Workshop 7: Time Series (25pts)

Task 1: Loading the data – 5 pts

In this part, you will load the data from the file *volume_per_year.csv*. There are dates and market volume across the years. You can load this file into the variable *volume*.

```
print (volume.head())
     Month  volume
0  1949-01   22400
1  1949-02   23600
2  1949-03   26400
3  1949-04   25800
4  1949-05   24200
```

Be aware, when you load a file, the dates are loaded as strings. You will need to use *read_cvs* wisely.

Task 2: Stationarity – 5 pts

A common assumption in many time series techniques is that the data are stationary.

A stationary process has the property that the mean, variance and autocorrelation structure do not change over time.

Questions:

A- Plot the volume across the years.

B- What do you deduce from the plot?

C- Testing stationarity

To test stationarity, we can use the Dickey-Fuller test or Rolling statistics (such as Moving Average and Moving variance)

Step1: Calculate the moving average with a window of 1 year. Store into a variable *ma*

Step2: Calculate the moving standard deviation with a window of 1 year. Store into a variable *msd*

Step3: Plot on the same graph:

Volume (blue), ma (green) and msd (red)

Step4: What do you conclude?

Step5: Using the package `from statsmodels.tsa.stattools import adfuller`

You will confirm your conclusion of the Step4 by finding this ouput:

```
In [21]: print (adtestoutput)
Test Statistic                  0.815369
p-value                         0.991880
#Lags Used                     13.000000
Number of Observations Used    130.000000
Critical Value (1%)             -3.481682
Critical Value (10%)            -2.578770
Critical Value (5%)             -2.884042
```

What is the null hypothesis of the Dickey-Fuller test?
What do you conclude?

Task 3: Make a Time Series stationary – 5pts

If the time series is not stationary, we can often transform it to stationarity with one of the following techniques.

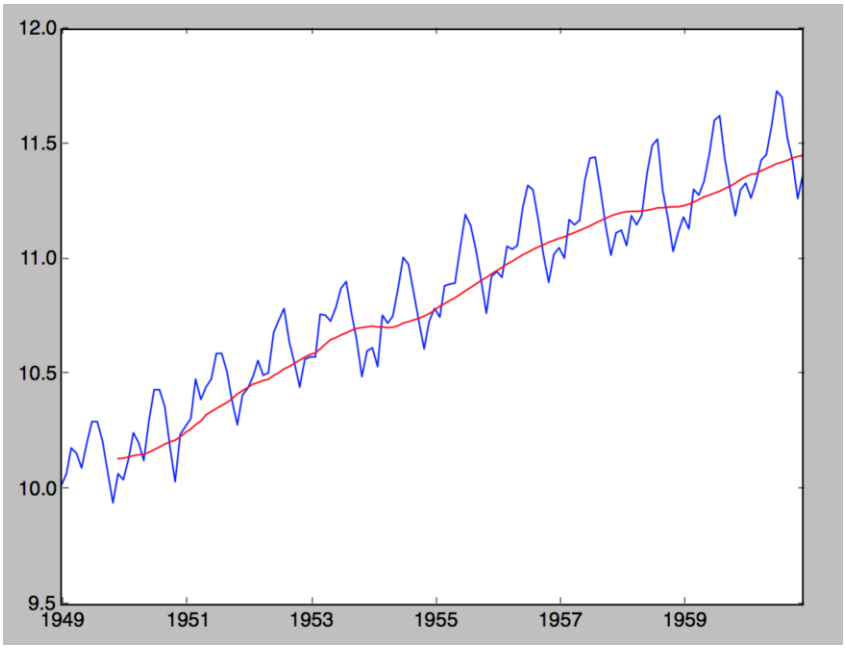
1- *We can difference the data.*
That is, given the series Z_i we create the new series
$Y_i = Z_i - Z_{i-1}$

The differenced data will contain one less point than the original data. Although you can difference the data more than once, one difference is usually sufficient.

2- *For non-constant variance, taking the logarithm or square root of the series may stabilize the variance. For negative data, you can add a suitable constant to make all the data positive before applying the transformation. This constant can then be subtracted from the model to obtain predicted (i.e., the fitted) values and forecasts for future points.*

Questions:

A- We are going to try to eliminate the trend previously observed.
Plot the logarithm of the volume.
B- What do you observe?
C- We are now going to try to smooth the data.
Store the logarithm of the volume data into a variable *logvolume*
Store the moving average with a 1-year window into a variable *mavolume*
Plot the graph representing *logvolume* and *mavolume.*



