# Лабораторная работа № 3

# Супонина Анастасия Павловна

## Цель работы.

Научиться применять распределения для преобразования случайных величин.

## Задание.

Сгенерировать выборку точек, равномерно распределенных внутри круга двумя способами:

1. Равномерно распределить точки внутри квадрата, внутрь которого вписана окружность, и отфильтровать точки, лежащие за пределами окружности.
2. Генерировать точки путем задания случайного угла и расстояния от центра окружности.

Радиус окружности R=10, размер выборки 1000 точек. Для созданных выборок сделать следующее:

a. Создать рисунок, иллюстрирующий расположение точек сгенерированной выборки внутри окружности;  
b. Найти выборочные средние координат точек и их дисперсию;  
c. Построить график плотности распределения расстояния от случайной равномернораспределенной точки в круге до фиксированной точки лежащей вне окружности(к примеру, с координатами X=20, Y=0).  
d. Построить график плотности распределения расстояния между двумя cлучайными точками, равномерно расположенными внутри круга

1. Равномерно распределить точки внутри квадрата, внутрь которого вписана окружность, и отфильтровать точки, лежащие за пределами окружности.

# Ввожу все необходимые для работы библиотеки  
import numpy as np  
import matplotlib.pyplot as plt  
from scipy.stats import gaussian\_kde

Для начала работы задаю окружность с радиусом R=10 и размер выборки.

circle\_radius = 10  
set\_size = 1000

Для того, чтобы нарисовать точки нам необходимо знать их значения по координатам x, y. Создаю два массива для записи x, y, а также счетчик для сгенерированных в цикле точек.

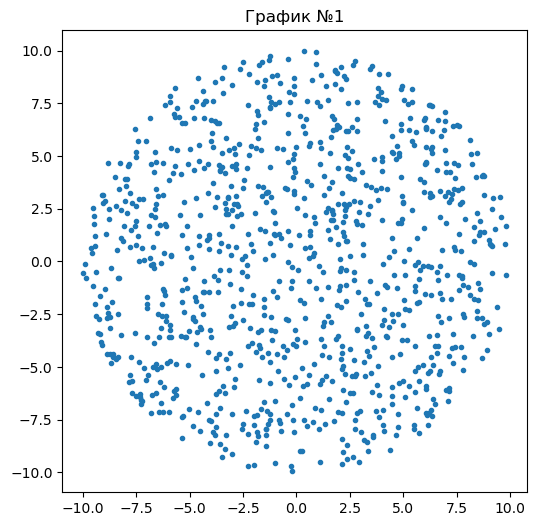
set\_x = []  
set\_y = []  
dots\_counter = 0

Пишу цикл для создания точек и записи их значений

while dots\_counter < set\_size:  
 x = np.random.uniform(-10,10)  
 y = np.random.uniform(-10,10)  
 # сейчас точки генерируются по квадрату, зададим дополнительное условие, чтобы они генерировались в окружности.  
 if x\*\*2 + y\*\*2 <= circle\_radius\*\*2:  
 set\_x.append(x)  
 set\_y.append(y)  
 dots\_counter += 1  
 # таким образом из случайно сгенерированных точек буду записывать только те, которые входят в окружность

1. Создать рисунок, иллюстрирующий расположение точек сгенерированной выборки внутри окружности;

plt.figure(dpi=100,figsize=(6,6))  
plt.title('График №1')  
plt.plot(set\_x, set\_y, '.')  
plt.show()



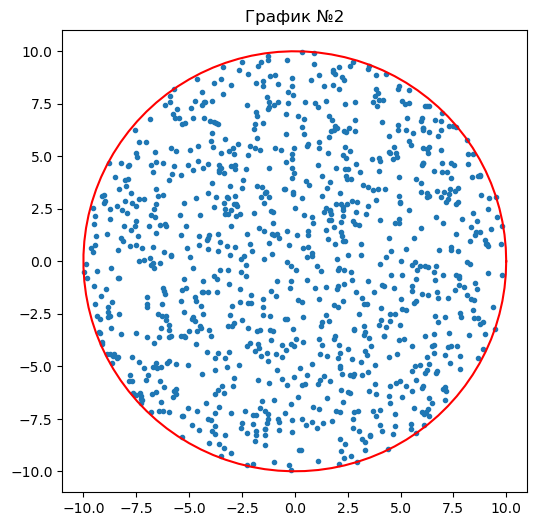
png

Теперь напишу отдельную функцию, которая будет рисовать на графике линию окружности, используя формулу для построения окружности через угол.

