Time series Analysis and Modeling

DATS 6313

LAB # 2

Autocorrelation Function (ACF)

In this LAB you will learn how to implement the Autocorrelation function program using python and apply to the real-world data. Set the numpy random seeds to 6313.

Using the Python program and the following packages in Python perform the following tasks :

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

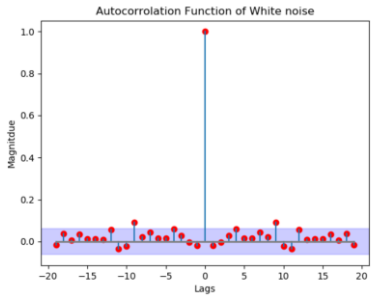
1- Let suppose a y vector is given as y(t) = [3, 9, 27, 81,243]. Without use of python or any other computer program, manually calculate the ACF for lag 0,1,2,3, and 4 ($\widehat{R_y}(0), \widehat{R_y}(1), \widehat{R_y}(2), \widehat{R_y}(3), \widehat{R_y}(4)$). Hint: The formula for the ACF calculation is given bellow. Display the ACF (two sided) on a graph (no python).

$$\hat{R}_y(\tau) = \frac{\sum_{t=\tau+1}^{T}(y_t - \overline{y})(y_{t-\tau} - \overline{y})}{\sum_{t=1}^{T}(y_t - \overline{y})^2}$$

2- Using Python program, create a time series dataset with normal white noise (zero mean and standard deviation of 1) and 1000 samples. The date range from Jan 1st 2000 through December 31st 2000. Plot the generated time series dataset versus time. Plot the histogram of generated WN. Calculate the sampled mean and sampled std of generated WN. You can use the following command to generate WN~(0,1): (T # of samples)

```
np.random.normal(mean, std, size=T)
```

3- Write a python code to estimate Autocorrelation Function. Note: You need to implement the provided equation for ACF in the lecture notes (see above). Hint: ACF plot must be double sided from negative # of lags to positive # of lags with highlighted insignificant region. An example of ACF function is show bellow:



a. Plot the ACF of the make-up dataset in question 1 and compare the result with the manual calculation. # of lags = 4.
b. Plot the ACF of the generated data in question 2. The ACF needs to be plotted using "stem" command. # of lags = 20.
c. Write down your observations about the ACF plot of stationary dataset.

4- Load the time series dataset from yahoo API. The yahoo API contains the stock value for 6 major companies.

```
stocks = ['AAPL','ORCL', 'TSLA', 'IBM','YELP', 'MSFT']
```

a. Plot the "Close" value of the stock for all companies versus time in one graph, subplot [3 figures in row and 2 figures in column]. Add grid, x-label, y-label, and title to each subplot. (figure size =16,8)

You will need the following package to be able to connect to yahoo API. Make sure to use the updated version of the pandas and pandas' data_reader (You can use the "pip install --upgrade pandas" and "pip install --upgrade pandas-datareader"). Pick the start date as '2000-01-01' and the end date is Jan 31st, 2023.

```
from pandas_datareader import data
 import yfinance as yf
 yf.pdr_override()

df = data.get_data_yahoo('AAPL', start='', end='')
```

b. Plot the ACF of the "Close" value of the stock for all companies versus lags in one graph, subplot [3 rows and 2 columns]. Add x-label, y-label, and title to each subplot. The number lags 50.

5- Write down your observations about the correlation between stationary and non-stationary time series (if there is any) and autocorrelation function?

Upload a formal **report (as a single pdf)** plus **the .py file** through BB by the due date.