Assignment 9

Submitted by Dhyan Patel

Date : 3/31/2023

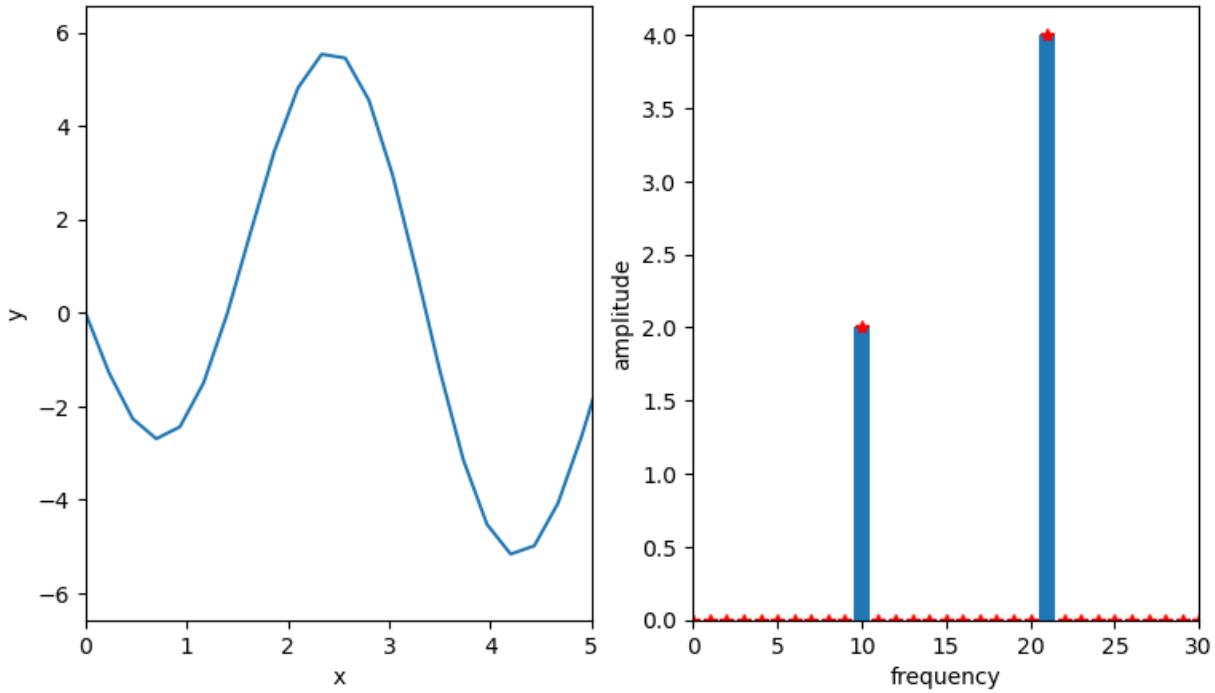
In [1]:
```
# importing the ncessary modules
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import scipy as sp
```

Question 1

In [2]:
```
# defining the function for plotting a magnitude vs frequency graph of 2.0*np.sin(2*np

def ffthw(xlow,xhigh,barwidth=0.2):
    N = 300
    T=1/N
    x = np.linspace(xlow,xhigh,N,endpoint=False)

    y = 2.0*np.sin(2*np.pi*x/7)-4.0*np.sin(3*np.pi*x/5)

    fig,ax = plt.subplots(1,2)
    fig.set_size_inches([9,5])
    ax[0].plot(x,y)
    ax[0].set_xlabel('x')
    ax[0].set_ylabel('y')
    ax[0].set_xlim(0,5)
    amp = abs(np.fft.fft(y))[:N//2]

    freq = np.fft.fftfreq(N,T)[:N//2]

    ax[1].bar(freq, 2*amp/N, width = barwidth)
    ax[1].plot(freq, 2*amp/N, '*r')
    ax[1].set_xlabel('frequency')
    ax[1].set_ylabel('amplitude')
    ax[1].set_xlim(0,30)

    return
```

