

Отчет о выполнении практической работы №5

Федоров Егор, Р3215, вариант 19

Необходимо определить следующие статистические характеристики:

- вариационный ряд
- экстремальные значения и размах
- оценки математического ожидания и среднеквадратического отклонения
- эмпирическую функцию распределения и её график
- гистограмму и полигон приведенных частот группированной выборки.

Для расчета характеристик и построения графиков нужно написать программу на одном из языков программирования. Листинг программы и результаты работы должны быть представлены в отчете по практической работе.

При выполнении практической работы были получены следующие результаты:

- Вариационный ряд:

$$\begin{aligned} -1.55 \leq -1.45 \leq -1.45 \leq -1.38 \leq -1.31 \leq -1.14 \leq -1.0 \leq -0.9 \leq 0.17 \leq 0.24 \leq \\ \leq 0.34 \leq 0.38 \leq 0.52 \leq 0.52 \leq 0.55 \leq 0.62 \leq 0.73 \leq 0.8 \leq 0.9 \leq 1.31 \end{aligned}$$

- Экстремальные значения:

$$x_{(20)} = 1.31 \quad x_{(1)} = -1.55 \quad R = x_{(n)} - x_{(1)} = 2.86$$

- Оценки математического ожидания и среднеквадратичного отклонения:

$$MX \approx \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i = -0.1550$$

$$DX \approx \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = 0.9065$$

$$\sigma = \sqrt{DX} \approx 0.9521$$

$$S^2 \approx 0.9542$$

$$S \approx 0.9768$$

- Эмпирическая функция распределения описана в таблице ???. График эмпирической функции распределения

x	n_i	$\frac{n_i}{n}$
-1.55	1	0.05
-1.45	2	0.1
-1.38	1	0.05
-1.31	1	0.05
-1.14	1	0.05
-1.0	1	0.05
-0.9	1	0.05
0.17	1	0.05
0.24	1	0.05
0.34	1	0.05
0.38	1	0.05
0.52	2	0.1
0.55	1	0.05
0.62	1	0.05
0.73	1	0.05
0.8	1	0.05
0.9	1	0.05
1.31	1	0.05

Таблица 1: Статистический ряд распределения

представлен на рисунке 1. Эмпирическая функция в аналитическом виде:

$$F^*(x) = \begin{cases} 0, & x \leq -1.55 \\ 0.05, & -1.55 < x \leq -1.45 \\ 0.15, & -1.45 < x \leq -1.38 \\ 0.2, & -1.38 < x \leq -1.31 \\ 0.25, & -1.31 < x \leq -1.14 \\ 0.3, & -1.14 < x \leq -1.0 \\ 0.35, & -1.0 < x \leq -0.9 \\ 0.4, & -0.9 < x \leq 0.17 \\ 0.45, & 0.17 < x \leq 0.24 \\ 0.5, & 0.24 < x \leq 0.34 \\ 0.55, & 0.34 < x \leq 0.38 \\ 0.6, & 0.38 < x \leq 0.52 \\ 0.7, & 0.52 < x \leq 0.55 \\ 0.75, & 0.55 < x \leq 0.62 \\ 0.8, & 0.62 < x \leq 0.73 \\ 0.85, & 0.73 < x \leq 0.8 \\ 0.9, & 0.8 < x \leq 0.9 \\ 0.95, & 0.9 < x \leq 1.31 \\ 1 & 1.31 < x \end{cases}$$

- Гистограмма и полигон приведенных частот группированной выборки приведены на рисунках 2 и 3 соответственно.

Листинг программы представлен в конце отчета.

