```
###########################################################################
###Code for figure 3.1
###plots the Wiener process and its corresponding step function
###WienerStepProcessGraphic.py
###Python 2.7
###########################################################################
import numpy as np
import matplotlib.pylab as plt
from NumericalSDE import *

#Number of steps.
n = 16**2
n_step = n/(2*16)

#Create an empty array to store the realizations.
w = wiener(n)
t = timegrid(n)

#Step function
w_step = np.zeros(n_step+1)
t_step = timegrid(n_step)


for k in range(0,n_step):
    w_step[k] = w[k*(n/n_step)]
w_step[n_step] = w[n-1]

#plot
plt.plot(t, w,'k', linewidth=0.6)
plt.step(t_step, w_step,'k', linewidth=1,color='b', where='post')

plt.xlabel('t', fontsize=16)
plt.ylabel('x', fontsize=16)
plt.show()


```