# MTH9893 Homework 2 Sample Solutions

Source: Q1,2 - Group03, Q3 - Group12, Q4 - Group 13

# Question 1

Consider the AR(1) model with the following choice of parameters: $\alpha = 0.1$, $\beta = 0.3$, $\sigma = 0.005$. The purpose of this Problem is to show that the MLE estimator of the parameters of the model is biased. Run three series of simulations of the models with the following numbers of observations: (i) $T = 100$, (ii) $T = 250$, and (iii) $T = 1250$.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima_model import ARMA
```

In [2]:
```
alpha = 0.1
beta = 0.3
sigma = 0.005
N = 2000 # # of simulations
x0 = alpha/(1-beta)
#simulate AR(1) with T = 100
list_T = [100,250,1250] #list of Ts
result = {} # Results as the dictionary of observations T to the estim
ators
for T in list_T:
    #Initialize the counter of alpha, beta, and sigma
    alpha_avg = 0
    beta_avg = 0
    sigma_avg = 0
    for n in range(N):
        x = np.zeros(T+1)
        x[0] = x0
        eps = np.random.normal(0.0,sigma,T)
        for i in range(1,T+1):
            x[i] = alpha + beta*x[i-1] + eps[i-1]
        #MLE estimate with statsmodels sometimes does not converge and
it's biased.
        #If we use CMLE instead, it converges and also biased, but con
verges to the real parameter
        #model = ARMA(x,order=(1,0)).fit(method = 'mle')
        y=x[0:T]
        yp=x[1:(T+1)]
        m=np.sum(y)/T
        mp=np.sum(yp)/T
        betaCMLE=np.inner(y-m,yp-mp)/np.inner(y-m,y-m)
        alphaCMLE=mp-betaCMLE*m
        sigmaCMLE=np.sqrt(np.inner(yp-betaCMLE*y-alphaCMLE,yp-betaCMLE
*y-alphaCMLE)/T)
        alpha_avg += alphaCMLE/N
        beta_avg += betaCMLE/N
        sigma_avg += sigmaCMLE/N
    result[T]=([alpha_avg,beta_avg,sigma_avg])
```

In [3]:
```
for T in list_T:
    print("The alpha, beta and sigma estimator for {0} observations ar
e \n {1}".format(T,result[T]))
```

```
The alpha, beta and sigma estimator for 100 observations are
 [0.10244455903281442, 0.28288256635465681, 0.0049504100377116668]
The alpha, beta and sigma estimator for 250 observations are
 [0.10141704918864249, 0.2900839259712058, 0.0049780240804698341]
The alpha, beta and sigma estimator for 1250 observations are
 [0.10017763961576477, 0.29875633957300124, 0.004993947686996846]
```

From the result above, we see that when observations are small, the expectation of beta and sigma(calculated from simulation) has some gaps between real parameters, and the gap shrinks as observations goes up.. Therefore, the MLE estimators are biased.

# Question 2

Consider the last 15 years worth of the USD EUR exchange rate and the differentials between the official (Federal Reserve and European Central Bank) short interest rates in these currencies. You can download the time series from Bloomberg: the rates screens are FEDL01 and EUORDEPO, respectively). Economic theory says that these two time series should be cointegrated with cointegrating vector $a = (1, -1)^T$. Design a test to verify this theory.

Step 1: We first run the adf test on the two series, to test they are I(1)

In [4]:
```
import pandas as pd
```

In [5]:
```
EUORDEPO = pd.read_excel("EUORDEPO.xlsx")
USDEUR = pd.read_excel("USDEUR.xlsx")
```

In [6]:
```
#Resample, rename, and concatenate the 3 datasets
FEDL01 = pd.read_excel("FEDL01.xlsx")
FEDL01 = FEDL01.set_index("Date")
FEDL01 = FEDL01.resample("W-MON").mean()
FEDL01 = FEDL01.rename(columns={"PX_LAST":"FEDL01"})

EUORDEPO = pd.read_excel("EUORDEPO.xlsx")
EUORDEPO = EUORDEPO.set_index("Date")
EUORDEPO = EUORDEPO.resample("W-MON").mean()
EUORDEPO = EUORDEPO.rename(columns={"PX_LAST":"EUORDEPO"})

