Практическая работа №5

Вариант 12
Выполнил студент группы P3212 Кобелев Роман

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Необходимо вычислить

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

Данные

In [ ]:
```
data = np.array([0.41, 1.63, -1.53, -0.20, 0.85, 0.09, 1.54, 0.25, 1.24, -0.26,
1.08, 0.42, -0.92, -0.91, 1.15, -0.82, 0.26, 0.96, 1.57, 0.72], dtype=float)
```

Вариационный ряд

In [ ]:
```
variation_series = np.sort(data)

print(*variation_series)
```
-1.53 -0.92 -0.91 -0.82 -0.26 -0.2 0.09 0.25 0.26 0.41 0.42 0.72 0.85 0.96 1.08 1.15 1.24 1.54 1.57 1.63

Экстремальные значения и размах

In [ ]:
```
max_value = np.max(data)
min_value = np.min(data)
spread = max_value - min_value

print("Максимальное значение: ", max_value)
print("Минимальное значение: ", min_value)
print("Размах: ", spread)
```
Максимальное значение: 1.63
Минимальное значение: -1.53
Размах: 3.16

Математическое ожидание

Статистичиский ряд

In [ ]:
```
unique_elements, counts = np.unique(variation_series, return_counts=True)
statistical_series = np.array([[key, value] for key, value in zip(unique_elements, counts)])

mean = round(sum(x * n for x, n in statistical_series) / sum(n for _, n in statistical_series), 5)

print("Статистический ряд: \n", statistical_series)
mean
```

Статистический ряд:
```
 [[-1.53  1.  ]
 [-0.92  1.  ]
 [-0.91  1.  ]
 [-0.82  1.  ]
 [-0.26  1.  ]
 [-0.2   1.  ]
 [ 0.09  1.  ]
 [ 0.25  1.  ]
 [ 0.26  1.  ]
 [ 0.41  1.  ]
 [ 0.42  1.  ]
 [ 0.72  1.  ]
 [ 0.85  1.  ]
 [ 0.96  1.  ]
 [ 1.08  1.  ]
 [ 1.15  1.  ]
 [ 1.24  1.  ]
 [ 1.54  1.  ]
 [ 1.57  1.  ]
 [ 1.63  1.  ]]
```
Out[ ]:
```
0.3765
```

Среднеквадратическое отклонение

Выборочная дисперсия

In [ ]:
```
dispersion = round(sum(((x-mean)**2) * n for x, n in statistical_series) / sum(n for _, n in statistical_series), 5  )
standart_deviation = round(np.sqrt(dispersion),5)
true_dispersion = round((len(data) * dispersion)/ (len(data) - 1), 5)
true_standart_deviation = round(np.sqrt(true_dispersion), 5)

print("Выборочная дисперсия: ", dispersion)
print("Выборочное среднеквадратическое отклонение: ", standart_deviation)
print("Исправленная выборочная дисперсия: ", true_dispersion)
print("Исправленное выборочное сдреднеквадратическое отклонение: ", true_standart_deviation)
```
```
Выборочная дисперсия:  0.80627
Выборочное среднеквадратическое отклонение:  0.89793
Исправленная выборочная дисперсия:  0.84871
Исправленное выборочное сдреднеквадратическое отклонение:  0.92125
```

