import numpy as np

import matplotlib.pyplot as plt

from scipy import stats

# Генераторы псевдослучайных чисел (ПСЧ) для выборок

# M, N - размер выборки (строки, столбцы)

# Равномерное распределение от 0 до 1

# МО = 0.5, СКО = 0.5/√3

x = np.random.rand(M, N)

# Равномерное распределение от A до B

# МО = (A+B)/2, СКО = (B-A)/√12

x = np.random.uniform(A, B, (M, N))

# Нормальное (Гауссово) распределение

# Параметры: m (среднее), s (СКО)

# МО = m, СКО = s

x = np.random.normal(m, s, (M, N))

# Экспоненциальное распределение

# Параметр: m (среднее)

# МО = m, СКО = m

x = np.random.exponential(m, (M, N))

# Биномиальное распределение

# Параметры: n (число испытаний), p (вероятность успеха)

# МО = n\*p, СКО = sqrt(n\*p\*(1-p))

x = np.random.binomial(n, p, (M, N))

# Пуассоновское распределение

# Параметр: p (среднее)

# МО = p, СКО = √p

x = np.random.poisson(p, (M, N))

# Равномерное целочисленное распределение

# Параметры: imin, imax

x = np.random.randint(imin, imax+1, (M, N))

# Выборочные числовые характеристики (точечные оценки)

# Меры смещения (положения)

Mx = np.mean(x)  # Среднее

Mex = np.median(x)  # Медиана

Mox = stats.mode(x, axis=None)[0][0]  # Мода (берём первое значение, если несколько)

# Меры вариабельности (рассеяния)

Dx = np.var(x, ddof=1)  # Дисперсия (несмещённая оценка)

Sx = np.std(x, ddof=1)  # СКО

Wx = np.max(x) - np.min(x)  # Размах

Wdx = np.percentile(x, 75) - np.percentile(x, 25)  # Межквартальный размах

# Меры формы

Asx = stats.skew(x)  # Асимметрия

Ex = stats.kurtosis(x)  # Эксцесс

# Мера корреляции

R = np.corrcoef(x, y)[0, 1]  # Коэффициент корреляции между x и y

# Визуализация

# Гистограмма

plt.hist(x, bins=100)

plt.title('Гистограмма')

plt.xlabel('Значение')

plt.ylabel('Частота')

plt.show()

# Ящик с усами (Boxplot)

plt.boxplot(x)

plt.title('Ящик с усами')

plt.ylabel('Значение')

plt.show()

# Эмпирическая функция распределения (ECDF)

sorted\_x = np.sort(x)

y = np.arange(1, len(x) + 1) / len(x)

plt.step(sorted\_x, y)

plt.title('Эмпирическая функция распределения')

plt.xlabel('Значение')

plt.ylabel('Доля')

plt.grid(True)

plt.show()