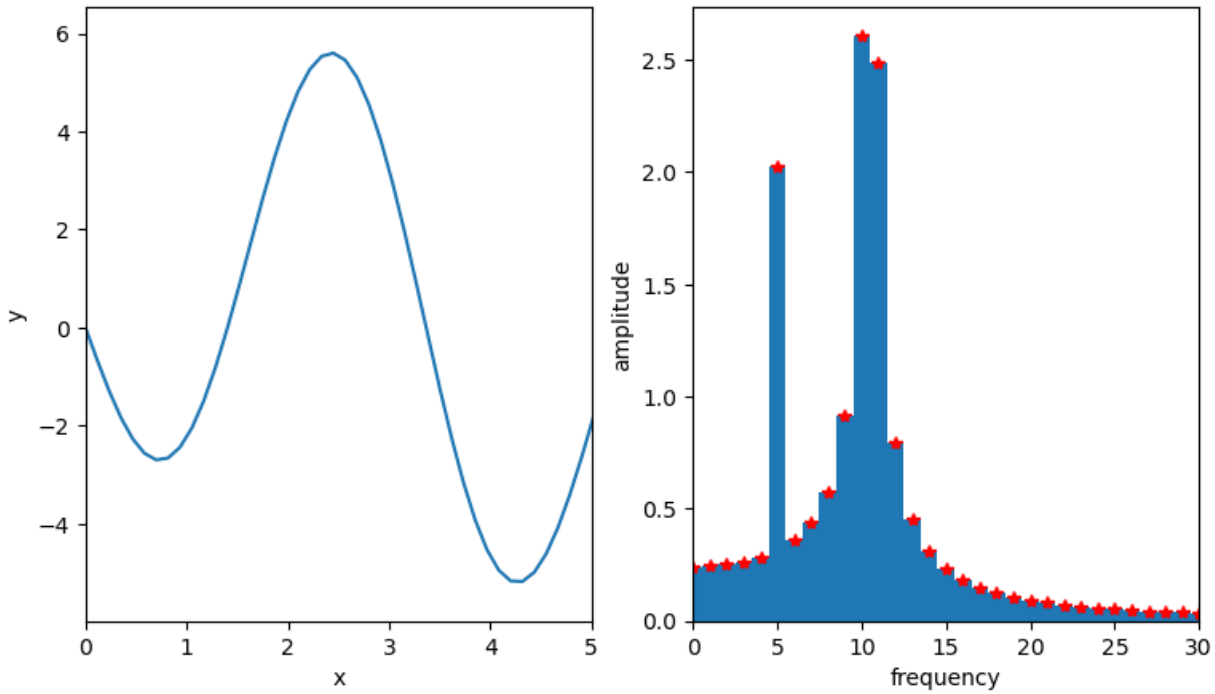
Part A

In [3]:
```
# calling the function to get the 2 frequencies

ffthw(0,70, barwidth = 1)
```



Part B

In [4]:
```
# calling the function with incorrect values to spread the frequency vs magnitude grap

ffthw(0,35, barwidth = 1)
```



Question 1

In [5]:
```
# calling in the provided data

d1 = pd.read_table('CupData1.dat' , sep = '[,]' , usecols = [0] , header = None , engi
```

