Unit 4 E14.1 problems A, B and C

December 4, 2020

Econometrics II, Bachelor degree in Economics

Universitat Autònoma de Barcelona

Student, Roylan Martinez Vargas

Professor, Michael Creel

NIU: 1539069

E14.1 On the text website, **www.pearsonglobaleditions.com/Stock_Watson**, you will find the data file **USMacro_Quarterly**, which contains quarterly data on several macroeconomic series for the United States; the data are described in the file **USMacro_Description**. The variable *PCEP* is the price index for personal consumption expenditures from the U.S. National Income and Product Accounts. In this exercise you will construct forecasting models for the rate of inflation, based on *PCEP*. For this analysis, use the sample period 1963:Q1–2012:Q4 (where data before 1963 may be used, as necessary, as initial values for lags in regressions).

a. i. Compute the inflation rate, $Infl = 400 \times [\ln(PCEP_t) - \ln(PCEP_{t-1})]$. What are the units of *Infl*? (Is *Infl* measured in dollars, percentage points, percentage per quarter, percentage per year, or something else? Explain.)

ii. Plot the value of *Infl* from 1963:Q1 through 2012:Q4. Based on the plot, do you think that *Infl* has a stochastic trend? Explain.

b. i. Compute the first four autocorrelations of $\Delta Infl$.

ii. Plot the value of $\Delta Infl$ from 1963:Q1 through 2012:Q4. The plot should look "choppy" or "jagged." Explain why this behavior is consistent with the first autocorrelation that you computed in part (i).

c i. Run an OLS regression of $\Delta Infl_t$ on $\Delta Infl_{t-1}$. Does knowing the change in inflation this quarter help predict the change in inflation next quarter? Explain.

ii. Estimate an AR(2) model for $\Delta Infl$. Is the AR(2) model better than an AR(1) model? Explain.

iii. Estimate an AR(p) model for $p = 0, \ldots, 8$. What lag length is chosen by BIC? What lag length is chosen by AIC?

iv. Use the AR(2) model to predict the change in inflation from 2012:Q4 to 2013:Q1—that is, predict the value of $\Delta Infl_{2013:Q1}$.

v. Use the AR(2) model to predict the level of the inflation rate in 2013:Q1—that is, $Infl_{2013:Q1}$.

[13]:
```
# TOOLS
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.formula.api import ols
```

[12]:
```
# Data without null values and with the obs column as the index
dataExcel = pd.read_csv("us_macro_quarterly.csv", index_col='obs').dropna()
data_converted_index = pd.to_datetime(dataExcel.index)
dataExcel.index = pd.DatetimeIndex(data_converted_index.values, freq =␣
 ↪data_converted_index.inferred_freq)

# New inflation rate variable
dataExcel['INFL'] = 400 * (np.log(dataExcel['PCECTPI']) - np.
 ↪log(dataExcel['PCECTPI']).shift())

# New change in inflation rate variable
dataExcel['changeINFL'] = dataExcel['INFL'] - dataExcel['INFL'].shift()
dataExcel
```

[12]:
```
                GDPC96   JAPAN_IP  PCECTPI      GS10       GS1     TB3MS  \
1959-01-01    2973.782   9.425179   17.137  3.990000  3.503333  2.773333
1959-04-01    3046.096  10.080844   17.204  4.256667  3.916667  3.000000
1959-07-01    3040.235  10.681870   17.307  4.503333  4.603333  3.540000
1959-10-01    3052.194  11.419493   17.401  4.583333  4.916667  4.230000
1960-01-01    3120.195  12.184435   17.424  4.486667  4.570000  3.873333
…                  …          …        …         …         …         …
2012-07-01   15533.985  95.792017  106.193  1.643333  0.183333  0.103333
2012-10-01   15539.628  94.258812  106.622  1.706667  0.173333  0.086667
2013-01-01   15583.948  94.725440  106.909  1.950000  0.153333  0.086667
```

```
2013-04-01  15679.677  95.992001  106.878  1.996667  0.126667  0.050000
2013-07-01  15839.347  97.558537  107.387  2.710000  0.123333  0.033333

              UNRATE    EXUSUK    CPIAUCSL       INFL  changeINFL
1959-01-01  5.833333  2.809507   28.993333        NaN         NaN
1959-04-01  5.100000  2.814537   29.043333   1.560818         NaN
1959-07-01  5.266667  2.808283   29.193333   2.387652    0.826833
1959-10-01  5.600000  2.802463   29.370000   2.166653   -0.220999
1960-01-01  5.133333  2.802970   29.396667   0.528356   -1.638297
…                …         …           …        …           …
2012-07-01  8.033333  1.581367  230.029667   1.672150    0.615436
2012-10-01  7.833333  1.606433  231.277000   1.612670   -0.059480
2013-01-01  7.700000  1.550633  232.102667   1.075254   -0.537416
2013-04-01  7.500000  1.536700  232.086667  -0.116003   -1.191258
2013-07-01  7.233333  1.552300  233.597000   1.900454    2.016457

[219 rows x 11 columns]
```