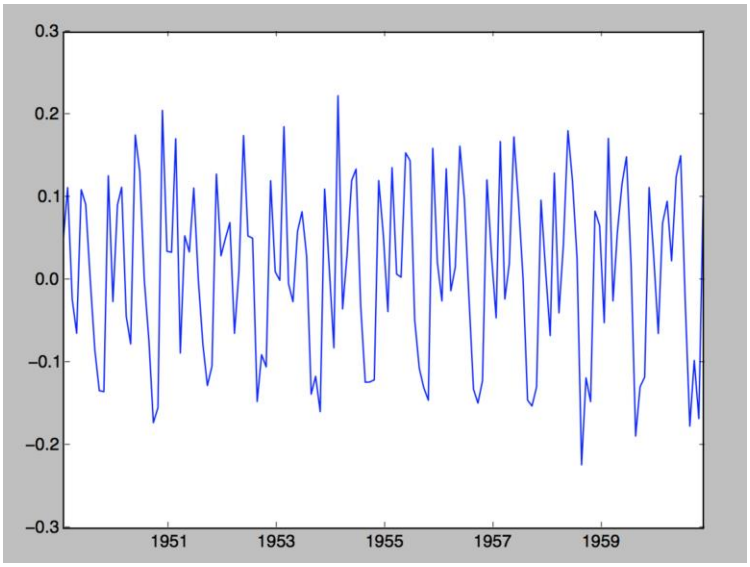
The red shows the trend. You just need to subtract *logvolume* – *mavolume* and store it into volume_without_trend.

D- Retest stationarity the same way as you did in the task 2.
E- Redo the study with an exponentially weighted movin average with a half period of one year. pd.ewma(your_data,halflife=12)
F- Retest stationarity for ewma.
G- What do you conclude with this different method

Task 4: Removing trend and seasonality with differencing – 5pts

Questions:

A- Remove the stationarity apply differencing to the log volume data.
You will need to use the function shift.
B- Plot the graph



C- Test stationarity

Task5: Forecast Time Series – 5pts

https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
ARIMA (Auto-Regressive Integrated Moving Averages) forecasting for a stationary time series is a linear regression equation.
Predictors depend on the parameters (p Number of AR terms ,d Number of Differences,q Number of MA terms) of the ARIMA model.

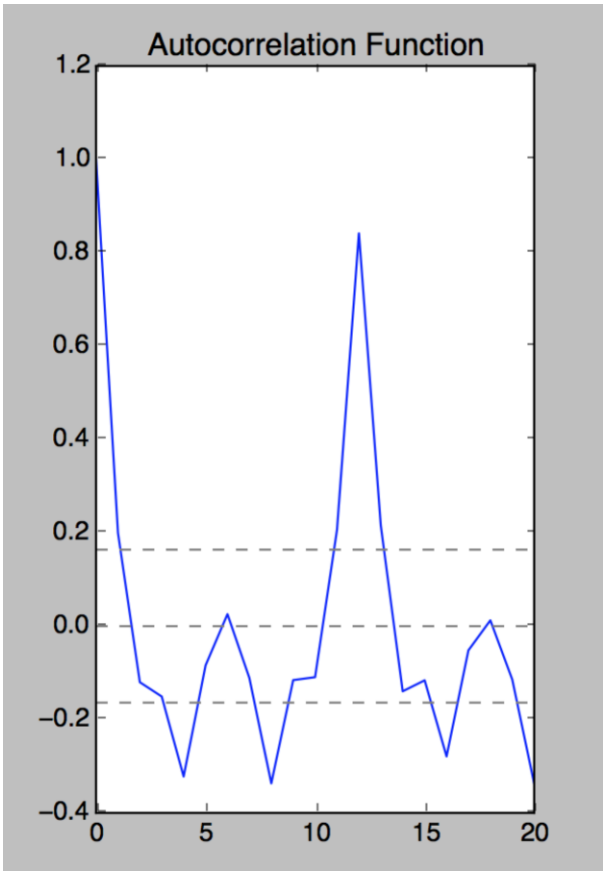
A- You need to study the ACF

Use the package:

```
from statsmodels.tsa.stattools import acf, pacf
```

Calculate the acf of the diff log volume obtained in the previous section.
Be aware of removing the non-defined values. If you don't do that, acf will return NAs. You can use the function dropna to remove these undefined values.

```
plt.subplot(121)
plt.plot(acf)
plt.axhline(y=0,linestyle='--',color='gray')
plt.axhline(y=-1.96/np.sqrt(len(volume_log_diff)),linestyle='--',color='gray')
plt.axhline(y=1.96/np.sqrt(len(volume _log_diff)),linestyle='--',color='gray')
plt.title('Autocorrelation Function')
```



B- Finally you will load the library

```
from statsmodels.tsa.arima_model import ARIMA
```

C- You will run the ARIMA model using p=2, d=1, q=2 on the log date (not differentiated since d=1). You can store the result of this function into the variable *model*

D- You will store the result of model.fit(disp=-1) into *results_ARIMA*

E- You will plot the log volume with *results_ARIMA.*

F- We need to convert the predicted values into the original scale one

```
predictions_ARIMA_diff = pd.Series(results_ARIMA.fittedvalues, copy=True)
```

```
print (predictions_ARIMA_diff.head())
Month
1949-02-01    0.009580
1949-03-01    0.017491
1949-04-01    0.027670
1949-05-01   -0.004521
1949-06-01   -0.023890
```

Find the function converting diff values into real one. (you should be able to use cumsum)

```
predictions_ARIMA_diff_cumsum.head())
Month
1949-02-01    0.009580
1949-03-01    0.027071
1949-04-01    0.054742
1949-05-01    0.050221
1949-06-01    0.026331
```

G- Apply exponential to go back to the initial scale