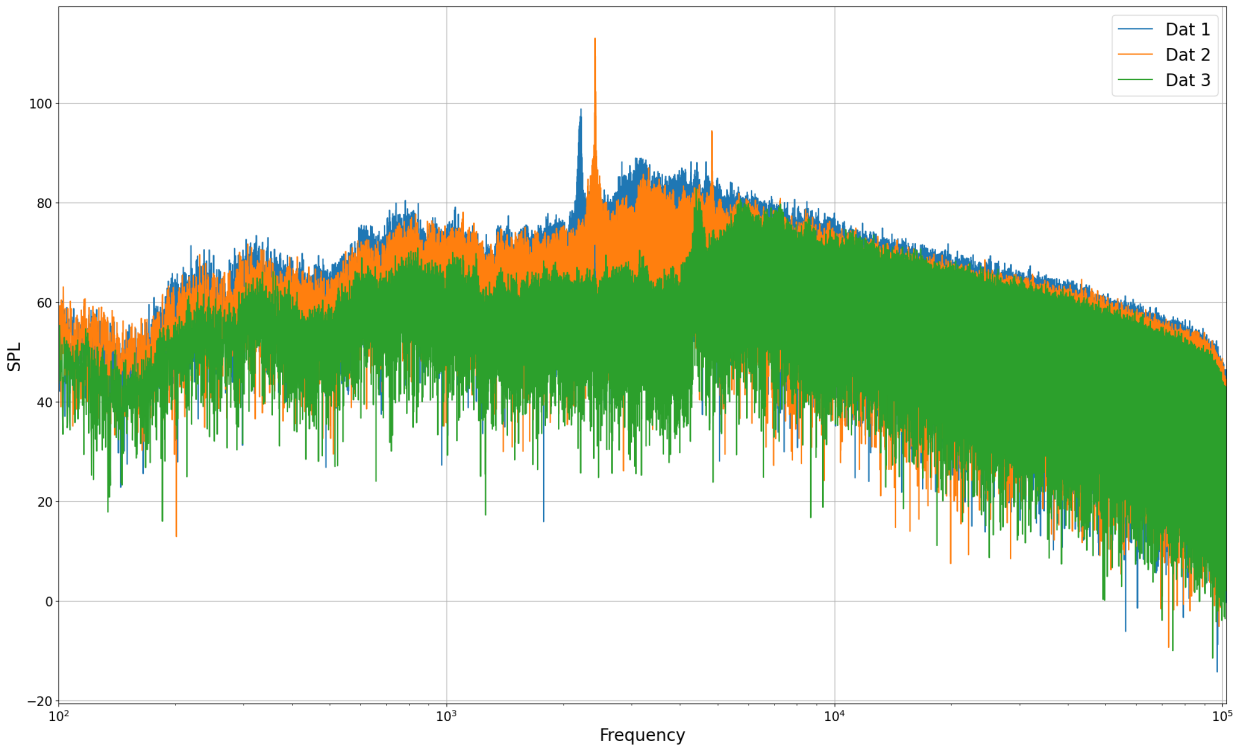
```
d2 = pd.read_table('CupData2.dat' , sep = '[,]' , usecols = [0] , header = None , engi
d3 = pd.read_table('CupData3.dat' , sep = '[,]' , usecols = [0] , header = None , engi
```

In [6]:
```
# converting into array for processing purposes

d1 = np.array(d1, dtype = float)
d2 = np.array(d2, dtype = float)
d3 = np.array(d3, dtype = float)
```

In [7]:
```
#initial parameters
totaltime = 5 # total time taken
bucksize = 4096 # the size of each bucket
totaldatp = len(d1) # number of data points in each column of the data
bucks = int(totaldatp/bucksize) # the number of buckets created after dividing them in
t = totaltime/totaldatp # time period
p = 0.00002 # conversion factor for pascals to decibels
tf = sp.fft.fftfreq(totaldatp,t)[:totaldatp//2] # the fft frequency created for all th
tdf = sp.fft.fftfreq(bucksize,t)[:bucksize//2] # the fft created for each bucket that
```