circle\_angels = np.linspace(0, 2\*np.pi, 100)  
circle\_x = circle\_radius\*np.cos(circle\_angels)  
circle\_y = circle\_radius\*np.sin(circle\_angels)

Добавляю отображение окружности к первому графику.

plt.figure(dpi=100,figsize=(6,6))  
plt.title('График №2')  
plt.plot(set\_x, set\_y, '.')  
plt.plot(circle\_x, circle\_y, color = 'red')  
plt.show()

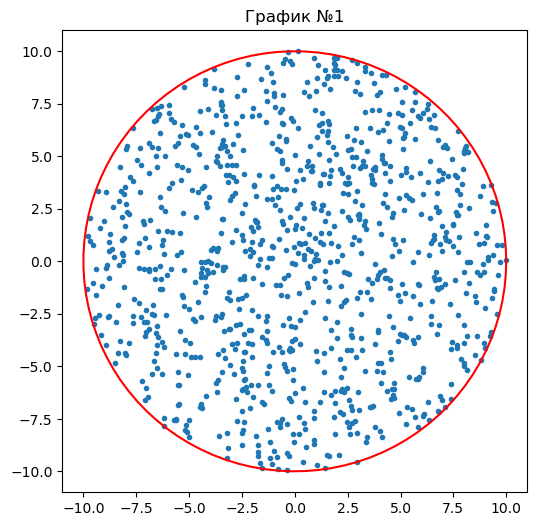


png

На графике видно, что при таком решении часть точек попадает на линию окружности, для того, чтобы избежать этого изменяю условия для окружности в цикле на < вместо <=.

while dots\_counter < set\_size:  
 x = np.random.uniform(-10,10)  
 y = np.random.uniform(-10,10)  
 # сейчас точки генерируются по квадрату, зададим дополнительное условие, чтобы они генерировались в окружности.  
 if x\*\*2 + y\*\*2 < circle\_radius\*\*2:  
 set\_x.append(x)  
 set\_y.append(y)  
 dots\_counter += 1  
 # таким образом из случайно сгенерированных точек буду записывать только те, которые входят в окружность

plt.figure(dpi=100,figsize=(6,6))  
plt.title('График №1')  
plt.plot(set\_x, set\_y, '.')  
plt.plot(circle\_x, circle\_y, color = 'red')  
plt.show()



png

Вычисляю окружность при помощи полярных координат.

1. Найти выборочные средние координат точек и их дисперсию;

print('Mean for x:', np.mean(set\_x))  
print('Var for x:', np.var(set\_x))  
print('Mean for y:', np.mean(set\_y))  
print('Var for y:', np.var(set\_y))

Mean for x: 0.1592505379020999  
Var for x: 23.896467269836094  
Mean for y: 0.2281278788170736  
Var for y: 24.40504820095258

1. Построить график плотности распределения расстояния от случайной равномернораспределенной точки в круге до фиксированной точки лежащей вне окружности(к примеру, с координатами X=20, Y=0). Задаю главную точку:

main\_x = 20  
main\_y = 0

Пишу функцию для расчета расстояния между двумя точками

def distance (x1, y1, x2, y2):  
 return np.sqrt((x2-x1)\*\*2 + (y2-y1)\*\*2)

Создаю пустой массив для помещения в него вычисленных расстояний и добавляем в массив все полученные расстояния

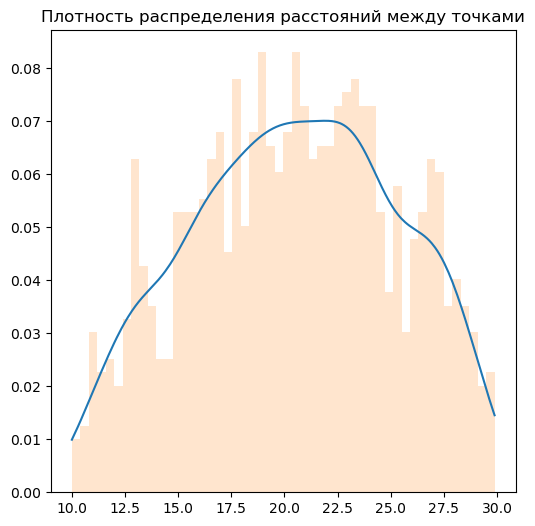
distances = []  
for i in range(set\_size):  
 distances.append(distance(set\_x[i], set\_y[i], main\_x, main\_y))  
distances.sort() # сортирую значения по возрастанию

Чтобы получить плотность использую gaussian\_kde

kde\_distances\_y = gaussian\_kde(distances).evaluate(distances) # получаем значения y, которые позволяют сгладить нашу плотность

Рисую гистограмму, по этой гистограмме сглаживаю линию плотности.

plt.figure(dpi=100,figsize=(6,6))  
plt.title('Плотность распределения расстояний между точками')  
plt.plot(distances, kde\_distances\_y)  
plt.hist(distances, bins = 50, alpha = 0.2, density = True)  
plt.show()



png

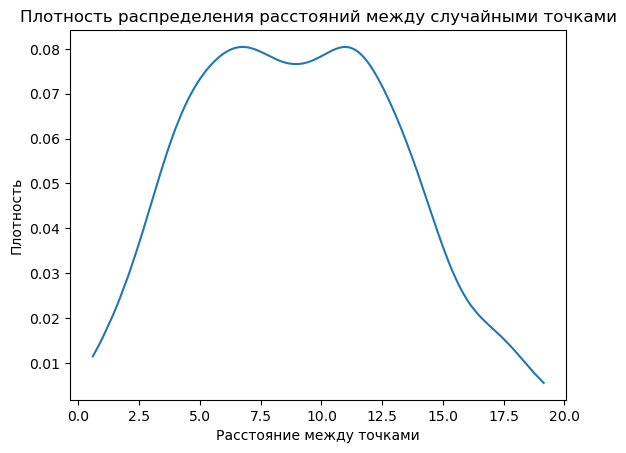
1. Построить график плотности распределения расстояния между двумя cлучайными точками, равномерно расположенными внутри круга

Для того чтобы построить расстояние между двумя точками в круге, беру случайным образом из set\_x, set\_y две случайные точки и для этой иттерации остальные значения из массива удаляю, провожу это то колво раз, которое запрашивается в задании, получаю эти значения и строю график плотности.

point\_distance = []  
for i in range(500):  
 point1, point2 = np.random.choice(set\_size, 2, replace = False)  
 point\_distance.append(distance(set\_x[point1], set\_y[point1], set\_x[point2], set\_y[point2]))

kde\_between\_points = gaussian\_kde(point\_distance)  
point\_distance.sort()

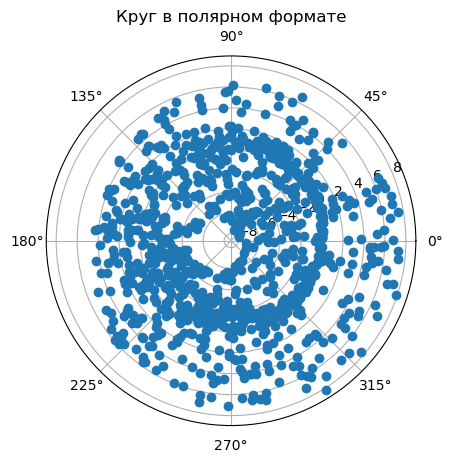
plt.figure(dpi=100)  
plt.title('Плотность распределения расстояний между случайными точками')  
plt.plot(point\_distance, kde\_between\_points(point\_distance))  
plt.xlabel("Расстояние между точками")  
plt.ylabel("Плотность")  
plt.show()



