2110233 Com Eng Math Lab - (1/2022) Take Home Quiz

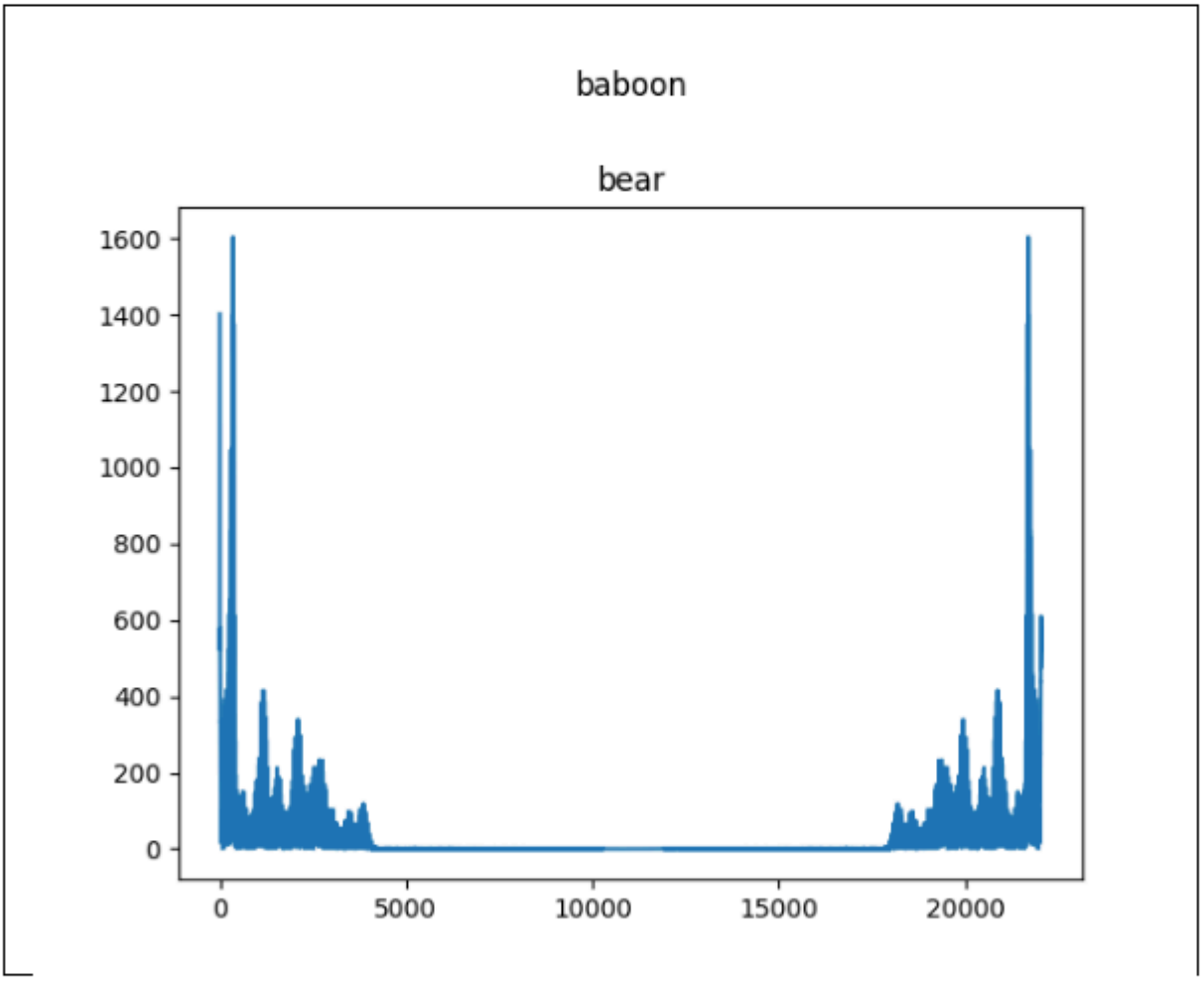
Submission: (1) pdf file and (2) three audio files before 23:59, 27 October 2022

1. (20 points) Download three different animal sounds from

- https://mixkit.co/free-sound-effects/animals/ or
- https://drive.google.com/drive/folders/1Yf5rG3MvCO04uFxEJErU0oYpmAmQ9U3G?usp=sharing or
- anywhere or you can collect it yourself

use DFT to visualize and analyze the spectrums of each audio. You should compare the spectrums on the same scale. Each audio should be less than 5 seconds.

