**DATS 6313 – Time Series Analysis & Modeling**

Instructor: Reza Jafari

**Lab #7**

Bradley Reardon

3/29/2022

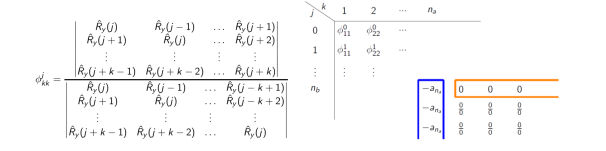
**Abstract:**

The lab is about Generalized Partial Autocorrelation table for Autoregressive (AR) & Moving Average (MA) Model.

**Introduction:**

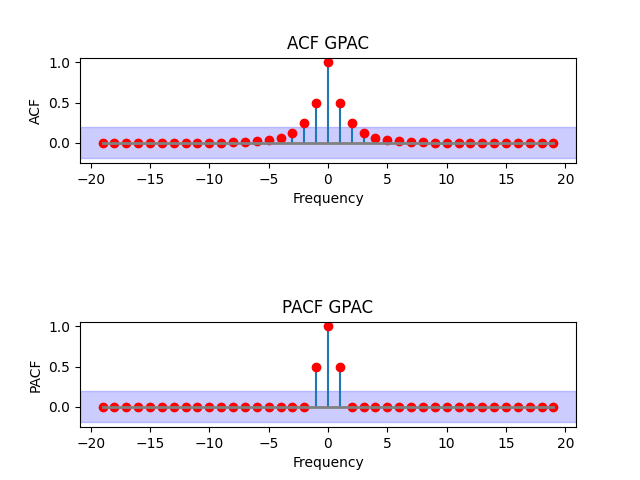
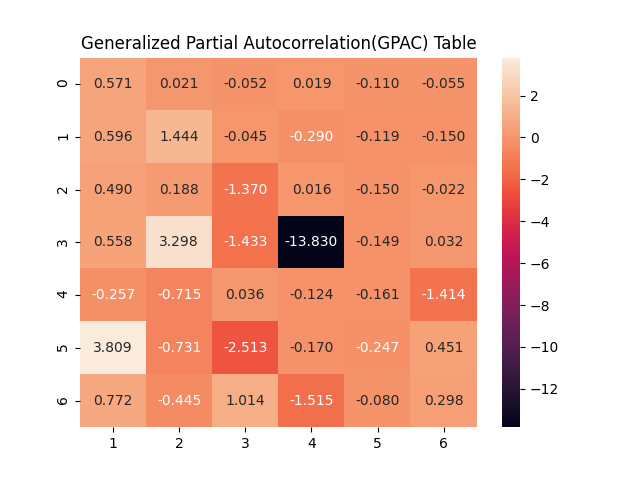
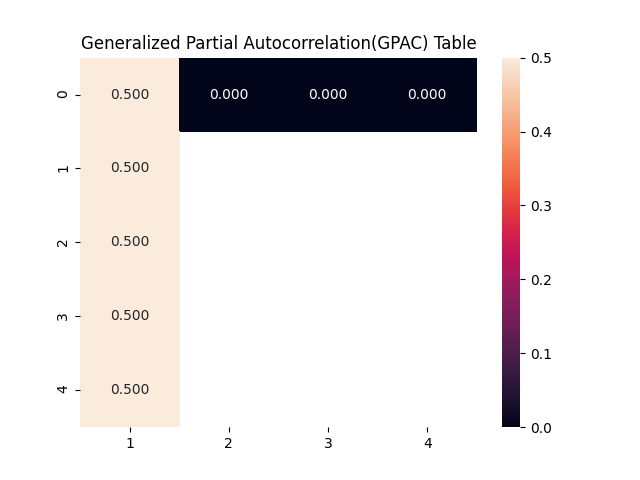
This experiment was performed to increase understanding of the application of GPAC and how the table can be converted into a python program.