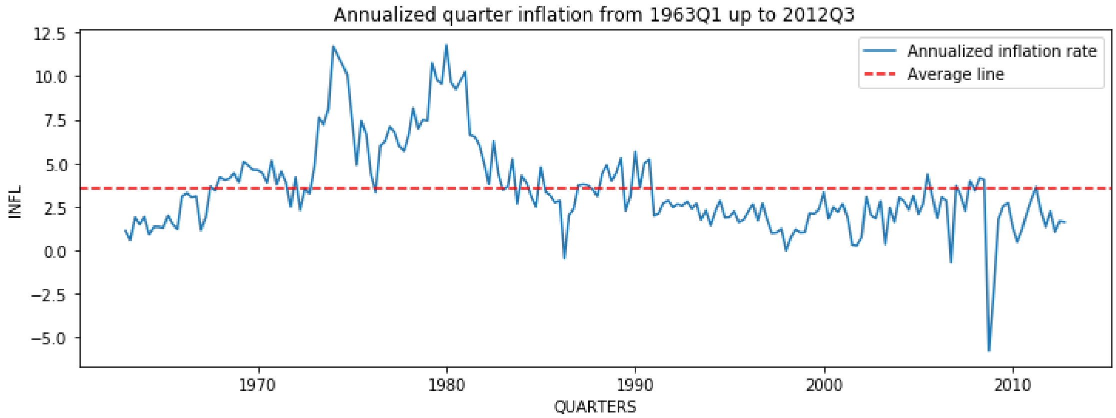
a. i.

The new variable **INFL** measures the approximate annualized percentage change of the **PCECTPI**. It is measured in percentage per year because the approximate percentage change of the quarter was annualized when multiplying it by four.

a. ii.

[11]:
```python
# Plot the infl from the first quarter of 1963 up to the fourth quarter of 2012
plt.plot(dataExcel.loc['1963-01-01': '2012-10-01', 'INFL'], label='Annualized␣
↪inflation rate')

# INFL mean and plot information
plt.axhline(y=np.mean(dataExcel['INFL'].loc['1963-01-01': '2012-10-01'] ),␣
↪linestyle='--', color='r', label='Average line')
plt.title('Annualized quarter inflation from 1963Q1 up to 2012Q3')
plt.ylabel('INFL')
plt.xlabel('QUARTERS')
plt.rcParams["figure.figsize"] = [12,4]
plt.legend()
plt.show()
```



INFL does look to have a a non-stationary or a weak stationary stochastic trend, because the values seem to be inside of a range of random numbers that are indexed by time but they do not seem to have the same autocovariance and same mean for all the **quarters** as the *red dotted line* intuitively shows.

b. i.

[4]:
```
# Compute correlation coefficient from 1 to 4
dataset = dataExcel['changeINFL'].loc['1963-01-01': '2012-10-01']
acf(dataset, fft=True, nlags=4)
```

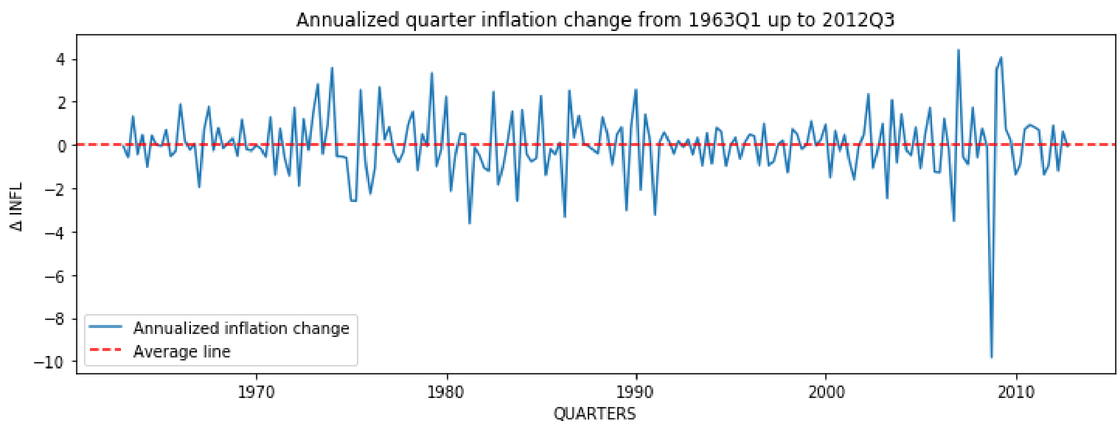
[4]:
```
array([ 1.        , -0.25309496, -0.18285883,  0.13264088, -0.09238671])
```

The autocorrelation with **1**, **2**, **3** and **4** lags are **- 0.25309496**, **- 0.18285883**, **0.13264088** and **- 0.09238671** respectively.

b. ii.

[5]:
```
# Plot the infl from the first quarter of 1963 up to the fourth quarter of 2012
plt.plot(dataExcel['changeINFL'].loc['1963-01-01': '2012-10-01'],␣
 ↪label='Annualized inflation change')

# INFL mean and plot information
plt.title('Annualized quarter inflation change from 1963Q1 up to 2012Q3')
plt.axhline(y=0, linestyle='--', color='r', label='Average line')
plt.ylabel('Δ INFL')
plt.xlabel('QUARTERS')
plt.rcParams["figure.figsize"] = [12,4]
plt.legend()
plt.show()
```



When there is a negative correlation we expect to see a pattern of values crossing the *red dotted line* too often. This pattern is caused because the previus value seems to be related with an opposite sign in the next value. As we see in the plot, the values cross the *red dotted line* too often and this is consistent with the fact that in the previus exercise, the first autocorrelation was of **-0.25309756405695916**, a negative correlation.

c. i.

[6]:
```
# Computing the model with ols
modelA = ols('changeINFL ~ changeINFL.shift()', data=dataExcel.loc['1963-01-01':
↪ '2012-10-01']).fit(cov_type='HC1')
print(modelA.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:             changeINFL   R-squared:                       0.064
Model:                            OLS   Adj. R-squared:                  0.059
Method:                 Least Squares   F-statistic:                     12.67
Date:                Fri, 04 Dec 2020   Prob (F-statistic):           0.000466
Time:                        11:28:35   Log-Likelihood:                -356.48
No. Observations:                 199   AIC:                             717.0
Df Residuals:                     197   BIC:                             723.5
Df Model:                           1
Covariance Type:                  HC1
================================================================================
======
                         coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
------
Intercept              0.0030      0.103      0.029      0.977      -0.200
0.206
```

```
changeINFL.shift()    -0.2531      0.071     -3.560      0.000      -0.392
-0.114
==============================================================================
Omnibus:                       90.025   Durbin-Watson:                   2.133
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              970.079
Skew:                          -1.389   Prob(JB):                    2.24e-211
Kurtosis:                      13.453   Cond. No.                         1.50
==============================================================================

Warnings:
[1] Standard Errors are heteroscedasticity robust (HC1)
```

It seems to be that the next change in inflation is approximately **- 0.2531** times the current change in inflation.
It is important to notice that is reasonable to be sceptic for two reasons, the obvius one is that, even though the correlation coefficient is significantly different from zero, the R squared of this model is quite small and therefore this model apparently just explains a small proportion of the variance in the sampled future changes. The second reason is that for seasonal patterns the OLS method does not assume normality and we would have wider confidence intervals and therefore lower precision (heteroskedasticity and autocorrelation (HC1) robust standard errors can improve it but it is still a problem. So, we could argue that the current inflation change helps to predict the change in inflation next quarter, but it is not appropiate to use OLS to compute the autoregressive models since we might get sligthly less precise estimates.

c. ii.