Part A

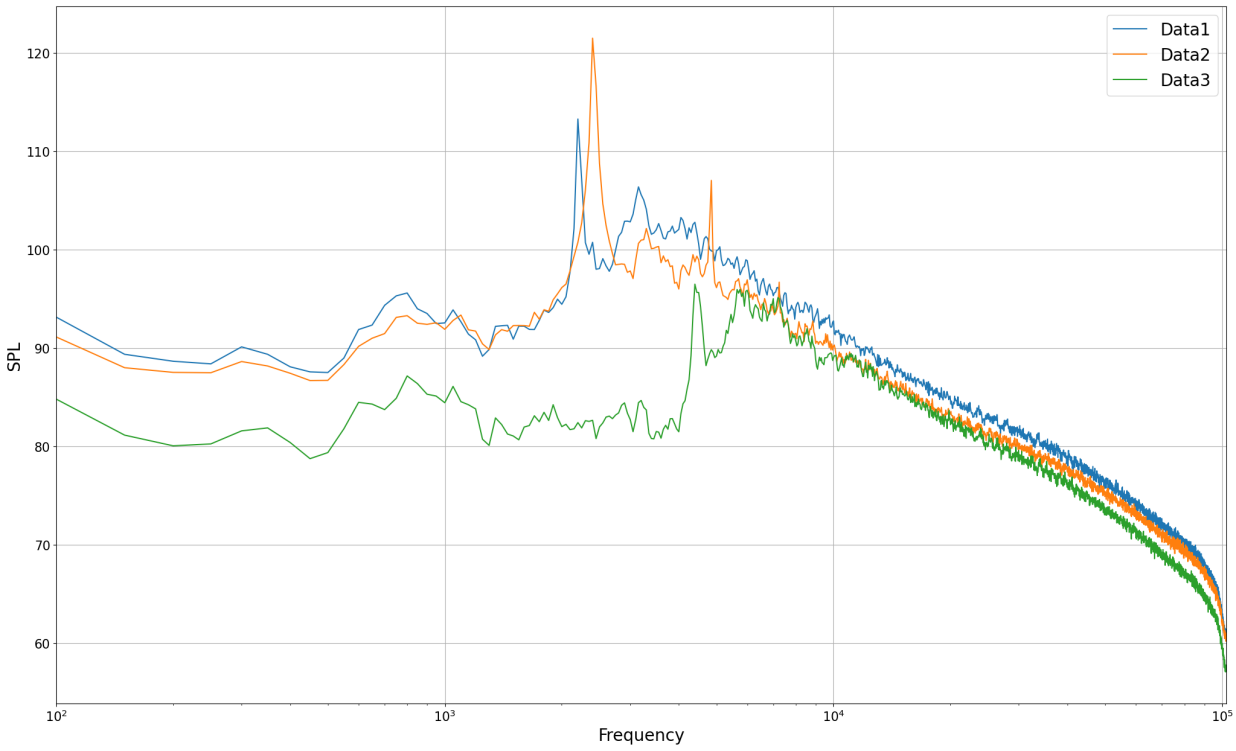
In [8]:
```
# plotting the graph
fig,ax = plt.subplots(figsize = (25,15))

# here in the plotting function itself we have taken the fft of the 2D matrix, then ta
# divided the data in half along with multiplying it by a factor of 2/(number of data
# SPL values

#for CupData1
ax.plot(tf,20*np.log10(((2/totaldatp)*(np.abs(sp.fft.fft2(d1)))[:totaldatp//2])/p), la

#for CupData2
ax.plot(tf,20*np.log10(((2/totaldatp)*(np.abs(sp.fft.fft2(d2)))[:totaldatp//2])/p), la

#for CupData3
ax.plot(tf,20*np.log10(((2/totaldatp)*(np.abs(sp.fft.fft2(d3)))[:totaldatp//2])/p), la

# setting the scale
ax.set_xscale('log')
# setting the limits
ax.set_xlim(10**2,tf[-1])
ax.grid("True")
ax.tick_params(axis='both', which='major', labelsize=15)
ax.legend(fontsize = 20)
ax.set_ylabel("SPL", fontsize = 20)
ax.set_xlabel("Frequency", fontsize = 20)
plt.show()
```



Part B

In [9]:

```
# plotting the graph

fig,ax = plt.subplots(figsize = (25,15))

# the first thing we have done here is to split the function into 250 equal parts then
# created then we took its absolute value along with reshaping into a matrix of 250 ro
# square of all the values and took sum of those values and then divided by 250 i.e. t
# took a squareroot of the matrix thus created, then converted it in SPL values and pl


# for CupData1
ax.plot(tdf,20.0*np.log10(((np.sqrt((np.square((2/bucksize)*(np.abs(sp.fft.fft2(np.spl

# for CupData2
ax.plot(tdf,20.0*np.log10(((np.sqrt((np.square((2/bucksize)*(np.abs(sp.fft.fft2(np.spl

# for CupData3
ax.plot(tdf,20.0*np.log10(((np.sqrt((np.square((2/bucksize)*(np.abs(sp.fft.fft2(np.spl

# setting the scale
ax.set_xscale('log')

# setting the limits
ax.set_xlim(10**2,tdf[-1])
ax.grid('True')
ax.tick_params(axis='both', which='major', labelsize=15)
ax.legend(fontsize = 20)
ax.set_ylabel("SPL", fontsize = 20)
ax.set_xlabel("Frequency", fontsize = 20)
plt.show()
```


Data1
Data2
Data3
SPL
Frequency
120
110
100
90
80
70
60
10^2
10^3
10^4
10^5

In [ ]: