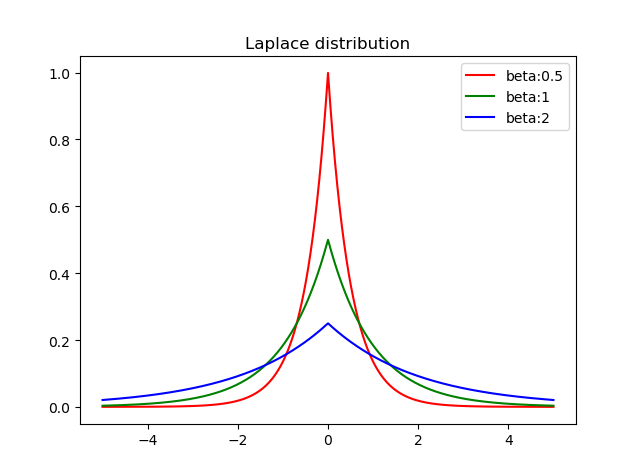
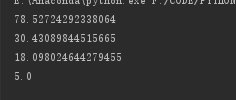
拉普拉斯分布函数

def laplace\_function(x, beta):  
 result = (1 / (2 \* beta)) \* np.e \*\* (-1 \* (np.abs(x) / beta))  
 return result  
  
  
# 在-5到5之间等间隔的取10000个数  
x = np.linspace(-5, 5, 10000)  
y1 = [laplace\_function(x\_, 0.5) for x\_ in x]  
y2 = [laplace\_function(x\_, 1) for x\_ in x]  
y3 = [laplace\_function(x\_, 2) for x\_ in x]  
  
plt.plot(x, y1, color='r', label='beta:0.5')  
plt.plot(x, y2, color='g', label='beta:1')  
plt.plot(x, y3, color='b', label='beta:2')  
plt.title("Laplace distribution")  
plt.legend()  
plt.show()





x = [50,40,20,5]

对以上列表添加噪音结果

对大型数据加噪音

import pandas as pd  
import numpy as np  
def loadDataSet(fileName):  
 numFeat = len(open(fileName).readline().split(',')) # 计算有多少列  
 dataMat = []  
 fr = open(fileName)  
 for line in fr.readlines()[0:-1]: # 遍历原始数据集每一行  
 lineArr =[]  
 curLine = line.strip().split(',') # 是一列表类型  
 #print(curLine)  
 for i in range(numFeat): # numFeat - 1的原因：因为原始数据的最后一列是类别，不是属性数据  
 lineArr.append((curLine[i])) # 一个一个传进lineArr列表向量  
 dataMat.append(lineArr) # 再传进dataMat列表向量  
 return dataMat  
  
a=loadDataSet("adult.data")  
b=loadDataSet("adult.test")  
EXdata=pd.DataFrame(data=a)  
EXtest=pd.DataFrame(data=b)  
Edata=np.array(EXdata)  
Etext=np.array(EXtest)  
  
def noisyCount(sensitivety, epsilon):  
 beta = sensitivety / epsilon  
 u1 = np.random.random()  
 u2 = np.random.random()  
 if u1 <= 0.5:  
 n\_value = -beta \* np.log(1. - u2)  
 else:  
 n\_value = beta \* np.log(u2)  
 return n\_value  
  
  
def laplace\_mech(data, sensitivety, epsilon):  
 for i in range(len(data)):  
 data[i] += noisyCount(sensitivety, epsilon)  
 return data  
  
def maxarray(array):  
 maxnumm = 0  
 for i in range(len(array)):  
 if array[i] > maxnumm:  
 maxnumm = array[i]  
 return maxnumm  
  
def minarray(array):  
 maxnumm = 9999999  
 for i in range(len(array)):  
 if array[i] < maxnumm:  
 maxnumm = array[i]  
 return maxnumm  
if \_\_name\_\_ == '\_\_main\_\_':  
 x1 = Edata[:, 0]  
 x = []  
 for i in x1:  
 x.append(int(i))  
 sensitivety = 0.01\*(maxarray(x)-minarray(x))  
 epsilon = 1  
 data = laplace\_mech(x, sensitivety, epsilon)  
 fo = open("lablace.txt", "w")  
 for i in range(len(data)):  
 print(Edata[:, 0][i])  
 print(data[i])  
 fo.write(str(Edata[:, 0][i]))  
 fo.write(str('\n'))  
 fo.write(str(data[i]))  
 fo.write(str('\n'))  
 fo.close()

实验结果保存在 Laplace.txt里面

接下里验证添加噪音对实验结果的影响：

print(maxarray(x)-maxarray(data)+minarray(data)-minarray(x))



对敏感度没有影响

print(sum(x2)/len(x2))  
print(sum(data)/len(data))

对也就是数学期望也没什么影响