**Method, Theory, and Procedures:**

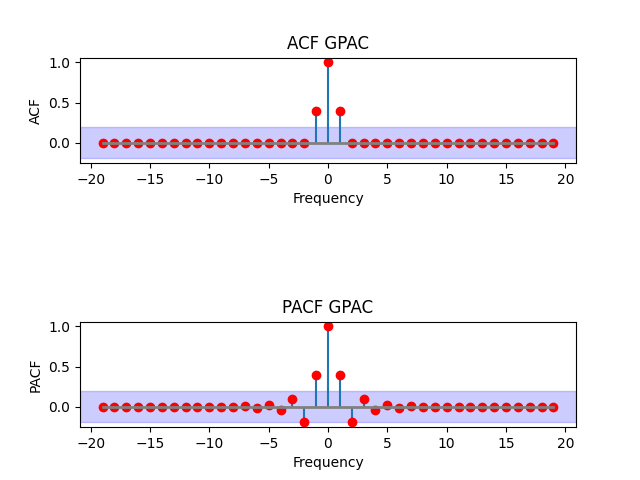
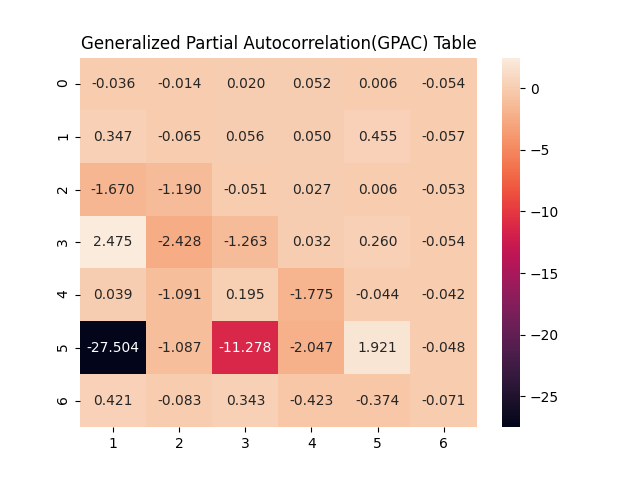
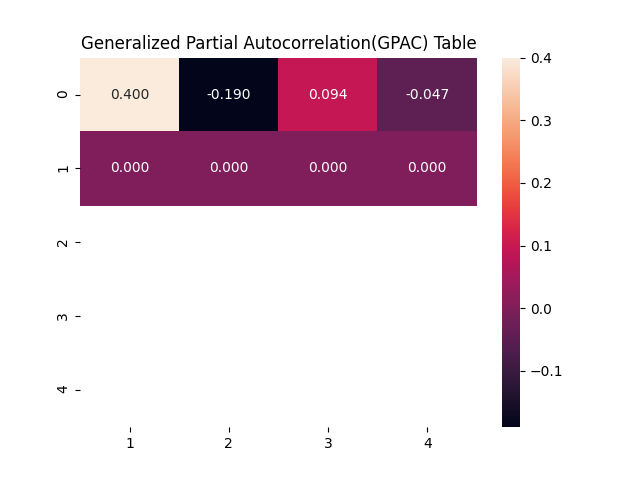
GPAC Table: 

**Answers to Lab Questions:**

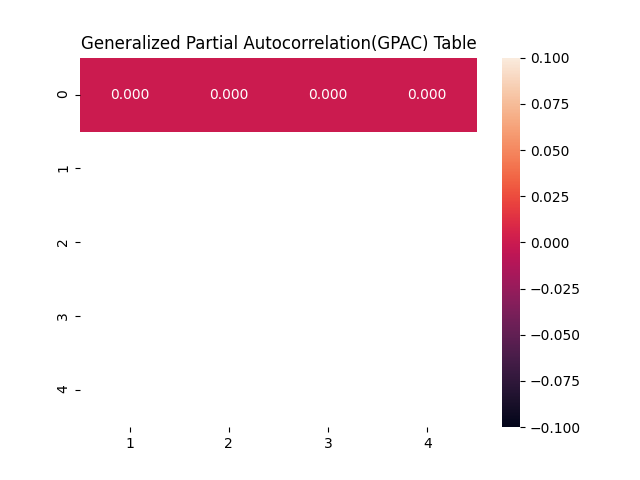
Example 1: 𝑦(𝑡) − 0.5𝑦(𝑡 − 1) = 𝑒(𝑡) ARMA (1,0)