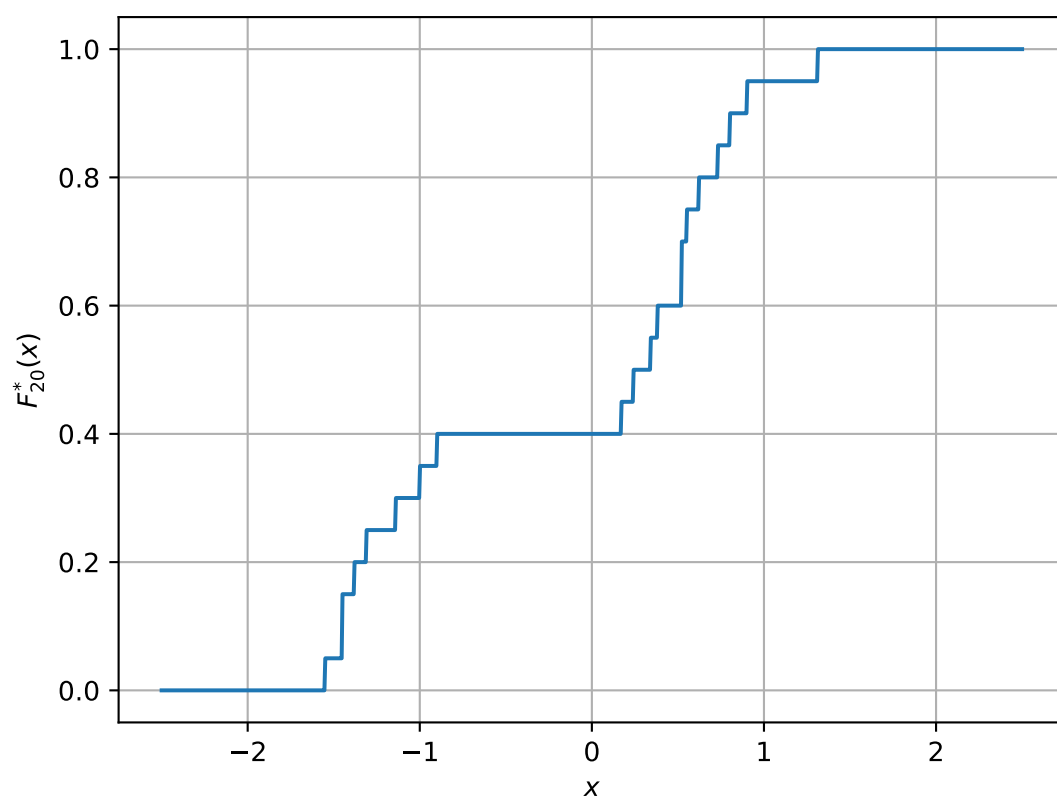


Рис. 1: График эмпирической функции распределения

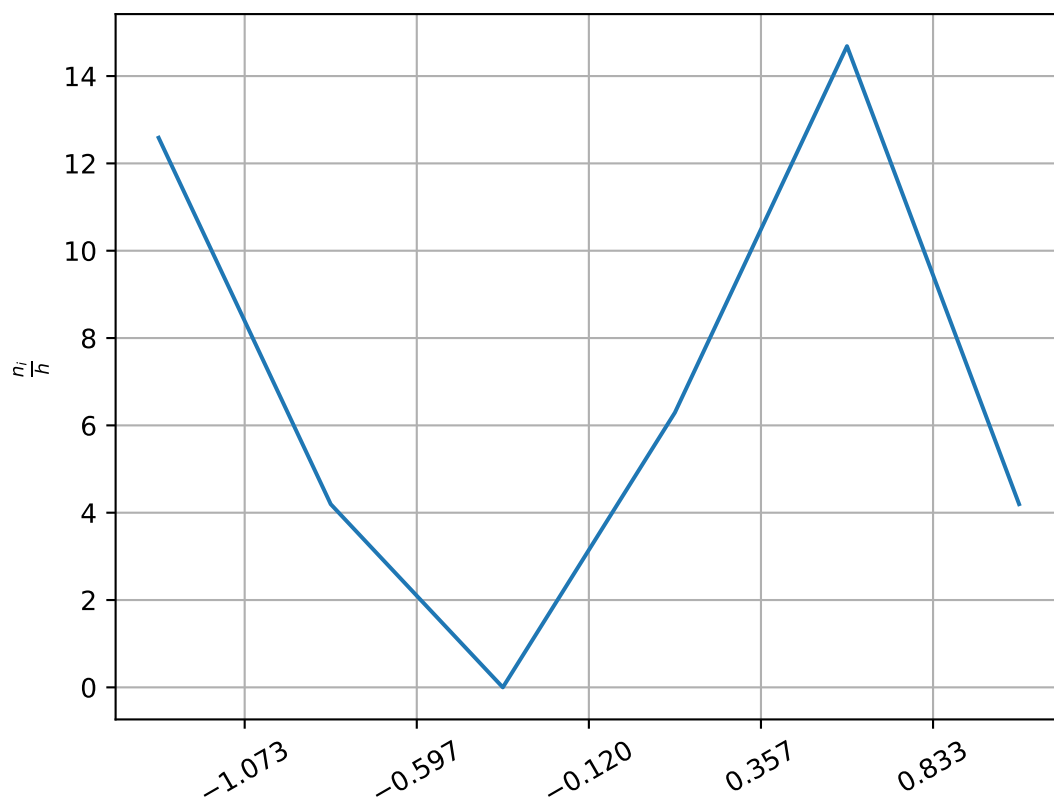


Рис. 2: Полигон приведенных частот группированной выборки

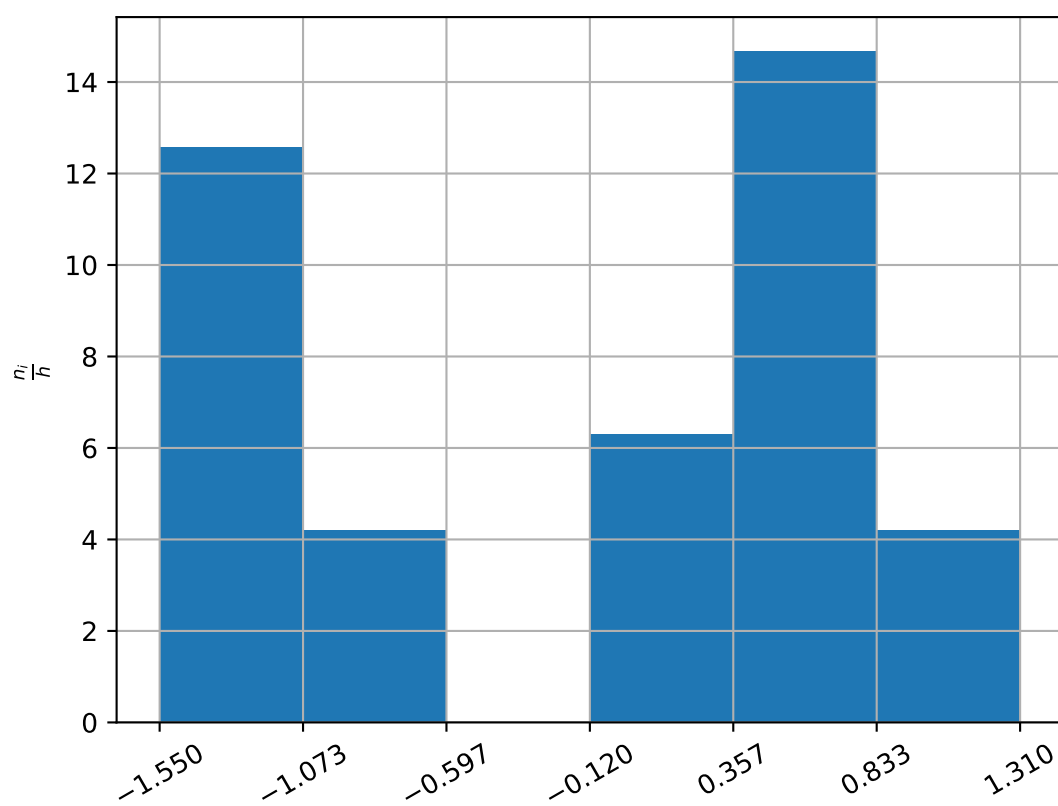


Рис. 3: Гистограмма приведенных частот группированной выборки

```

from collections import Counter
import matplotlib.pyplot as plt
from math import log, ceil
import numpy as np

# Var 19
data = [
    0.34, -1.38,
    -1.14, 0.8,
    0.73, 0.38,
    1.31, 0.52,
    -1.55, -0.90,

    0.90, -1.00,
    0.24, 0.62,
    0.55, -1.45,
    -1.45, 0.52,
    0.17, -1.31,
]
assert(len(data) == 20)

def print_variation_series(data):
    print("### Вариационный ряд")
    s = sorted(data)
    for i in s[:-1]:
        print(i, end = " \leq ")
    print(s[-1])

def print_stat_series(data):
    print("### Статистический ряд")
    cnt = Counter(data)
    keys = sorted(list(set(data)))
    for k in keys:
        print(f"{k} & {cnt[k]} & {cnt[k]/len(data)} \\\\")

def print_extreme_values(data):
    print("### Экстремальные значения")
    s = sorted(data)
    print("x_(1) = ", s[0])
    print("x_(n) = ", s[-1])

def print_sampling_range(data):
    s = sorted(data)
    print(f"### Размах выборки: {s[-1] - s[0]:.4f}")

def empirical_distribution_function(data):
    return lambda x: len([y for y in data if y < x])/len(data)

def print_emp_func(data):
    f = empirical_distribution_function(data)
    for x in sorted(list(set(data + [1.32]))):
        print(f"{x} & {f(x)} \\\\")

def print_emp_func_cases(data):
    s = sorted(list(set(data)))
    f = empirical_distribution_function(data)
    for i in range(1, len(s)):

