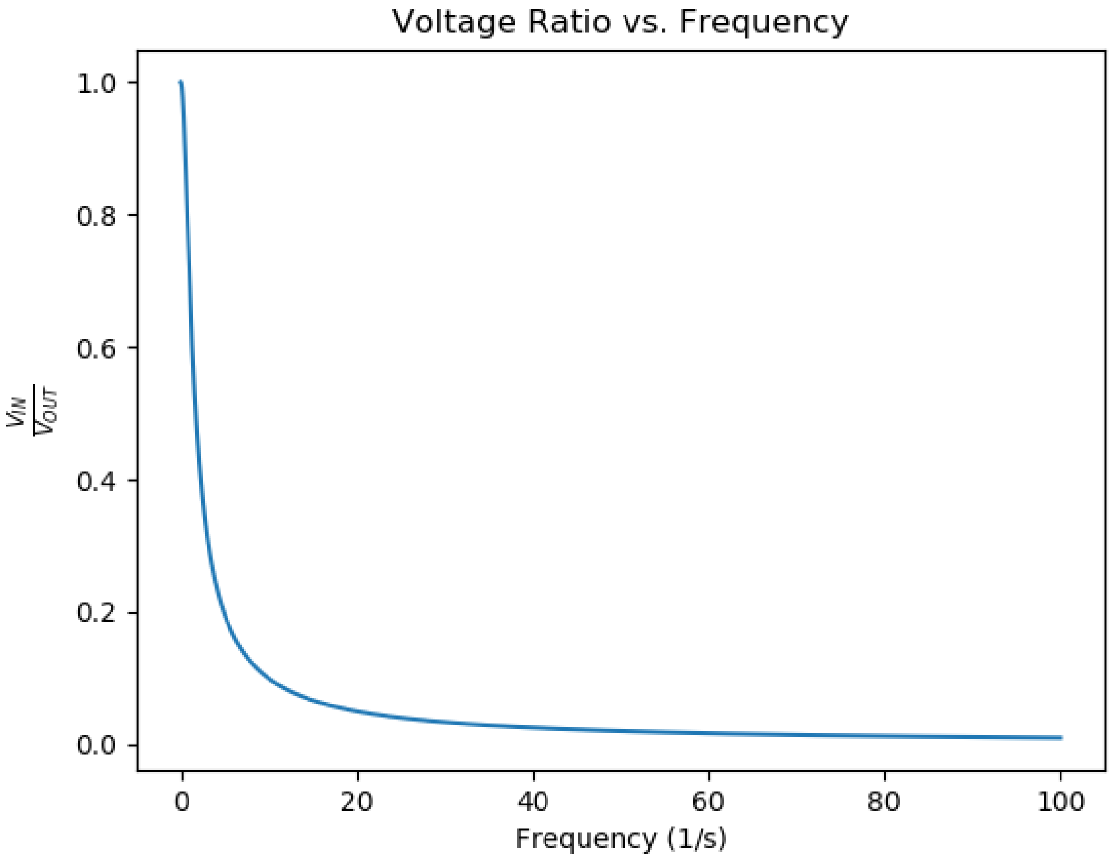
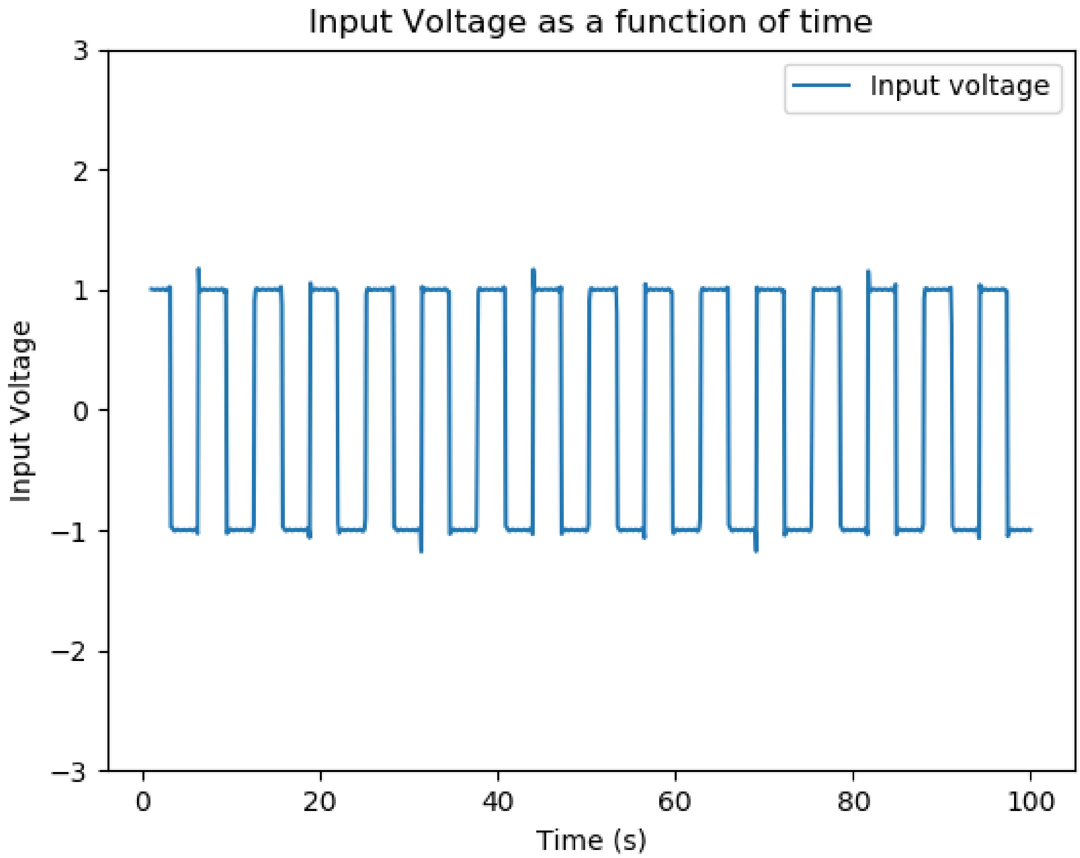

Voltage Ratio vs. Frequency
1.0
0.8
0.6
0.4
0.2
0.0
$\frac{V_{IN}}{V_{OUT}}$
0
20
40
60
80
100
Frequency (1/s)


Input Voltage as a function of time
Input voltage
Input Voltage
Time (s)
3
2
1
0
−1
−2
−3
0
20
40
60
80
100

```
import matplotlib.pyplot as plt
from math import *

R = 1.0
L = 1.0
V_out = R
V_ratio = []
frequency = []
w = 1.0
dw = 0.1
w_final = 100.0

t = 1.0
dt = 0.1
t_f = 100.0

#for plot of ratio vs. frequency
#this is a low frequency filter

'''
while w < w_final:
        V_in = sqrt(R**2+(w*L)**2)
        v_ratio = V_out/V_in
        V_ratio.append(v_ratio)
        frequency.append(w)
        w = w + dw
'''

k = 1.0
dk = 1.0
k_f = 100.0
V_In = []
V_Out = []
Time = []
shift = pi
#for new Vin that is a square wave

      t < t_f:
        k = 1.0
        V_in = 0.0
        V_out = 0.0

              k < k_f:
                V_in = V_in + ((4.0/pi)*(sin(((2.0*k)-1.0)*w*t))/((2.0*k)-1.0))
                #V_out = V_out + ((4.0/pi)*(sin((((2.0*k)-1.0)*w*t)+ shift))/
((2.0*k)-1.0))
                #I am not sure how the output voltage changes but I think that it is
jut going to be more like a sine wave (edges will be smoother)
                k = k + dk


        Time.append(t)
        V_In.append(V_in)
        V_Out.append(V_out)
        t = t + dt



#plt.plot(frequency,V_ratio)
#plt.ylabel(r"$\frac{V_{IN}}{V_{OUT}}$", fontsize =12)
#plt.xlabel('Frequency (1/s)')
#plt.title('Voltage Ratio vs. Frequency')
```

```
plt.plot(Time, V_In, label='Input voltage')
#plt.plot(Time, V_Out, label='Output Voltage')
plt.xlabel('Time (s)')
plt.ylim(-3,3)
plt.ylabel('Input Voltage')
plt.title('Input Voltage as a function of time')
plt.legend()
plt.show()
```