1.1) Display Fourier Spectrum for each animal sound



1.2) Analyze the spectrum for each animal sound

baboon.mp3 has a maximum magnitude at 171.21 hz
bark.mp3 has a maximum magnitude at 511.42 hz, 2149.96hz
bear-roar.mp3 has a maximum magnitude at 21697.82 hz

1.3) Put your code here or explain how you compare or analyze the spectrum (You can use any tools and library, just explain in this section).

```
   magnitude = np.abs(np.fft.fft(signal))
    frequency=np.linspace(0,22050,len(magnitude))
    plt.plot (frequency,magnitude)
    plt.title(title)
    plt.savefig(title+'.png)
# Use np.fft.fft to obtain the fft spectrum and plot the
graph between fft spectrum and frequencies.
```

Note:
-If you use python, you can use "scipy.io.wavfile.read" for reading WAV file, see https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.wavfile.read.html
-for reading mp3 file, see https://librosa.org/doc/main/generated/librosa.load.html

2. (20 points) Moving Average
Given that your student ID: 6xxxxxxx21, assign 7th and 8th to a_0 and a_1, respectively. If a_0 is zero, replace a_0 by the value of 5. Calculate $2N + 1$-day moving average of "GOOGL" stock price (close) using Fourier transform where $N = a_0 a_1$, e.g., Student ID is 6230186821, $N = 68$. Download "googl_close.csv" file for "GOOGL" stock price (close) using

```
!wget
https://drive.google.com/uc?id=1i5MiyFxVU4uiu6MJ_aSSeCtnEXkfv
0wc -O googl_close.csv
```

```
import pandas as pd
df = pd.read_csv("googl_close.csv", parse_dates=['Date'])
df.head()
```

	Date	Close
0	2004-08-19	2.511011
1	2004-08-20	2.710460
2	2004-08-23	2.737738
3	2004-08-24	2.624374
4	2004-08-25	2.652653

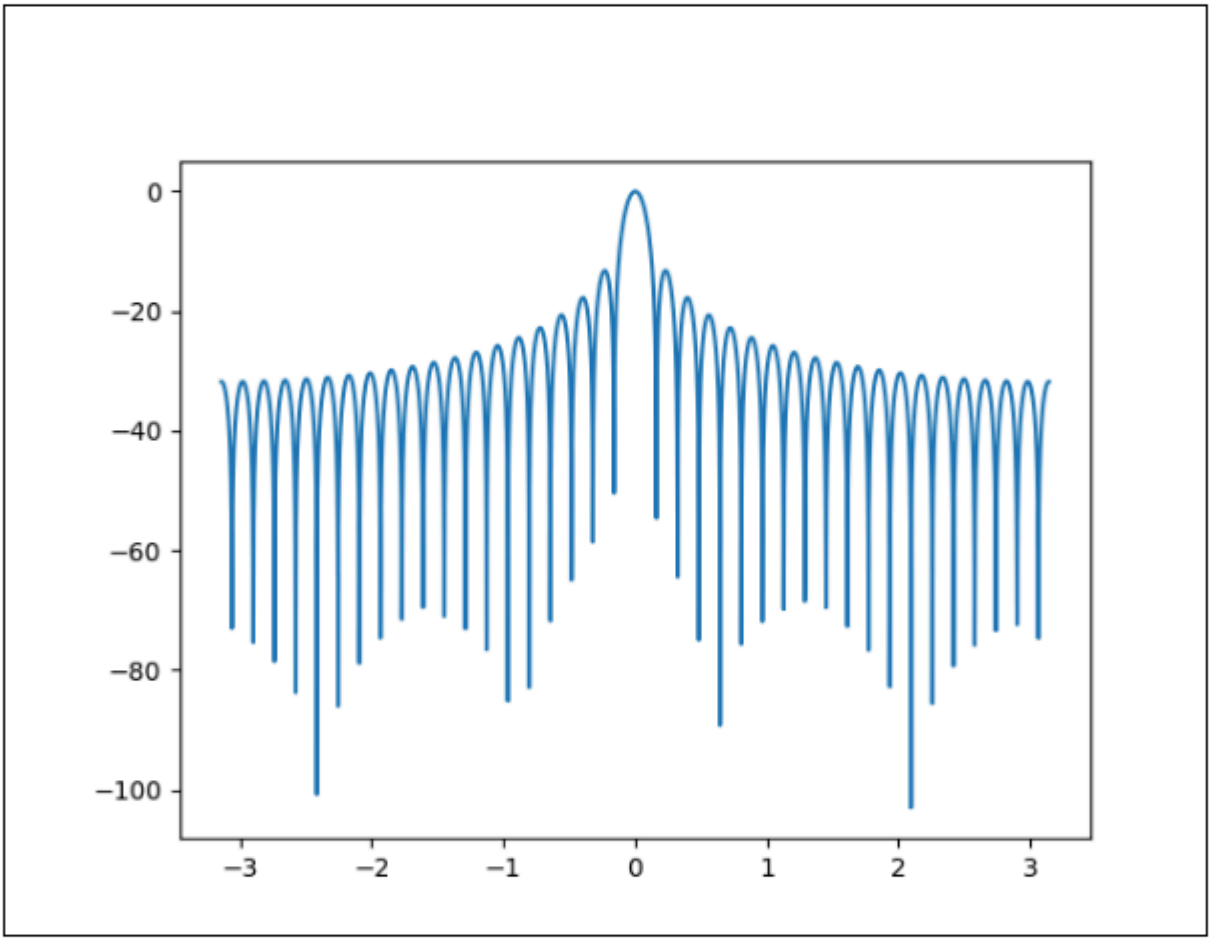
2.1) Student ID and N

Student ID =6330261921 $N = 19$

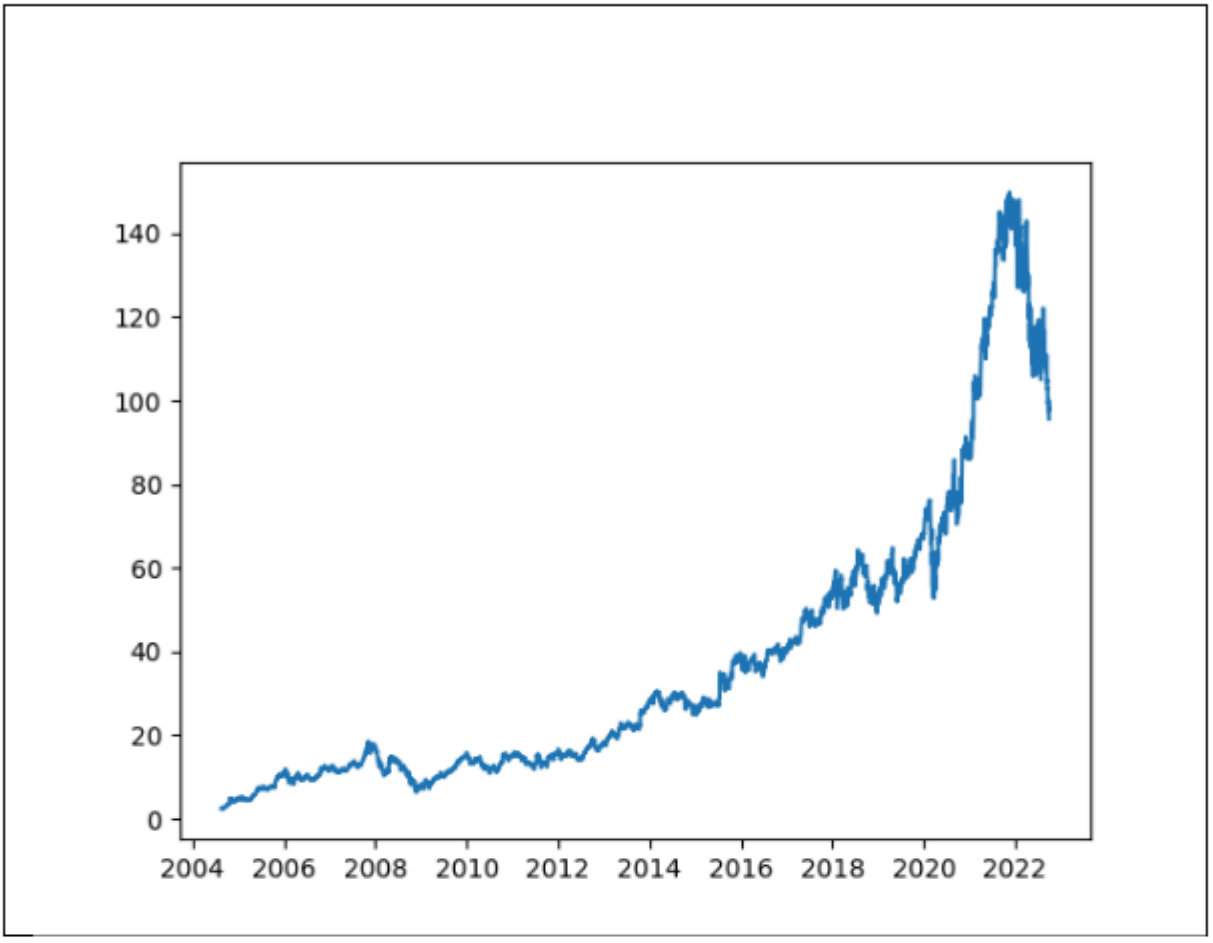
The output is the average values of the input over $2N + 1$-day