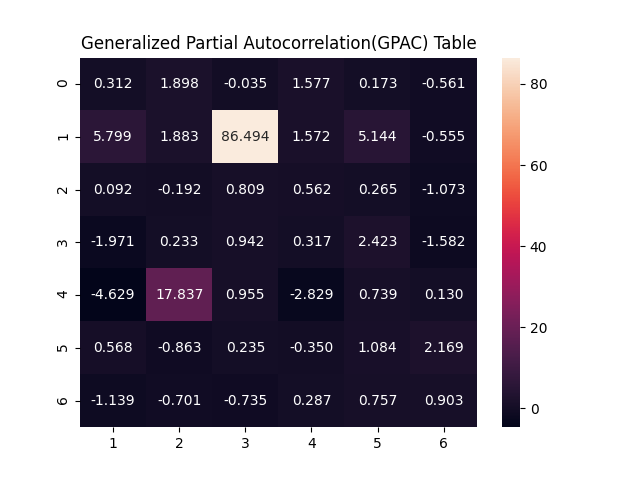
****

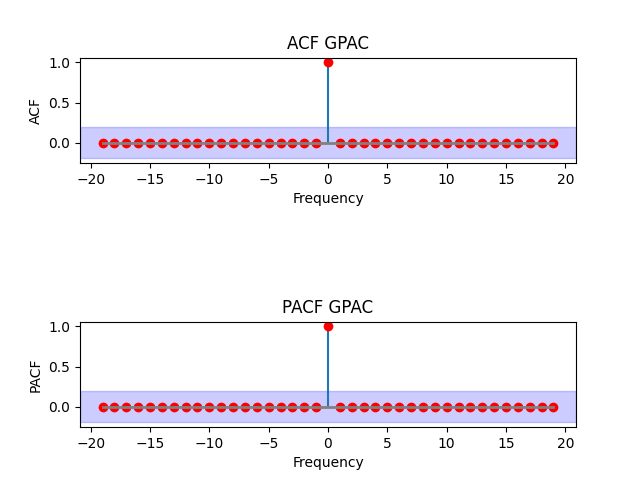
Example 2: ARMA (0,1): y(t) = e(t) + 0.5e(t-1)

****

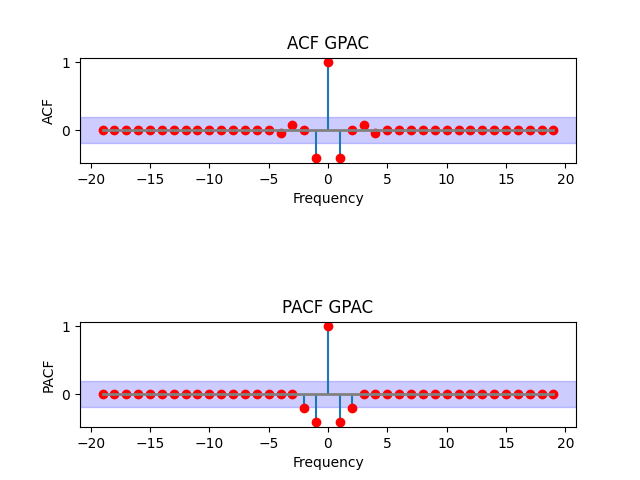
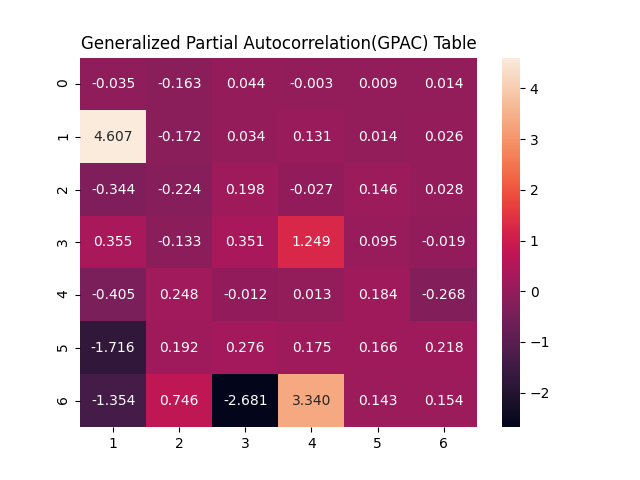
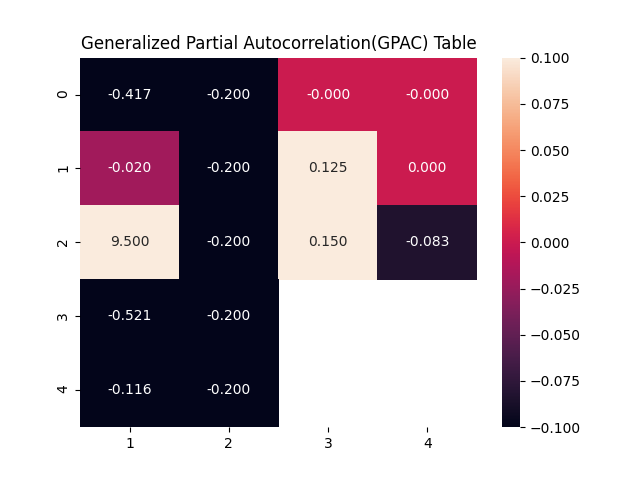
Example 3: ARMA (1,1): y(t) + 0.5y(t-1) = e(t) + 0.5e(t-1)



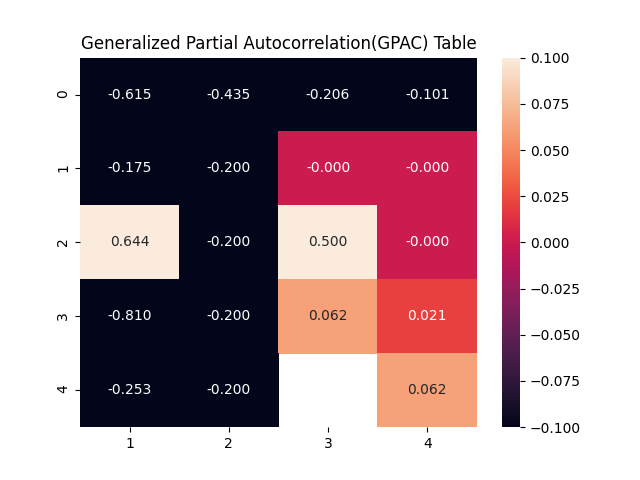


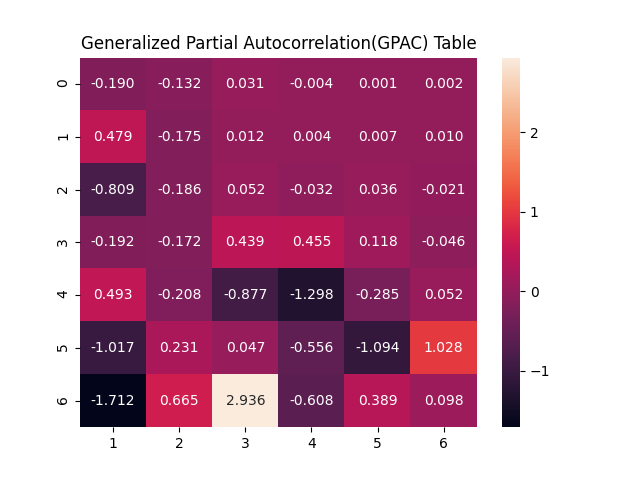


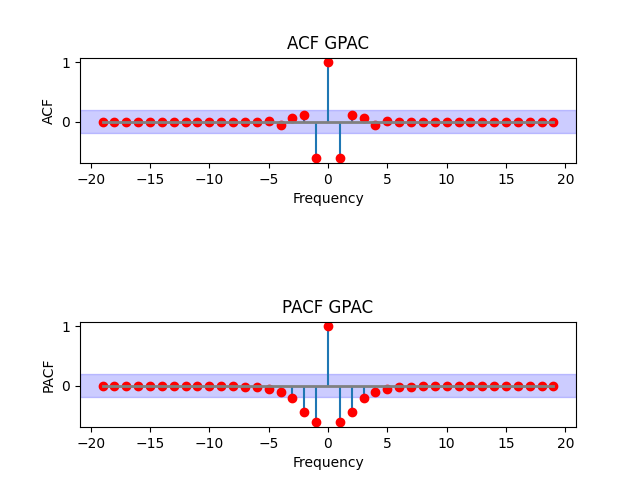
Example 4: ARMA (2,0): y(t) + 0.5y(t-1) + 0.2y(t-2) = e(t)



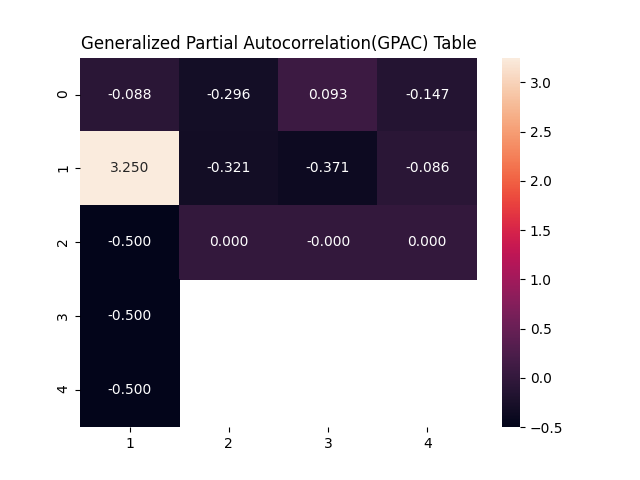
Example 5: ARMA (2,1): y(t) + 0.5y(t-1) + 0.2y(t-2) = e(t) - 0.5e(t-1)