```

```

        print(f"{f(s[i])}, & {s[i-1]} < x \leq {s[i]} \\\\")

def print_estimate_expected_value(data):
    avg = sum(data) / len(data)
    print(f"Оценка среднего значения: {avg:.4f}")

def compute_dispersia(data):
    n = len(data)
    average = sum(data) / n
    return 1 / n * sum([ (xi - average)**2 for xi in data ])

def print_dispersia(data):
    disp = compute_dispersia(data)
    print(f"Дисперсия выборки: {disp:.4f}")

def print_standard_deviation(data):
    standard_deviation = compute_dispersia(data)**0.5
    print(f"Стандартное отклонение {standard_deviation:.4f}")

def print_most_common_value(data):
    most_common = max(set(data), key=data.count)
    print(f"Мода выборки:", most_common)

def print_median(data):
    s = sorted(data)
    k = int(len(data) // 2)
    median = (s[k-1] + s[k]) / 2
    print(f"Медиана: {median:.4f}")

def draw_emp_dist_function(data):
    f = empirical_distribution_function(data)
    x_space = np.linspace(-2.5, 2.5, 1000)
    y_space = [f(x) for x in x_space]
    plt.cla()
    plt.grid()
    plt.plot(x_space, y_space)
    plt.xlabel(r"$x$")
    plt.ylabel(r"$F_{20}^{*}(x)$")
    plt.savefig("emp_func.pdf")

def draw_polygon(data):
    bins_cnt = ceil(1 + log(len(data))/log(2))
    sampling_range = max(data) - min(data)
    dx = sampling_range / bins_cnt
    bins = [ min(data) + dx * i for i in range(0, bins_cnt + 1) ]
    cnt = [ len(list(filter(lambda x: bins[i-1] <= x <= bins[i], data))) / dx for i in range(1, bins_cnt) ]
    x_space = [ (bins[i] + bins[i+1]) / 2 for i in range(0, len(bins) - 1) ]

    plt.cla()
    plt.xticks(bins, rotation=30)
    plt.grid()
    plt.xlabel(r"$x$")
    plt.ylabel(r"$\frac{n_i}{h}$")
    plt.plot(x_space, cnt)
    plt.savefig("polygon.pdf")

```

```

def draw_hist(data):
    bins_cnt = ceil(1 + log(len(data))/log(2))
    sampling_range = max(data) - min(data)
    h = sampling_range / bins_cnt
    bins = [ min(data) + h * i for i in range(0, bins_cnt + 1) ]
    cnt = [ len(list(filter(lambda x: bins[i-1] <= x <= bins[i], data))) / h for i in range(1, bins_cnt) ]

    plt.cla()
    plt.xticks(bins, rotation=30)
    plt.grid()
    plt.xlabel(r"$x$")
    plt.ylabel(r"$\frac{n_i}{h}$")
    plt.hist(bins[:-1], bins, weights=cnt)
    plt.savefig("hist.pdf")

def compute_fixed_dispersia(data):
    n = len(data)
    dispersia = compute_dispersia(data)
    fd = n / (n-1) * dispersia;
    return fd

print_variation_series(data)
print("")
print_stat_series(data)
print("")
print_extreme_values(data)
print("")
print_sampling_range(data)
print("")
print_estimate_expected_value(data)
print("")
print_dispersia(data)
print("")
print_standard_deviation(data)
print("")
print_most_common_value(data)
print("")
print_median(data)
print("")
print_emp_func(data)
print("")
fd = compute_fixed_dispersia(data)
print(f"Исправленная дисперсия: {fd}")
fstd = (fd)**0.5
print(f"Исправленное стандартное отклонение: {fstd}")
print_emp_func_cases(data)
draw_emp_dist_function(data)
draw_polygon(data)
draw_hist(data)

```