dpEur,data_orig[133:nro
w(data_orig),]$DPEUR), "\n")
```

```
## SUM For AR:  0.0223908
```

Conclusion:

Clearly, the ADL model performed better forecasts than the AR model, as it scored less at all errors scores.

This means that the insight used in part (d), about using the USA monthly inflation series at lags 1 and 12 in order to forecast the EURO monthly inflation, was correct.