МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Российский государственный университет им. А.Н. Косыгина   
(Технологии. Дизайн. Искусство)»

Отчет по лабораторной работе № 4

по дисциплине «Теория вероятностей и математическая статистика»

Тема: «Первичная обработка простой случайной выборки»

Выполнил: Ольховский Н.С., ИТИВ-223

Проверил: Вахромеева Е. Н.

Москва 2024

Вариант 9

**Задача**

1. Преобразовать выборку в вариационный ряд;

2. Вычислить точечные оценки МО, дисперсии, СКО, коэффициентов

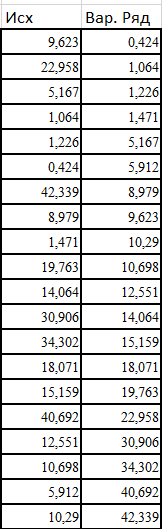
вариации, асимметрии и эксцесса;

3. Вычислить точечные оценки медианы и размаха;

4. Построить точечную оценку функции распределения;

5. Построить интервальные оценки для МО СВ при неизвестной дисперсии и для дисперсии с надежностью 0.9.

**Результат, рассчитанный в Excel**

****

**Результат, рассчитанный в Python**

Математическое ожидание: 15.366100

Дисперсия: 166.578080

Среднеквадратическое отклонение: 12.906930

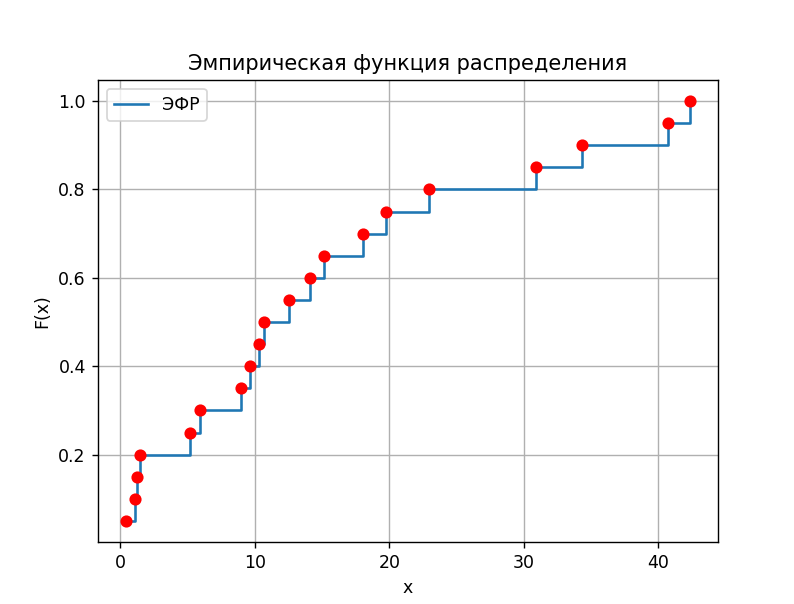
Коэффициент вариации: 84.01%

Асимметрия: 0.808118

Эксцесс: -0.592712

Медиана: 11.624500

Размах: 41.915000

****

Доверительный интервал для МО с уровнем доверия 0.9:

Нижняя граница: 10.266443597255357

Верхняя граница: 20.299456402744642

**Тексты программы**

import numpy as np

from scipy.stats import skew, kurtosis, t

import matplotlib.pyplot as plt

# Исходные данные выборки

data = [0.424, 1.064, 1.226, 1.471, 5.167, 5.912, 8.979, 9.623, 10.29, 10.698, 12.551, 14.064, 15.159, 18.071, 19.763, 22.958, 30.906, 34.302, 40.692, 42.339]

### 1. Преобразовать выборку в вариационный ряд

data = np.sort(data)

### 2. Вычислить точечные оценки МО, дисперсии, СКО, коэффициентов вариации, асимметрии и эксцесса

# Число элементов в выборке

n = len(data)

# Математическое ожидание

mean = np.mean(data)

# Дисперсия

variance = np.var(data, ddof=1)

# Среднеквадратическое отклонение

std\_dev = np.sqrt(variance)

# Коэффициент вариации

coef\_variation = (std\_dev / mean) \* 100

# Асимметрия

skewness = skew(data, bias=False)

# Эксцесс

kurt = kurtosis(data, bias=False)

print(f"Математическое ожидание: {mean:.6f}")

print(f"Дисперсия: {variance:.6f}")

print(f"Среднеквадратическое отклонение: {std\_dev:.6f}")

print(f"Коэффициент вариации: {coef\_variation:.2f}%")

print(f"Асимметрия: {skewness:.6f}")

print(f"Эксцесс: {kurt:.6f}")

### 3. Вычислить точечные оценки медианы и размаха

# Медиана

median = np.median(data)

# Размах

range\_value = np.max(data) - np.min(data)

print(f"Медиана: {median:.6f}")

print(f"Размах: {range\_value:.6f}")

### 4. Построить точечную оценку функции распределения

# Количество элементов в выборке

n = len(data)

# Вычисление Эмпирической функции распределения

F\_n = np.arange(1, n + 1) / n

# Построение графика

plt.step(data, F\_n, where='post', label='ЭФР')

plt.scatter(data, F\_n, color='red', zorder=5)

# Оформление графика

plt.xlabel('x')

plt.ylabel('F(x)')

plt.title('Эмпирическая функция распределения')

plt.grid(True)

plt.legend()

plt.show()

### 5. Построить интервальные оценки для МО СВ при неизвестной дисперсии и для дисперсии с надежностью 0.9

n = len(data) # Размер выборки

mean = np.mean(data) # Выборочное среднее

std = np.std(data, ddof=1) # Выборочное стандартное отклонение

confidence\_level = 0.9 # Уровень доверия

t\_value = t.ppf((1 + confidence\_level) / 2, df=n - 1) # Критическое значение t-распределения

SE = std\_dev / np.sqrt(n) # Размер стандартной ошибки

# Вычисляем доверительный интервал

margin\_of\_error = t\_value \* SE

lower\_bound = mean - margin\_of\_error

upper\_bound = mean + margin\_of\_error

print("Доверительный интервал для математического ожидания с уровнем доверия {}:".format(confidence\_level))

print("Нижняя граница:", lower\_bound)

print("Верхняя граница:", upper\_bound)