Assign.6_YM

May 28, 2018

@author: Yiming

```
In [1]: import pandas as pd
        import numpy as np
        import statsmodels.api as sm
        from patsy import  dmatrices
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
/Users/yimingcai/anaconda/lib/python3.6/site-packages/statsmodels/compat/pandas.py:56: FutureWa
  from pandas.core import datetools
```

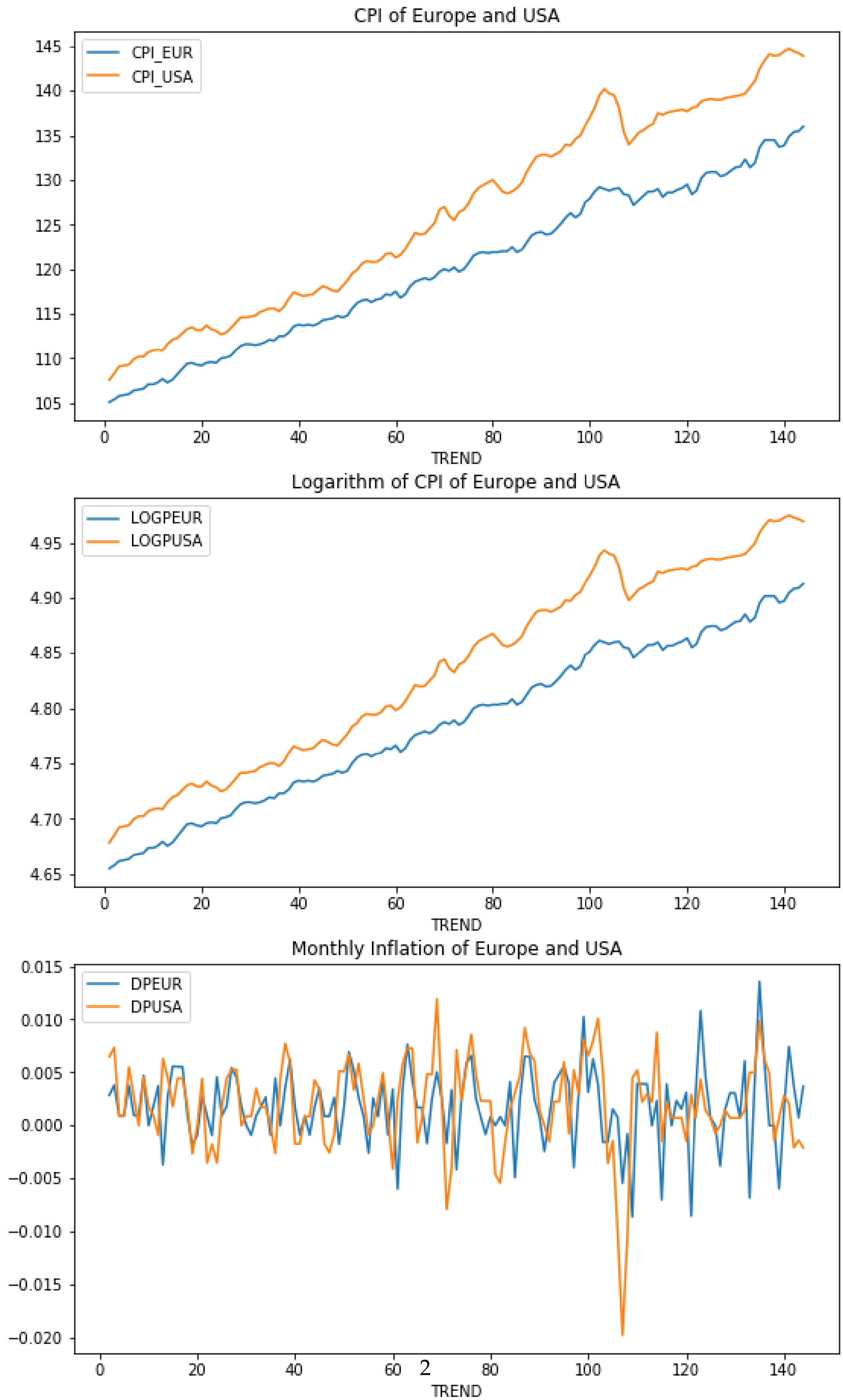
```
In [2]: df = pd.read_excel("Test6_data.xlsx")
```

(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 = log(CPI). What conclusions do you draw from these plots?

```
In [3]: fig, ax = plt.subplots(3, 1, figsize = (9, 16))
        df_plot = df.set_index("TREND").copy()
        df_plot[["CPI_EUR", "CPI_USA"]].plot(ax = ax[0])
        df_plot[["LOGPEUR", "LOGPUSA"]].plot(ax = ax[1])
        df_plot[["DPEUR", "DPUSA"]].plot(ax = ax[2])

        ax[0].set_title("CPI of Europe and USA")
        ax[1].set_title("Logarithm of CPI of Europe and USA")
        ax[2].set_title("Monthly Inflation of Europe and USA")

Out[3]: <matplotlib.text.Text at 0x11dff2668>
```


CPI of Europe and USA
CPI_EUR
CPI_USA
TREND
Logarithm of CPI of Europe and USA
LOGPEUR
LOGPUSA
TREND
Monthly Inflation of Europe and USA
DPEUR
DPUSA
TREND

Conclusion:

1. There exists a deterministic trend for CPI in both Europe and USA, so as their logarithm terms.

2. The CPI for both Europe and USA are non-stationary as their means varies with time.

3. The inflation rate, however, seems stationary and thus can be used for time-series analysis.

4. The inflation rate of Europe and USA seem correlated.

(b) Perform the Augmented Dickey-Fuller (ADF) test for the two log(CPI) series. In the ADF test equation, include a constant (), a deterministic trend term (t), three lags of DP = log(CPI) and, of course, the variable of interest log(CPIt1). Report the coefficient of log(CPIt1) and its standard error and t-value, and draw your conclusion.

```
In [4]: df_lagged1_term = df[["LOGPEUR", "LOGPUSA", "DPEUR", "DPUSA"]].shift(1).rename(columns



        df_lagged2_term = df[["DPEUR", "DPUSA"]].shift(2).rename(columns= {"DPEUR":"DPEUR_L2",
                                                                          "DPUSA":"DPUSA_L2"}

        df_lagged3_term = df[["DPEUR", "DPUSA"]].shift(3).rename(columns = {"DPEUR":"DPEUR_L3"
                                                                           "DPUSA":"DPUSA_L3"
        df_lagged = pd.concat([df, df_lagged1_term, df_lagged2_term, df_lagged3_term], axis =1

In [5]: y_eur, X_eur = dmatrices("DPEUR ~ TREND+LOGPEUR_L1 +DPEUR_L1+DPEUR_L2+DPEUR_L3", df_lag
        y_usa, X_usa = dmatrices("DPUSA ~TREND+LOGPUSA_L1+DPUSA_L1+DPUSA_L2+DPUSA_L3" , df_lag

In [6]: eur_mod = sm.OLS(y_eur, X_eur).fit()
        usa_mod = sm.OLS(y_usa, X_usa).fit()

In [7]: print (eur_mod.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  DPEUR   R-squared:                       0.120
Model:                            OLS   Adj. R-squared:                  0.087
Method:                 Least Squares   F-statistic:                     3.662
Date:                Mon, 28 May 2018   Prob (F-statistic):            0.00388
Time:                        11:49:13   Log-Likelihood:                 601.82
No. Observations:                 140   AIC:                            -1192.
Df Residuals:                     134   BIC:                            -1174.
```

```
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.6420      0.226      2.837      0.005       0.194       1.090
TREND          0.0002    8.5e-05      2.795      0.006    6.94e-05       0.000
LOGPEUR_L1    -0.1374      0.049     -2.826      0.005      -0.234      -0.041
DPEUR_L1       0.1442      0.087      1.665      0.098      -0.027       0.316
DPEUR_L2      -0.0902      0.085     -1.059      0.292      -0.259       0.078
DPEUR_L3      -0.1128      0.086     -1.317      0.190      -0.282       0.057
==============================================================================
Omnibus:                       17.430   Durbin-Watson:                   2.029
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               28.871
Skew:                          -0.606   Prob(JB):                     5.38e-07
Kurtosis:                       4.865   Cond. No.                     7.04e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.04e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
In [8]: print (usa_mod.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  DPUSA   R-squared:                       0.326
Model:                            OLS   Adj. R-squared:                  0.301
Method:                 Least Squares   F-statistic:                     12.97
Date:                Mon, 28 May 2018   Prob (F-statistic):           2.72e-10
Time:                        11:49:13   Log-Likelihood:                 595.89
No. Observations:                 140   AIC:                            -1180.
Df Residuals:                     134   BIC:                            -1162.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.3494      0.127      2.747      0.007       0.098       0.601
TREND          0.0002   5.72e-05      2.645      0.009    3.82e-05       0.000
LOGPUSA_L1    -0.0743      0.027     -2.734      0.007      -0.128      -0.021
DPUSA_L1       0.6091      0.084      7.248      0.000       0.443       0.775
DPUSA_L2      -0.1513      0.096     -1.568      0.119      -0.342       0.040
DPUSA_L3      -0.0064      0.086     -0.075      0.941      -0.177       0.164
==============================================================================
Omnibus:                        6.073   Durbin-Watson:                   1.993
```

```
Prob(Omnibus):                  0.048   Jarque-Bera (JB):               7.828
Skew:                          -0.228   Prob(JB):                      0.0200
Kurtosis:                       4.065   Cond. No.                    3.82e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.82e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

As the results indicate, for both variables, the ADF statistic is greater than the critical value of 3.5. Therefore, the non-stationarity hypothesis is not rejected.

(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 = log(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: DPEURt = + 1DPEURt6 + 2DPEURt12 + t . Estimate the parameters of this model (sample Jan 2000 - Dec 2010).

```
In [9]: dpeur_l6 = df[["DPEUR"]].shift(6).rename(columns = {"DPEUR": "DPEUR_L6"})
        dpeur_l12 = df[["DPEUR"]].shift(12).rename(columns = {"DPEUR": "DPEUR_L12"})
        df_lagged_c = pd.concat([df_lagged, dpeur_l6, dpeur_l12], axis= 1)
```

```
In [10]: from statsmodels.tsa.stattools import acf, pacf
```

```
In [11]: dpeur = df_lagged_c[df_lagged_c.TREND <= 132].DPEUR.dropna().values
```

```
In [12]: acf(dpeur, nlags= 12)[1:]
```

```
Out[12]: array([ 0.08325178, -0.10916313, -0.1990793 , -0.15896745, -0.08844387,
                 0.40291713, -0.0350489 , -0.17326768, -0.16204276, -0.11141802,
                 0.01458082,  0.55447498])
```

```
In [13]: pacf(dpeur, nlags= 12)[1:]
```

```
Out[13]: array([ 0.08389217, -0.11872908, -0.18732104, -0.15337675, -0.12507951,
                 0.39304979, -0.20987763, -0.18067838, -0.07363718, -0.08214183,
                 0.05004201,  0.45590301])
```

The lags with largest ACF and PACF were found at lag6 and lag12

Estimation:

```
In [14]: y_c, X_c = dmatrices("DPEUR ~ DPEUR_L6+DPEUR_L12", data= df_lagged_c[df_lagged_c.TREND
```

```
In [15]: mod_c = sm.OLS(y_c, X_c).fit()
         print (mod_c.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  DPEUR   R-squared:                       0.423
Model:                            OLS   Adj. R-squared:                  0.413
Method:                 Least Squares   F-statistic:                     42.55
Date:                Mon, 28 May 2018   Prob (F-statistic):           1.38e-14
Time:                        11:49:13   Log-Likelihood:                 542.43
No. Observations:                 119   AIC:                            -1079.
Df Residuals:                     116   BIC:                            -1071.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0004      0.000      1.365      0.175      -0.000       0.001
DPEUR_L6       0.1887      0.077      2.442      0.016       0.036       0.342
DPEUR_L12      0.5980      0.084      7.157      0.000       0.432       0.763
==============================================================================
Omnibus:                       10.597   Durbin-Watson:                   1.626
Prob(Omnibus):                  0.005   Jarque-Bera (JB):               19.695
Skew:                          -0.321   Prob(JB):                     5.29e-05
Kurtosis:                       4.887   Cond. No.                         406.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

(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 DPEURt = + 1DPEURt6 + 2DPEURt12 + 1DPUSAt1 + 2DPUSAt12 + t (sample Jan 2000 - Dec 2010).

In [16]:
```
dpusa_l6 = df[["DPUSA"]].shift(6).rename(columns = {"DPUSA": "DPUSA_L6"})
dpusa_l12 = df[["DPUSA"]].shift(12).rename(columns = {"DPUSA": "DPUSA_L12"})
df_lagged_d = pd.concat([df_lagged_c, dpusa_l6, dpusa_l12], axis= 1)
```

In [17]:
```
y_d, X_d = dmatrices("DPEUR~DPEUR_L6+DPEUR_L12+DPUSA_L1+DPUSA_L6+DPUSA_L12", df_lagged
```

In [18]:
```
mod_d = sm.OLS(y_d, X_d).fit()
print (mod_d.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  DPEUR   R-squared:                       0.560
Model:                            OLS   Adj. R-squared:                  0.541
Method:                 Least Squares   F-statistic:                     28.79
Date:                Mon, 28 May 2018   Prob (F-statistic):           9.84e-19
Time:                        11:49:13   Log-Likelihood:                 558.57
```

```
No. Observations:                 119   AIC:                            -1105.
Df Residuals:                     113   BIC:                            -1088.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0004      0.000      1.545      0.125      -0.000       0.001
DPEUR_L6       0.2030      0.079      2.584      0.011       0.047       0.359
DPEUR_L12      0.6368      0.087      7.279      0.000       0.463       0.810
DPUSA_L1       0.2264      0.051      4.429      0.000       0.125       0.328
DPUSA_L6      -0.0560      0.055     -1.023      0.308      -0.165       0.052
DPUSA_L12     -0.2301      0.054     -4.247      0.000      -0.337      -0.123
==============================================================================
Omnibus:                       10.600   Durbin-Watson:                   2.011
Prob(Omnibus):                  0.005   Jarque-Bera (JB):               15.286
Skew:                           0.443   Prob(JB):                     0.000479
Kurtosis:                       4.516   Cond. No.                         512.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

The p-value for DPUSA at lag 6 ("DPUSA_L6") is 0.308, which is not significant at 95% confidence level. Therefore we keep variable "DPUSA_L6" out of the model, and new estimation process goes as follows:

```
In [19]: y_d2, X_d2 = dmatrices("DPEUR~DPEUR_L6+DPEUR_L12+DPUSA_L1+DPUSA_L12", df_lagged_d[df_
         mod_d2 = sm.OLS(y_d2, X_d2).fit()
         print (mod_d2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  DPEUR   R-squared:                       0.556
Model:                            OLS   Adj. R-squared:                  0.541
Method:                 Least Squares   F-statistic:                     35.71
Date:                Mon, 28 May 2018   Prob (F-statistic):           2.55e-19
Time:                        11:49:13   Log-Likelihood:                 558.02
No. Observations:                 119   AIC:                            -1106.
Df Residuals:                     114   BIC:                            -1092.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0003      0.000      1.267      0.208      -0.000       0.001
DPEUR_L6       0.1687      0.071      2.374      0.019       0.028       0.310
```

```
DPEUR_L12       0.6552      0.086      7.651      0.000       0.486       0.825
DPUSA_L1        0.2326      0.051      4.582      0.000       0.132       0.333
DPUSA_L12      -0.2265      0.054     -4.189      0.000      -0.334      -0.119
==============================================================================
Omnibus:                       10.147   Durbin-Watson:                   2.014
Prob(Omnibus):                  0.006   Jarque-Bera (JB):               15.787
Skew:                           0.386   Prob(JB):                     0.000373
Kurtosis:                       4.609   Cond. No.                         481.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

(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.

In [20]:
```python
def predict_future(trend, mod = "C", data = df_lagged_d):
    if mod == "C":
        mod = mod_c
        exogs = ["const","DPEUR_L6", "DPEUR_L12"]
    elif mod == "D":
        mod = mod_d2
        exogs = ["const","DPEUR_L6", "DPEUR_L12", "DPUSA_L1", "DPUSA_L12"]
    else:
        raise Exception("Model does not exist")
    data = sm.add_constant(data)
    predicted_values = mod.predict(data[data.TREND == trend][exogs])
    return predicted_values.values[0]
```

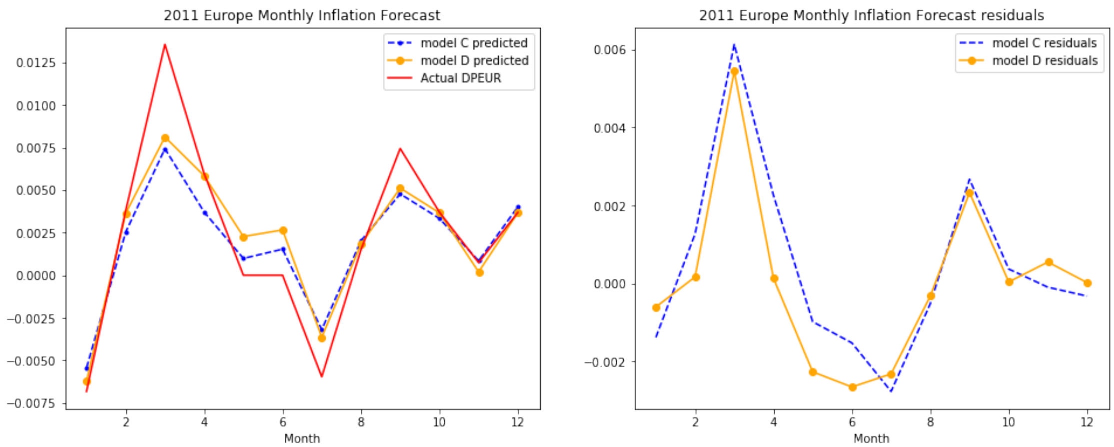
In [21]:
```python
#predicted values
trends = range(133, 145)
mod_c_predicted = []
mod_d_predicted =[]
for trend in trends:
    mod_c_predicted.append(predict_future(trend, mod = "C"))
    mod_d_predicted.append(predict_future(trend, mod= "D"))
#actual values
real_dpeur = df_lagged_d[df_lagged_d.TREND.isin(trends)].DPEUR.values
```

In [22]:
```python
fig, ax = plt.subplots(1, 2, figsize = (16, 6))
xs = range(1, 13)
ax[0].plot(xs, mod_c_predicted, label = "model C predicted", linestyle = "--", marker
ax[0].plot(xs, mod_d_predicted, label = "model D predicted", marker ="o", c= "orange"
```

```
ax[0].plot(xs, real_dpeur, label = "Actual DPEUR", c = "red")
ax[0].legend()
ax[0].set_title("2011 Europe Monthly Inflation Forecast")
ax[0].set_xlabel("Month")

ax[1].plot(xs, real_dpeur- mod_c_predicted , label = "model C residuals", linestyle =
ax[1].plot(xs, real_dpeur-  mod_d_predicted, label = "model D residuals", marker ="o"
ax[1].set_title("2011 Europe Monthly Inflation Forecast residuals")
ax[1].set_xlabel("Month")
ax[1].legend()
```

Out[22]: <matplotlib.legend.Legend at 0x11f13b278>




```
In [23]: RMSE_C  = np.sqrt(sum([resid**2 for resid in (real_dpeur- mod_c_predicted)])/12)
         RMSE_D  = np.sqrt(sum([resid**2 for resid in (real_dpeur- mod_d_predicted)])/12)
         MAE_C = sum([np.abs(resid) for resid in (real_dpeur- mod_c_predicted)])/12
         MAE_D = sum([np.abs(resid) for resid in (real_dpeur- mod_d_predicted)])/12
         SUM_C = sum([resid for resid in (real_dpeur- mod_c_predicted)])
         SUM_D = sum([resid for resid in (real_dpeur- mod_d_predicted)])
```

In [24]: RMSE_C

Out[24]: 0.0023241954513586365

In [25]: RMSE_D

Out[25]: 0.0021105252005109189

In [26]: MAE_C

Out[26]: 0.0016924268523882621

In [27]: MAE_D

```
Out[27]: 0.0014036628892379894
```

```
In [28]: SUM_C
```

```
Out[28]: 0.0050653525975492527
```

```
In [29]: SUM_D
```

```
Out[29]: 0.00047846854944003504
```

Based on the statistics above, model D outperforms Model C.