$$y[n] = \frac{1}{2N+1} \sum_{k=-N}^{N} x[n - k]$$

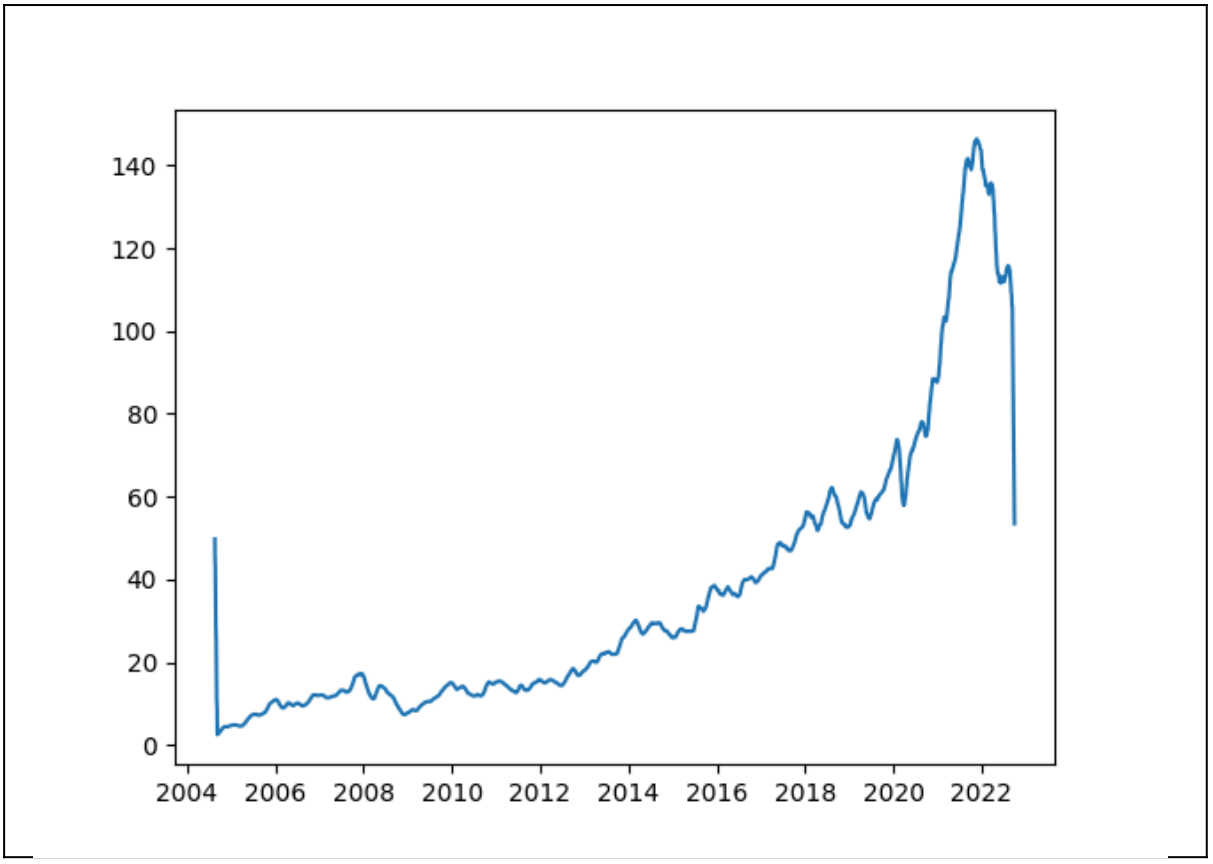
2.2) Plot log-magnitude plots for $2N + 1$-day moving average filter ($20log_{10}\left|H\left(e^{j\omega}\right)\right|$ vs. ω)



2.3) Plot the sequence of the original stock quotes (GOOGL – CLOSE)



2.4) Plot the output, $y[n]$



2.5) Explain how you calculate 2.1) and 2.4) in frequency domain or put your code here

```
start = -np.pi
stop = np.pi+0.001
h = []
while (start < stop):
    h.append(1/(N*2+1) * np.sin(start*(N*2+1)/2) /
np.sin(start/2))
    start += 0.001
H = np.array(h)
plt.plot(np.arange(-np.pi, np.pi+0.001, 0.001),
         20*np.log(np.abs(H))/np.log(10))
x = scipy.fft.fftshift(scipy.fft.fft(np.array(df['Close'])))
Y =
scipy.fft.ifft(H[((H.shape[0]-x.shape[0])//2):(-(H.shape[0]-x
.shape[0])//2)] * x)
plt.plot(df['Date'],np.abs(Y))
```