USDEUR = pd.read_excel("USDEUR.xlsx")
USDEUR = USDEUR.set_index("Date")
USDEUR = USDEUR.resample("W-MON").mean()
USDEUR = USDEUR.rename(columns={"PX_LAST":"USDEUR"})

mydata = pd.concat([FEDL01,EUORDEPO,USDEUR],axis = 1)
```

In [7]:
```
mydata["Rate_Diff"] = mydata["FEDL01"] - mydata["EUORDEPO"]
mydata["Rate_Diff_dif1"] = mydata["Rate_Diff"].diff().fillna(0)
mydata["USDEUR_dif1"] = mydata["USDEUR"].diff().fillna(0)
mydata.head()
```

Out[7]:

| | FEDL01 | EUORDEPO | USDEUR | Rate_Diff | Rate_Diff_dif1 | USDEUR_dif1 |
|---|---|---|---|---|---|---|
| **Date** | | | | | | |
| **2000-01-03** | 5.430 | 2.0 | 0.97630 | 3.430 | 0.000 | 0.00000 |
| **2000-01-10** | 5.536 | 2.0 | 0.97094 | 3.536 | 0.106 | -0.00536 |
| **2000-01-17** | 5.584 | 2.0 | 0.97762 | 3.584 | 0.048 | 0.00668 |
| **2000-01-24** | 5.526 | 2.0 | 0.98802 | 3.526 | -0.058 | 0.01040 |
| **2000-01-31** | 5.608 | 2.0 | 1.01306 | 3.608 | 0.082 | 0.02504 |

In [8]:
```
import statsmodels.tsa.stattools as tsa

adf_results = {}
test_cols =  ["Rate_Diff", "USDEUR","Rate_Diff_dif1","USDEUR_dif1"]
for col in test_cols:
    adf_results[col] = tsa.adfuller(mydata[col])
```

In [9]:
```
for col in test_cols:
    print(adf_results[col][1])
```

```
0.046016542204
0.534223483227
2.95931468278e-05
1.91901093143e-28
```

From above, we find the first 2 columns(level) both > 0.01 and thus insignificant at 1% level. The last 2 columns(first-order difference) both significantly < 0.01. Therefore, we can confirm that the difference between European and US short rates, and USDEUR, are both I(1) series

Step 2: we use the given cointegration vector $a = (1, -1)^T$ to calculate the cointegrating residual process $u_t$ that is, the difference between those two series.

In [10]:
```
mydata["Coint Resid"] = mydata["Rate_Diff"] - mydata["USDEUR"]
```

Step 3: we perform a unit root test on $u_t$ to determine whether it is I(0) (i.e. stationary)

In [11]:
```
adf_results["Coint Resid"] = tsa.adfuller(mydata["Coint Resid"])
print(adf_results["Coint Resid"][1])
```

```
0.0583524329855
```

Since the p-value is between 0.05 and 0.1, we may reject the null hypothesis at 10% significance level, but accept it at 5% level. That is, with 90% confidence, we can say that the USD_EUR exchange rate and the differentials between the official short interest rates in these currencies.

# Question 3

Let

$$\hat{\varepsilon}_t = X_t - \alpha - \delta t - \beta X_{t-1}$$

for $t = 1, \ldots, T$, be the disturbances implied from the data. According to the model specification, each $\hat{\varepsilon}_t$ is independently drawn from $N(0, \sigma^2)$, and thus:

$$\begin{aligned}
\mathcal{L}(\theta|y) &= \frac{1}{(2\pi\sigma^2)^{T/2}}\exp(-\frac{1}{2\sigma^2}\sum_{t=1}^{T}\varepsilon_t^2) \\
&= \frac{1}{(2\pi\sigma^2)^{T/2}}\exp(-\frac{1}{2\sigma^2}\sum_{t=1}^{T}(X_t - \alpha - \delta t - \beta X_{t-1})^2) \\
-\log\mathcal{L}(\theta|y) &= \frac{1}{2}T\log\sigma^2 + \frac{1}{2\sigma^2}\sum_{t=1}^{T}(X_t - \alpha - \delta t - \beta X_{t-1})^2 + const
\end{aligned}$$

Minimizing the LLF yields:

$$\begin{pmatrix} T & \sum X_{t-1} & \frac{T^2+T}{2} \\ \sum X_{t-1} & \sum X_{t-1}^2 & \sum tX_{t-1} \\ \frac{T^2+T}{2} & \sum tX_{t-1} & \frac{T(T+1)(2T+1)}{6} \end{pmatrix}\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \\ \hat{\delta} \end{pmatrix} = \begin{pmatrix} \sum X_t \\ \sum X_{t-1}X_t \\ \sum tX_t \end{pmatrix}$$

$$\hat{\sigma}^2 = \frac{1}{T}\sum(X_t - \hat{\beta}X_{t-1} - \hat{\delta}t - \hat{\alpha})^2$$

The summations above are from t = 1 to T.
This can also be explicitly rewritten as:

$$\hat{\delta} = \frac{\sum(X_{t-1} - \hat{X})^2\sum(t-\bar{t})(X_t - \hat{X}_+) - \sum(t-\bar{t})(X_{t-1} - \hat{X})\sum(X_{t-1} - \hat{X})(X_t - \hat{X}_+)}{\sum(t-\bar{t})^2\sum(X_{t-1} - \hat{X})^2 - [\sum(t-\bar{t})(X_{t-1} - \hat{X})]^2}$$

$$\hat{\beta} = \frac{\sum(t-\bar{t})^2\sum(X_{t-1} - \hat{X})(X_t - \hat{X}_+) - \sum(t-\bar{t})(X_{t-1} - \hat{X})\sum(t-\bar{t})(X_t - \hat{X}_+)}{\sum(t-\bar{t})^2\sum(X_{t-1} - \hat{X})^2 - [\sum(t-\bar{t})(X_{t-1} - \hat{X})]^2}$$

$$\hat{\alpha} = \hat{X}_+ - \hat{\delta}\bar{t} - \hat{\beta}\hat{X}$$
$$\hat{\sigma}^2 = \frac{1}{T}\sum(X_t - \hat{\beta}X_{t-1} - \hat{\delta}t - \hat{\alpha})^2$$

where:

$$\hat{X} = \frac{1}{T}\sum_{t=1}^{T} X_{t-1}$$
$$\hat{X}_+ = \frac{1}{T}\sum_{t=1}^{T} X_t$$
$$\bar{t} = \frac{1}{T}\sum_{t=1}^{T} t$$

# Question 4

The GARCH(1,1) model is:

$$\epsilon_t = \sigma_t z_t$$
$$\sigma_t^2 = \kappa + \eta\sigma_{t-1}^2 + \xi\epsilon_{t-1}^2$$

So

$$E(\sigma_t^2) = \frac{\kappa}{1-\eta-\xi}$$

And

$$E(\epsilon_t^2) = E(\sigma_t^2)$$
$$E(\epsilon_t^4) = 3E(\sigma_t^4)$$

Then:

$$E(\sigma_t^4) = E(\kappa^2 + (\eta\sigma_{t-1}^2 + \xi\epsilon_{t-1}^2)^2 + 2\kappa(\eta\sigma_{t-1}^2 + \xi\epsilon_{t-1}^2))$$
$$= \kappa^2 + \eta^2 E(\sigma_{t-1}^2) + 2\eta\xi E(\sigma_{t-1}^2) + 3\xi^2 E(\sigma_{t-1}^2) + 2\kappa\eta\frac{\kappa}{1-\eta-\xi} + 2\kappa\xi\frac{\kappa}{1-\eta-\xi}$$

Since $E(\sigma_t^4) = E(\sigma_{t-1}^4)$

$$(1-(\xi+\eta)^2 - 2\xi^2)E(\sigma_t^4) = \kappa^2 + 2(\eta+\xi)\frac{\kappa^2}{1-\eta-\xi}$$

Thus

$$\frac{E(\epsilon_t^4)}{(E(\epsilon_t^2))^2} = 3\frac{(1-\xi-\eta)^2(1+2(\eta+\xi)\frac{1}{1-\eta-\xi})}{1-(\xi+\eta)^2-2\xi^2} \qquad = 3\frac{(1-\eta-\xi)(1+\eta+\xi)}{1-(\xi+\eta)^2-2\xi^2} = 3\frac{1-(\eta+\xi)^2}{1-(\xi+\eta)^2-2}$$

```
In [ ]:
```