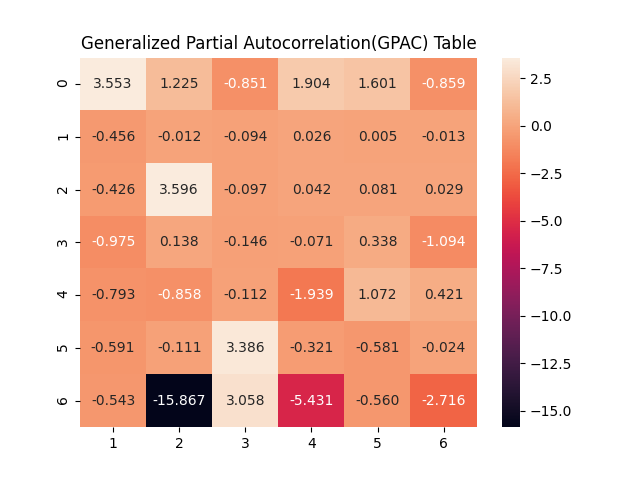


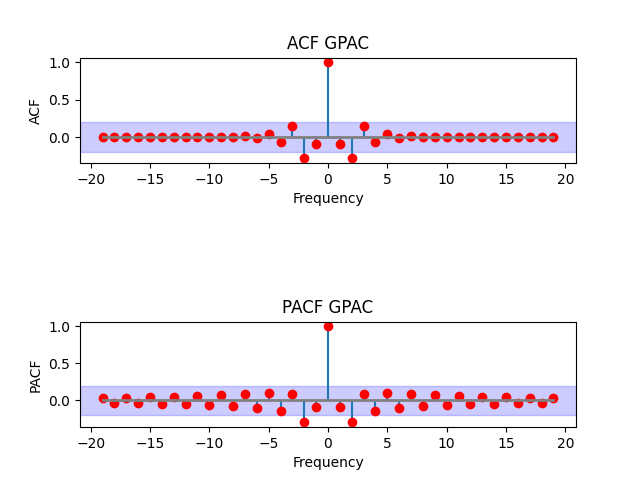




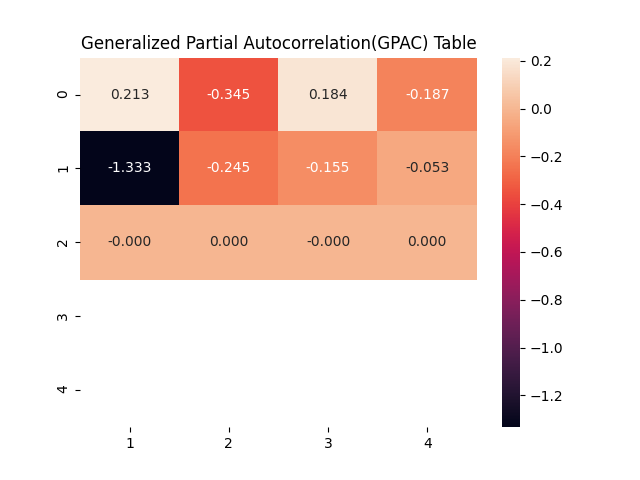
Example 6: ARMA (1,2): y(t) + 0.5y(t-1) = e(t) + 0.5e(t-1) - 0.4e(t-2)