[7]:
```python
# Make AR(1) model
modelB = ARIMA(dataExcel['changeINFL'].loc['1963-01-01': '2012-10-01'], (1, 0,␣
↪0)).fit()
print(modelB.summary())

# Make AR(2) model
modelC = ARIMA(dataExcel['changeINFL'].loc['1963-01-01': '2012-10-01'], (2, 0,␣
↪0)).fit()
print(modelC.summary())
```

```
                              ARMA Model Results
==============================================================================
Dep. Variable:             changeINFL   No. Observations:                  200
Model:                     ARMA(1, 0)   Log Likelihood                -357.800
Method:                       css-mle   S.D. of innovations              1.448
Date:                Fri, 04 Dec 2020   AIC                            721.600
Time:                        11:28:35   BIC                            731.495
Sample:                    01-01-1963   HQIC                           725.604
                         - 10-01-2012
================================================================================
====
                       coef    std err          z      P>|z|      [0.025
```

```
0.975]
--------------------------------------------------------------------------------
----
const                 0.0022      0.082      0.027      0.979      -0.158
0.163
ar.L1.changeINFL     -0.2518      0.068     -3.692      0.000      -0.386
-0.118
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -3.9707           +0.0000j            3.9707            0.5000
-----------------------------------------------------------------------------
                              ARMA Model Results
==============================================================================
Dep. Variable:             changeINFL   No. Observations:                  200
Model:                     ARMA(2, 0)   Log Likelihood                -350.645
Method:                       css-mle   S.D. of innovations              1.396
Date:                Fri, 04 Dec 2020   AIC                            709.290
Time:                        11:28:35   BIC                            722.483
Sample:                    01-01-1963   HQIC                           714.629
                         - 10-01-2012
================================================================================
====
                       coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------------
----
const                 0.0023      0.063      0.036      0.971      -0.121
0.125
ar.L1.changeINFL     -0.3184      0.068     -4.676      0.000      -0.452
-0.185
ar.L2.changeINFL     -0.2616      0.068     -3.855      0.000      -0.395
-0.129
                                    Roots
=============================================================================
                  Real          Imaginary           Modulus         Frequency
-----------------------------------------------------------------------------
AR.1           -0.6085           -1.8579j            1.9550           -0.3004
AR.2           -0.6085           +1.8579j            1.9550            0.3004
-----------------------------------------------------------------------------
```

The coefficient of determination is not a good indicator when comparing different models since it might increase just with the fact of increasing the number of lagged regressors in the model. To avoid that we can use the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC).
To compare both models, I will directly use the BIC and AIC values:
The AR(1) has a AIC of **721.600** and a BIC of **731.495**

The AR(2) has a AIC of **709.290** and a BIC of **722.483**
As we see, the AR(2) seems to be a better model to explain the next quarter change in inflation with the current and last quarter change in inflation since both AIC and BIC in AR(2) have lower values than in AR(1)

c. iii.

[15]:
```
# Compute AIC and BID for different models
# I used a function to avoid memory leaks with the assignment of variables
def arima_n(delta):
    aic = []
    bic = []
    for delta in range(1, delta + 1):
        model = ARIMA(dataExcel['changeINFL'].loc['1963-01-01': '2012-10-01'],␣
 ↪(delta, 0, 0)).fit()
        aic.append(model.aic)
        bic.append(model.bic)
        print(f'AR({delta}) | AIC: {model.aic:0.10f} | BIC: {model.bic:0.10f}')
    print(f'Optimal AIC lenght: {aic.index(min(aic)) + 1}')
    print(f'Optimal BIC length: {bic.index(min(bic)) + 1}')

arima_n(8)
```

```
AR(1) | AIC: 721.5998453243 | BIC: 731.4947974240
AR(2) | AIC: 709.2901820046 | BIC: 722.4834514708
AR(3) | AIC: 711.2728970970 | BIC: 727.7644839297
AR(4) | AIC: 710.8450970553 | BIC: 730.6350012546
AR(5) | AIC: 710.3517631010 | BIC: 733.4399846668
AR(6) | AIC: 710.7607442781 | BIC: 737.1472832105
AR(7) | AIC: 712.5827088755 | BIC: 742.2675651744
AR(8) | AIC: 714.4160715128 | BIC: 747.3992451783
Optimal AIC lenght: 2
Optimal BIC length: 2
```

The optimal AIC length is **2** with a value of **709.2901820046234**
The optimal BIC length is **2** with a value of **722.4834514708156**

c. iv.

[9]:
```
# Print the first step out-of-sample forecast for the change in inflation
modelC.forecast(1)
```

[9]:
```
(array([-0.13851351]),
 array([1.39621897]),
 array([[-2.87505242,  2.59802539]]))
```

The predicted change in inflation by AR(2) from 2012Q4 to 2013Q1 is **- 0.13851351** with a standard error of **1.39621897** and a 95% confidence interval between **- 2.87505242** and **2.59802539**

c. iv.

[10]:
```
# Print the first step out-of-sample forecast for the inflation rate
modelD = ARIMA(dataExcel['INFL'].loc['1963-01-01': '2012-10-01'], (2, 0, 0)).
 ↪fit(trend='nc')
print(modelD.forecast(1))
```

```
(1.5463120230697975, array([1.43504143]), array([[-1.2663175 ,  4.35894155]]))
```

The predicted inflation by AR(2) for 2013Q1 is **1.87520473** with a standard error of **1.40925276** and a 95% confidence interval between **- 0.88687992** and **4.63728937**