MTH9893 Time Series Analysis HW1

- Group 01
- Author: Pan, Hongchao & Sun, Yu
- Kernel version: Python 3.5
- Packages: pandas_datareader, datetime, pandas, statsmodels
- Data: 02/04/2007-02/05/2017 **Adjust close price** of 'SPY' and 'IWV'
- Notes: **Please install pandas before running the notebook**
- Notes: **The total running time of this notebook may up to 5min and please do not comment the tables and figures**

Question 1

In [1]:
```
# import the packages
# using DataReader function from pandas_datareader
# Ref: https://github.com/pydata/pandas-datareader/blob/master/pandas_
datareader/data.py
from pandas_datareader import data
import datetime as dt
import pandas as pd
import statsmodels.api as sm
from statsmodels.tsa.ar_model import AR
import numpy as np
import matplotlib.pyplot as plt
import time
```

In [2]:
```
# Define a function to get the data
def get_data():
    # Reading the historical daily returns of SPY and IWV over the las
t 10 years
    start, end=dt.datetime(2007,2,4),dt.datetime(2017,2,5)
    data1=data.DataReader('SPY',data_source='yahoo',start=start,end=en
d)
    data2=data.DataReader('IWV',data_source='yahoo',start=start,end=en
d)

    # Rename the Adj close of two tickers
    data1.rename(columns={'Adj Close':'SPY Adj Close'},inplace=True)
    data2.rename(columns={'Adj Close':'IWV Adj Close'},inplace=True)

    # Merge the data and only use the adjust close price for both tick
ers
    dataAll=data1.merge(data2,left_index=True,right_index=True).loc[:,
['SPY Adj Close','IWV Adj Close']]
    dataAll['SPY Adj Close']=dataAll['SPY Adj Close'].astype(float)
    dataAll['IWV Adj Close']=dataAll['IWV Adj Close'].astype(float)

    # Get the log/percentage returns
    dataAll['SPY log return']=np.log(dataAll['SPY Adj Close']).diff()[
1:]
    dataAll['IWV log return']=np.log(dataAll['IWV Adj Close']).diff()[
1:]
    dataAll['SPY pert return']=dataAll['SPY Adj Close'].pct_change()
    dataAll['IWV pert return']=dataAll['IWV Adj Close'].pct_change()

    # Get the difference of returns between two tickers
    dataAll['log difference']=dataAll['SPY log return']-dataAll['IWV l
og return']
    dataAll['pert difference']=dataAll['SPY pert return']-dataAll['IWV
pert return']

    return dataAll
```

In [3]:
```
dataAll=get_data()
```

In [4]:
```
# Plot the log differences VS percentage differences
plt.figure(1)
plt.subplot(211)
plt.title('Log return VS percentage return of differences between two
tickers')
plt.plot(dataAll['log difference'])
plt.ylabel('Log difference')
plt.xlabel('Date')

plt.subplot(212)
# Plot the percentage differences
plt.plot(dataAll['pert difference'],color='green')
plt.ylabel('Percentage difference')
plt.xlabel('Date')
plt.show()
```

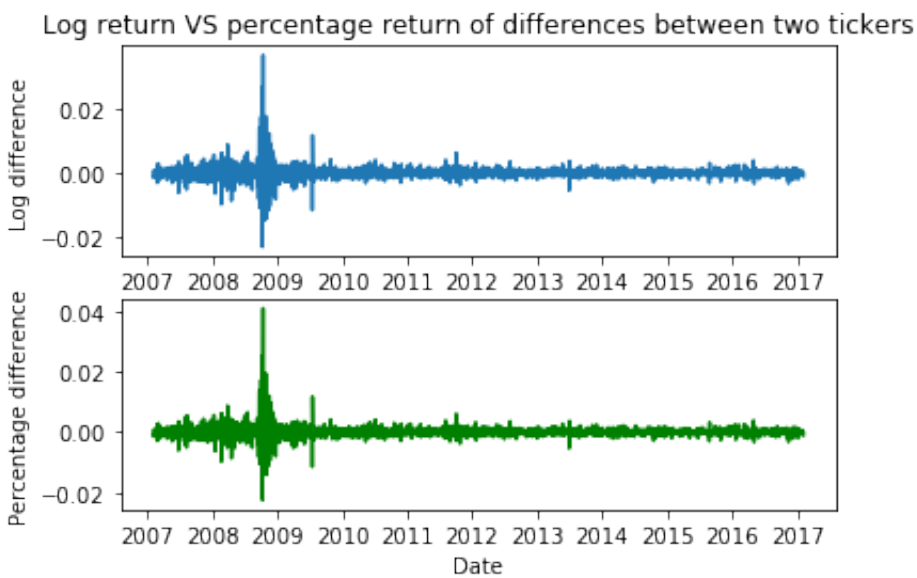


Figure 1 Log return and percentage return comparison of differences between two tickers over past 10 years

In [5]:
```
# Set the maximum lags of AR model to a variable
# max_ma=0: us AR model
# ic: use AIC and BIC criteria
def LagAnalysis(maxAR):
    ST=time.time()
    LogRes=sm.tsa.arma_order_select_ic(dataAll['log difference'].dropn
a(),max_ar=maxAR,max_ma=0,ic=['aic','bic'],trend='c')
    PertRes=sm.tsa.arma_order_select_ic(dataAll['pert difference'].dro
pna(),max_ar=maxAR,max_ma=0,ic=['aic','bic'],trend='c')
    ET=time.time()

    return (LogRes, PertRes, ET-ST)
```

In [6]:
```
# Compare the performance of different maxlag of AR model
def performance(maxARs):
    cols=['maxlag','p of log based on aic','p of log based on bic','p
of pert based on aic','p of pert based on bic','running time(s)']
    ind=[i for i in range(len(maxARs))]

    # Use list comprehension to avoid for loop
    res=[[ar, LagAnalysis(ar)[0].aic_min_order[0], LagAnalysis(ar)[0].
bic_min_order[0],
                   LagAnalysis(ar)[1].aic_min_order[0],LagAnalysis(ar)
[1].bic_min_order[0],LagAnalysis(ar)[2]] for ar in maxARs]

    # Get the corresponding value
    df=pd.DataFrame(res,index=ind,columns=cols)

    return df
```

In [7]:
```
# Set maxlag of AR as 8,10,12 based on previous attempt
startT=time.time()
maxARList=[8,10,12]
df_perf=performance(maxARList)
endT=time.time()
print('Computation time: %s seconds' %(endT-startT))
#print(df_perf)
df_perf
```

```
Computation time: 172.30728912353516 seconds
```

Out[7]:

	maxlag	p of log based on aic	p of log based on bic	p of pert based on aic	p of pert based on bic	running time(s)
0	8	8	8	8	8	4.143735
1	10	10	10	10	9	9.371878
2	12	11	10	11	11	24.724778

Table 1 Performance of the analysis with different maxlag in AR model

In [8]:
```
Log15,Pert15,Time15=LagAnalysis(15)
```

In [9]:
```
print('Result of maxlag=15')
print("p of log based on aic is {}, p of pert based on aic is {}, and running time is {:.2f} seconds".format(
Log15.aic_min_order[0], Pert15.aic_min_order[0],Time15))
```

```
Result of maxlag=15
p of log based on aic is 15, p of pert based on aic is 15, and runni
ng time is 81.07 seconds
```

In [10]:
```
Log15.aic
```

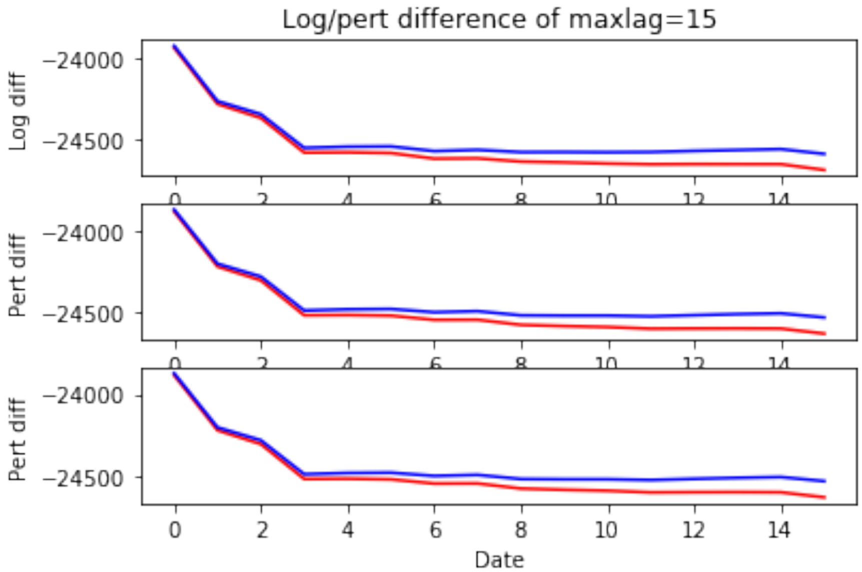
Out[10]:

	0
0	-23930.053329
1	-24277.893778
2	-24364.521803
3	-24578.733328
4	-24577.363790
5	-24581.510736
6	-24616.304470
7	-24614.556784
8	-24634.336624
9	-24639.669064
10	-24646.689696
11	-24651.236535
12	-24650.147336
13	-24650.503428
14	-24650.677895
15	-24686.190710

Table 2 The AIC values of AR(p) with p from 0 to 15

In [11]:
```
def show_plot():
    plt.figure(2)
    plt.subplot(311)
    plt.title('Log/pert difference of maxlag=15')
    plt.plot(Log15.aic,'r',Log15.bic,'b')
    plt.ylabel('Log diff')
    plt.xlabel('Date')

    plt.subplot(312)
    plt.plot(Pert15.aic,'r',Pert15.bic,'b')
    plt.ylabel('Pert diff')
    plt.xlabel('Date')

    plt.subplot(313)
    plt.plot(Pert15.aic,'r',Pert15.bic,'b')
    plt.ylabel('Pert diff')
    plt.xlabel('Date')
    plt.show()

    plt.figure(3)
    plt.subplot(111)
    plt.title('Log VS percentage returns of maxlag=15')
    plt.plot(Log15.aic,'r',Pert15.aic,'b')
    plt.ylabel('Pert VS log diff of maxlag=15')
    plt.xlabel('Date')

    plt.show()
```

In [12]:
```
show_plot()
```



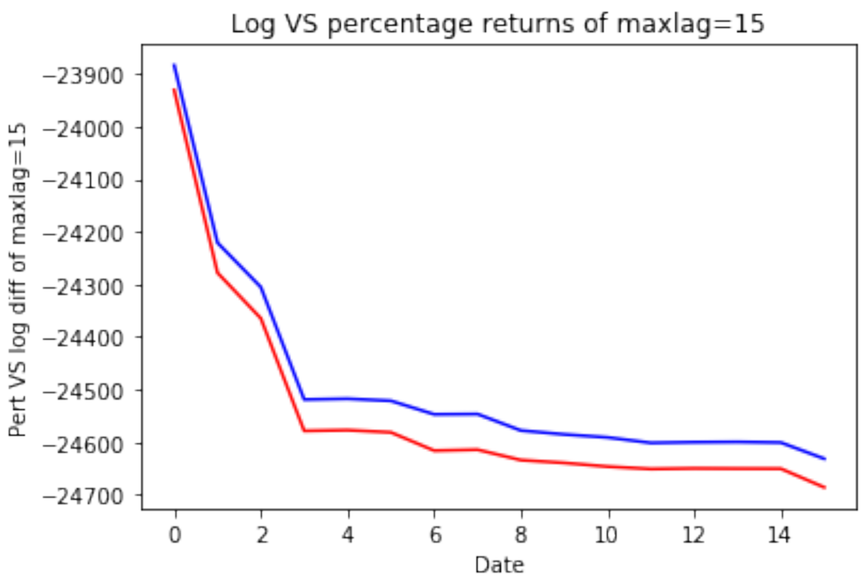


Figure 2 AIC and BIC values comparasion (top), AIC values of log return and percentage return comparison (bottom) of AR(p) model with p from 0 to 15

Discussion of Q1

1. The comparasion of log returns and percentage returns of the differences between two tickers in Figure 1 shows that the analysts can use either log daily return or percentage daily return for long-term period analysis.
2. The AIC (red line) and BIC (blue line) criteria are consistent of this analysis.
3. AIC or BIC drops dramatically when maxlag<=4 and approach a slope after maxlap>4. In the meantime, the running time of analysis with maxlag=12 increases sharply from 20.87s to 95.43s with maxlag=15.
4. Table 1, Table2, and Figure 2 consistently shows the best parameter p of AR(p) is 11 (percentage returns, no matter AIC or BIC criterion), or 11 (log returns with AIC criterion) or 10 (log returns with BIC criterion) with the consideration of computation cost.
5. Since the difference of AIC/BIC value between AR(10) and AR(11) are very small, **we conclude that the best fit of this AR model is AR(11).**

In [ ]:

```

```

Problem 2

Firstly, we solve OU process by changing variable $Y_t = X_t e^{\lambda t}$, then

$$dY_t = e^{\lambda t}\lambda\mu dt + \gamma e^{\lambda t} dW_t$$

and

$$X_t = X_0 e^{-\lambda t} + \mu(1 - e^{-\lambda t}) + \gamma \int_0^t e^{-\lambda(t-s)} dW_s$$

Multiply the above equation by $e^{\lambda \Delta t}$, we have

$$\begin{aligned}
e^{\lambda\Delta t} X_t &= X_0 e^{-\lambda(t-\Delta t)} + \mu(e^{\lambda\Delta t} - e^{-\lambda(t-\Delta t)}) + \gamma e^{\lambda\Delta t} \int_0^t e^{-\lambda(t-s)} dW_s \\
&= X_0 e^{-\lambda(t-\Delta t)} + \mu(1 - e^{-\lambda(t-\Delta t)}) + \mu(e^{\lambda\Delta t} - 1) + \gamma \int_0^{t-\Delta t} e^{-\lambda(t-\Delta t-s)} dW_s + \gamma \int_{t-\Delta t}^t e^{-\lambda(t-\Delta t-s)} \\
&= X_{t-\Delta t} + \mu(e^{\lambda\Delta t} - 1) + \gamma \int_{t-\Delta t}^t e^{-\lambda(t-\Delta t-s)} dW_s
\end{aligned}$$

Thus X_t can be rewritten as

$$X_t = e^{-\lambda\Delta t} X_{t-\Delta t} + \mu(1 - e^{-\lambda\Delta t}) + \gamma \int_{t-\Delta t}^t e^{-\lambda(t-s)} dW_s$$

Compare with the $AR(1)$ time series model $X_t = \alpha + \beta X_{t-1} + \epsilon_t$ and denote $\Delta t = 1$, we have

$$\begin{aligned}
\alpha &= \mu(1 - e^{-\lambda}) \\
\beta &= e^{-\lambda} \\
\sigma^2 &= \frac{\gamma(1 - e^{-2\lambda})}{2\lambda}
\end{aligned}$$

Problem 3

The $AR(2)$ model:

$$X_t = \alpha + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \varepsilon_t \quad \varepsilon_t \sim N(0, \sigma^2)$$

Let $u = \frac{\alpha}{1-\beta_1-\beta_2}$, $Y_t = X_t - u$, we have:

$$Y_t = \beta_1 Y_{t-1} + \beta_2 Y_{t-2} + \varepsilon_t$$

which can be recast in matrix form:

$$\begin{bmatrix} Y_t \\ Y_{t-1} \end{bmatrix} = \begin{bmatrix} \beta_1 & \beta_2 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} Y_{t-1} \\ Y_{t-2} \end{bmatrix} + \begin{bmatrix} \varepsilon_t \\ 0 \end{bmatrix}$$

and therefore is equivalent to an AR(1) model:

$$\xi_t = \mathbf{F}\xi_{t-1} + \varepsilon_t$$

We can re-write the formula recursively:

$$\xi_t = \mathbf{F}^t \xi_0 + (\varepsilon_t + \mathbf{F}\varepsilon_{t-1} + \cdots + \mathbf{F}^t \varepsilon_0)$$

By eigen-decomposition, we know that

$$\mathbf{F}^k = \mathbf{V}\boldsymbol{\Gamma}^{\mathbf{k}}\mathbf{V}^{-1}$$

where

$$\boldsymbol{\Gamma}^{\mathbf{k}} = \begin{bmatrix} \lambda_1^k & \\ & \lambda_2^k \end{bmatrix}$$

The stationary requires that $|\lambda_j| < 1$ $j = 1, 2$. The eigenvalues of $\mathbf{F}$ can be solved by:

$$det(\mathbf{F} - \lambda I) = 0 \Longrightarrow \lambda^2 - \beta_1\lambda - \beta_2 = 0$$

Then we have

$$\lambda_{1,2} = \frac{\beta_1 \pm \sqrt{\beta_1^2 + 4\beta_2}}{2}$$

(1) If the roots are real, for $|\lambda| < 1$,

$$\frac{\beta_1 + \sqrt{\beta_1^2 + 4\beta_2}}{2} < 1 \Rightarrow \sqrt{\beta_1^2 + 4\beta_2} < 2 - \beta_1 \Rightarrow \beta_1^2 + 4\beta_2 < (2-\beta_1)^2 \Rightarrow \beta_1 + \beta_2 < 1$$

$$\frac{\beta_1 - \sqrt{\beta_1^2 + 4\beta_2}}{2} > -1 \Rightarrow -\sqrt{\beta_1^2 + 4\beta_2} > -2 - \beta_1 \Rightarrow \beta_1^2 + 4\beta_2 < (2 + \beta_1)^2 \Rightarrow \beta_2 - \beta_1 < 1$$

And $\lambda_1 \lambda_2 = -\beta_2 < 1 \Rightarrow \beta_2 > -1$

(2) If the roots are complex, $\beta_1^2 + 4\beta_2 < 0$

$$\lambda_{1,2} = \frac{\beta_1}{2} \mp \frac{\sqrt{\beta_1^2 + 4\beta_2}}{2} i$$

to be inside the unit circle:

$$\frac{\beta_1^2}{4} - \frac{\beta_1^2 + 4\beta_2}{4} = -\beta_2 < 1 \Rightarrow \beta_2 > -1$$

So we conclude that, the conditions for $AR(2)$ model to be covariance stationary are:

$$(1)\ \beta_2 + \beta_1 < 1$$
$$(2)\ \beta_2 - \beta_1 < 1$$
$$(3)\ \beta_2 > -1$$