png

1. Генерировать точки путем задания случайного угла и расстояния от центра окружности.
2. Создать рисунок, иллюстрирующий расположение точек сгенерированной выборки внутри окружности;

radian = []  
count = 0   
while count < set\_size:  
 rad = np.random.uniform()  
 radian.append(rad)  
 count += 1  
  
  
set\_circle\_x = []  
set\_circle\_y = []  
for i in radian:  
 r = np.random.uniform(-10,10)  
 circle\_x = r\*np.cos(i)  
 circle\_y = r\*np.sin(i)  
 set\_circle\_x.append(circle\_x)  
 set\_circle\_y.append(circle\_y)  
 dots\_counter += 1  
  
plt.axes(projection ='polar')  
  
# Задайте заголовок для полярного графика  
plt.title('Круг в полярном формате')  
  
# Постройте окружность с радиусом 2 в полярной системе координат  
rads = np.arange(0, (2\*np.pi), 0.01)  
  
plt.polar(set\_circle\_x, set\_circle\_y, 'o')  
  
# Отобразить полярный график  
plt.show()



png

1. Найти выборочные средние координат точек и их дисперсию;

print('Mean for x:', np.mean(set\_circle\_x))  
print('Var for x:', np.var(set\_circle\_x))  
print('Mean for y:', np.mean(set\_circle\_y))  
print('Var for y:', np.var(set\_circle\_y))

Mean for x: 0.014004656878831014  
Var for x: 25.15625981510718  
Mean for y: 0.13435230442640883  
Var for y: 9.736578501877752

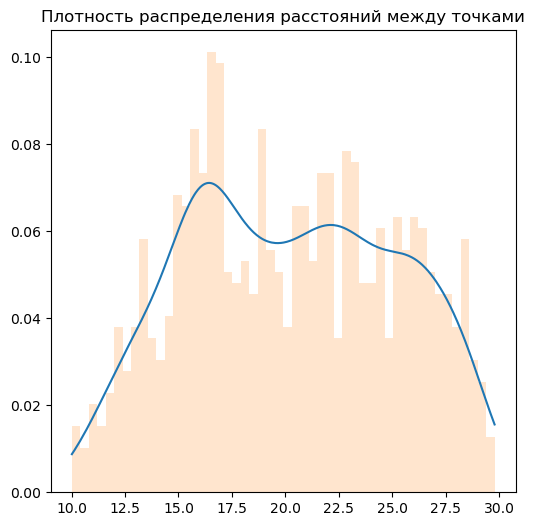
1. Построить график плотности распределения расстояния от случайной равномернораспределенной точки в круге до фиксированной точки лежащей вне окружности(к примеру, с координатами X=20, Y=0).

main\_x = 20  
main\_y = 0

distances = []  
for i in range(set\_size):  
 distances.append(distance(set\_circle\_x[i], set\_circle\_y[i], main\_x, main\_y))  
distances.sort() # сортирую значения по возрастанию

kde\_distances\_y = gaussian\_kde(distances).evaluate(distances) # получаем значения y, которые позволяют сгладить нашу плотность

plt.figure(dpi=100,figsize=(6,6))  
plt.title('Плотность распределения расстояний между точками')  
plt.plot(distances, kde\_distances\_y)  
plt.hist(distances, bins = 50, alpha = 0.2, density = True)  
plt.show()



png

1. Построить график плотности распределения расстояния между двумя cлучайными точками, равномерно расположенными внутри круга

point\_distance = []  
for i in range(500):  
 point1, point2 = np.random.choice(set\_size, 2, replace = False)  
 point\_distance.append(distance(set\_circle\_x[point1], set\_circle\_y[point1], set\_circle\_x[point2], set\_circle\_y[point2]))

kde\_between\_points = gaussian\_kde(point\_distance)  
point\_distance.sort()

plt.figure(dpi=100)  
plt.title('Плотность распределения расстояний между случайными точками')  
plt.plot(point\_distance, kde\_between\_points(point\_distance))  
plt.xlabel("Расстояние между точками")  
plt.ylabel("Плотность")  
plt.show()



png