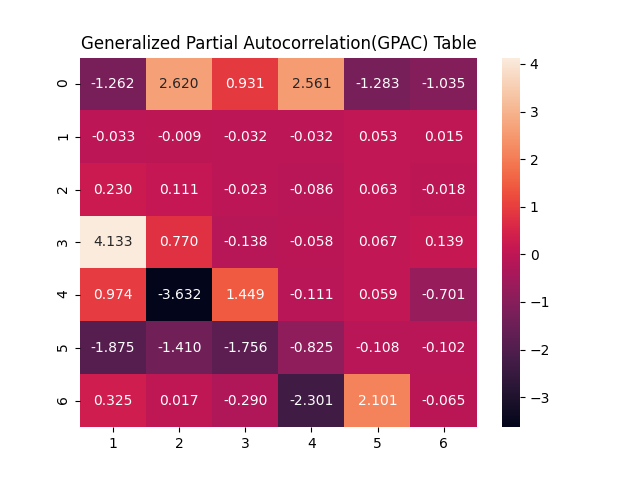
****

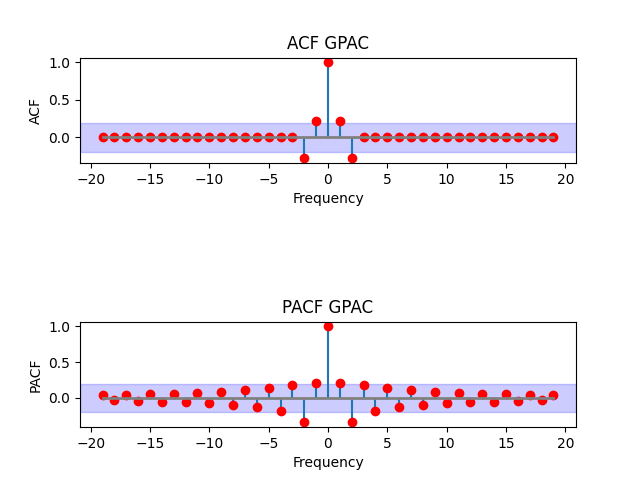
****

****

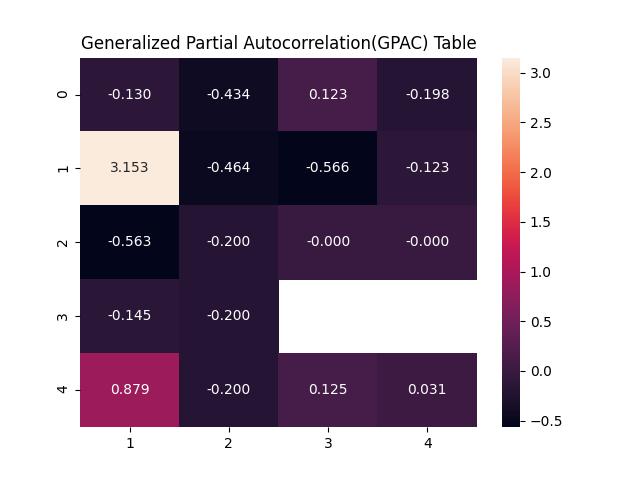
Example 7: ARMA (0,2): y(t) = e(t) + 0.5e(t-1) - 0.4e(t-2)

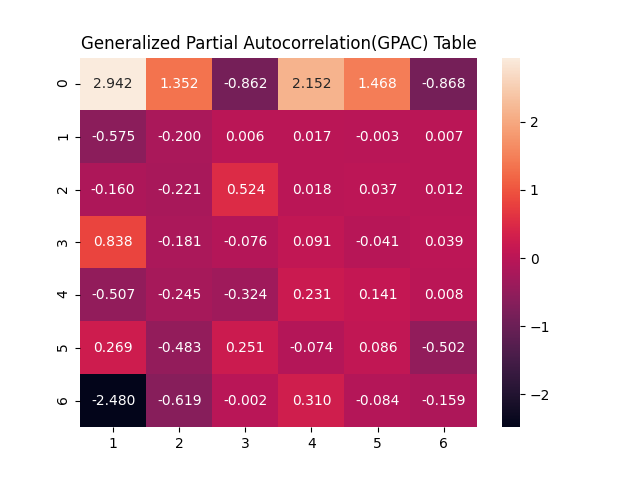
****

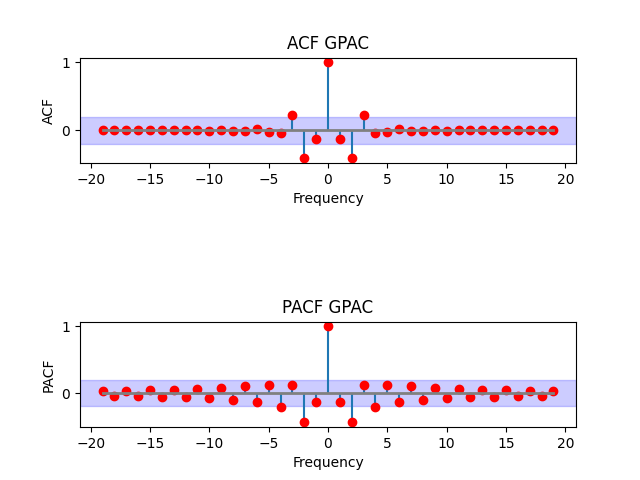
****

****

Example 8: ARMA (2,2): y(t)+0.5y(t-1) +0.2y(t-2) = e(t)+0.5e(t-1) - 0.4e(t-2)

****

****

****

**8.** It seems that higher order ARMA models have less static ACF and PACF graphs.

**Conclusion:**

GPAC charts can be used to determine the order of an ARMA model.

**Appendix:**

import numpy as np  
import statsmodels.api as sm  
from matplotlib import pyplot as plt  
import toolbox  
  
print("====================QUESTION 1==========================")  
samples\_size = input("Enter # of samples: ")  
mean = input("Enter Mean (WN): ")  
var = input("Enter Variance (WN): ")  
ar\_order = input("Enter AR order: ")  
ma\_order = input("Enter MA order: ")  
  
ar\_inputs = []  
if int(ar\_order) > 0:  
 for i in range(int(ar\_order)):  
 text = "Enter a" + str(i + 1) + ": "  
 input\_value = input(text)  
 ar\_inputs.append(float(input\_value))  
  
ma\_inputs = []  
if int(ma\_order) > 0:  
 for i in range(int(ma\_order)):  
 text = "Enter b" + str(i + 1) + ": "  
 input\_value = input(text)  
 ma\_inputs.append(float(input\_value))  
  
ar = np.r\_[1, ar\_inputs]  
ma = np.r\_[1, ma\_inputs]  
  
samples\_size = int(samples\_size)  
ar\_order = int(ar\_order)  
ma\_order = int(ma\_order)  
mean = float(mean)  
var = float(var)  
  
arma\_process = sm.tsa.ArmaProcess(ar, ma)  
mean\_y = mean \* (1 + np.sum(ma\_inputs)) / (1 + np.sum(ar\_inputs))  
y = arma\_process.generate\_sample(samples\_size, scale=np.sqrt(var) + mean\_y)  
  
  
acf\_lags = 60  
ry = arma\_process.acf(lags=acf\_lags)  
toolbox.gpac\_calc(ry, 5, 5)  
  
acf\_lags = 15  
new\_ry = toolbox.auto\_correlation\_cal(y, acf\_lags)  
toolbox.gpac\_calc(new\_ry, 7, 7)  
  
lags = 20  
  
acf = arma\_process.acf(lags=lags)  
pacf = arma\_process.pacf(lags=lags)  
  
fig, axs = plt.subplots(2, 1)  
fig.subplots\_adjust(hspace=1.5, wspace=0.5)  
axs = axs.ravel()  
  
ry = arma\_process.acf(lags=lags)  
a1 = ry  
a2 = a1[::-1]  
a = np.concatenate((a2[:-1], a1))  
x1 = np.arange(0, lags)  
x2 = -x1[::-1]  
x = np.concatenate((x2[:-1], x1))  
(marker, stemlines, baselines) = axs[0].stem(x, a,  
 use\_line\_collection=True, markerfmt='o')  
plt.setp(marker, color='red', marker='o')  
plt.setp(baselines, color='gray', linewidth=2, linestyle='-')  
m = 1.96 / np.sqrt(100)  
axs[0].axhspan(-m, m, alpha=.2, color='blue')  
axs[0].set\_title("ACF GPAC")  
axs[0].set\_ylabel("ACF")  
axs[0].set\_xlabel("Frequency")  
  
ry = arma\_process.pacf(lags=lags)  
a1 = ry  
a2 = a1[::-1]  
a = np.concatenate((a2[:-1], a1))  
x1 = np.arange(0, lags)  
x2 = -x1[::-1]  
x = np.concatenate((x2[:-1], x1))  
(marker, stemlines, baselines) = axs[1].stem(x, a,  
 use\_line\_collection=True, markerfmt='o')  
plt.setp(marker, color='red', marker='o')  
plt.setp(baselines, color='gray', linewidth=2, linestyle='-')  
m = 1.96 / np.sqrt(100)  
axs[1].axhspan(-m, m, alpha=.2, color='blue')  
axs[1].set\_title("PACF GPAC")  
axs[1].set\_ylabel("PACF")  
axs[1].set\_xlabel("Frequency")  
  
plt.show()  
  
  
samples\_size = 5000  
arma\_process = sm.tsa.ArmaProcess(ar, ma)  
mean\_y = mean \* (1 + np.sum(ma\_inputs)) / (1 + np.sum(ar\_inputs))  
arma\_process.generate\_sample(samples\_size, scale=np.sqrt(var) + mean\_y)  
  
  
acf\_lags = 60  
ry = arma\_process.acf(lags=acf\_lags)  
toolbox.gpac\_calc(ry, 5, 5)  
  
  
samples\_size = 10000  
arma\_process = sm.tsa.ArmaProcess(ar, ma)  
mean\_y = mean \* (1 + np.sum(ma\_inputs)) / (1 + np.sum(ar\_inputs))  
arma\_process.generate\_sample(samples\_size, scale=np.sqrt(var) + mean\_y)  
  
acf\_lags = 60  
ry = arma\_process.acf(lags=acf\_lags)  
toolbox.gpac\_calc(ry, 5, 5)