Эмперическая функция распределения

График

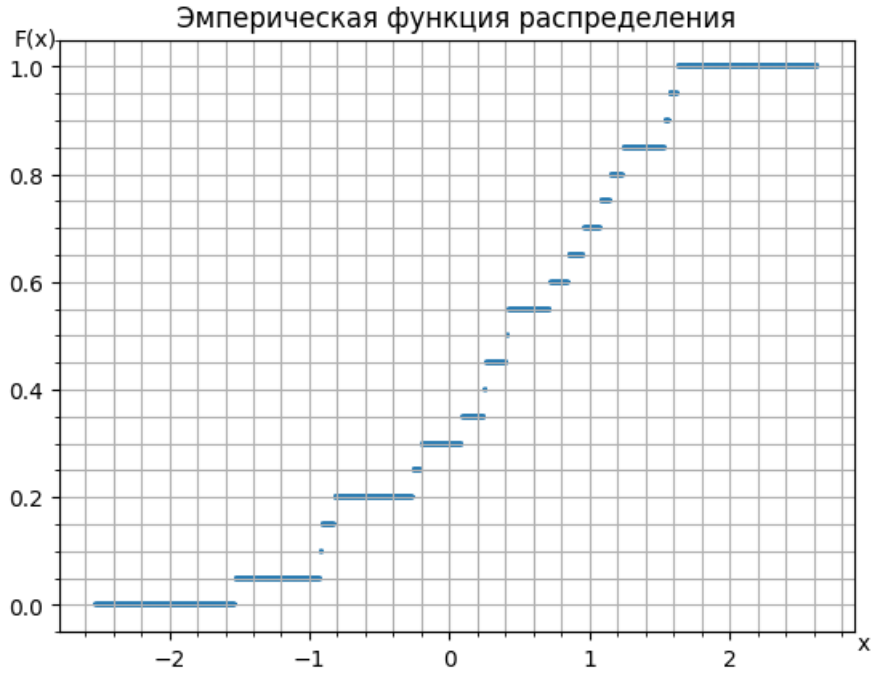
In [ ]:
```
def empirical_distribution_function(x):
    freq = 0
    for i in range(len(statistical_series)):
        if statistical_series[i][0] >= x:
            break
        freq += statistical_series[i][1]
    return freq / len(data)
```
In [ ]:

```
freq = 0
print(f"F(x) = {freq / len(statistical_series)}:\tx < {statistical_series[0][0]}")
for i in range(len(statistical_series) - 1):
    freq += statistical_series[i][1]
    print(
        f"F(x) = {freq / len(statistical_series)}:\t{statistical_series[i][0]} <= x < {statistical_series[i+1][0]}")
freq += statistical_series[-1][1]
print(f"F(x) = {freq / len(statistical_series)}:\t{statistical_series[-1][0]} <= x")

x_values = np.linspace(min_value - 1, max_value + 1, 10000)
ecdf_values = [empirical_distribution_function(x)
               for x in x_values]
plt.grid(which='both')
plt.minorticks_on()
plt.scatter(x_values, ecdf_values, marker='_', s=1)
plt.xlabel('x', ha='right', x=1.0)
plt.ylabel('F(x)', va='top', y=1.0)
plt.gca().yaxis.set_label_coords(-0.03, 1.02)
plt.gca().yaxis.get_label().set_rotation(0)
plt.gca().xaxis.set_label_coords(1.02, 0)
plt.title('Эмпирическая функция распределения')

plt.show()
```

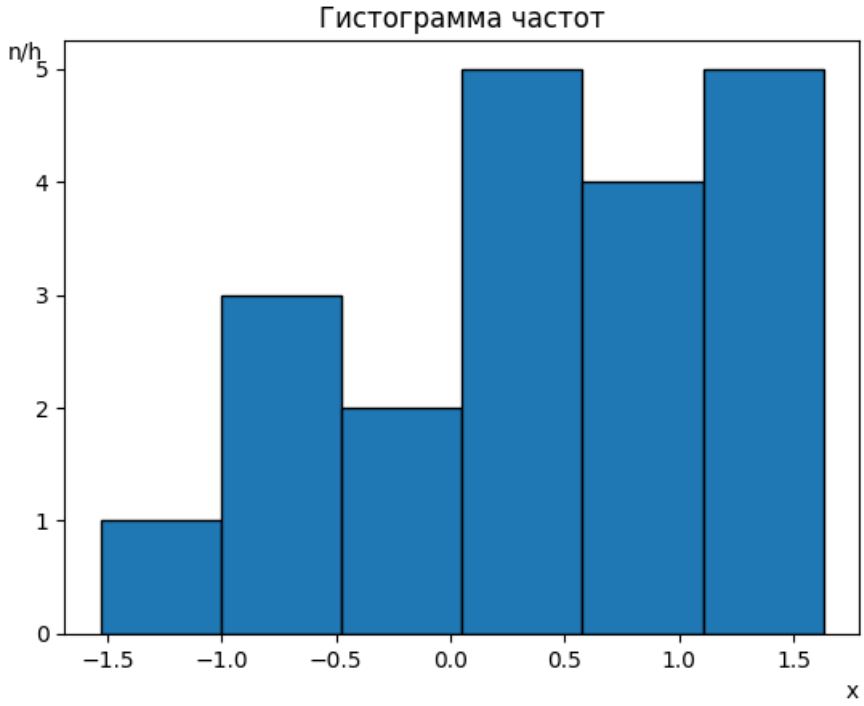
```
F(x) = 0.0: x < -1.53
F(x) = 0.05: -1.53 <= x < -0.92
F(x) = 0.1: -0.92 <= x < -0.91
F(x) = 0.15: -0.91 <= x < -0.82
F(x) = 0.2: -0.82 <= x < -0.26
F(x) = 0.25: -0.26 <= x < -0.2
F(x) = 0.3: -0.2 <= x < 0.09
F(x) = 0.35: 0.09 <= x < 0.25
F(x) = 0.4: 0.25 <= x < 0.26
F(x) = 0.45: 0.26 <= x < 0.41
F(x) = 0.5: 0.41 <= x < 0.42
F(x) = 0.55: 0.42 <= x < 0.72
F(x) = 0.6: 0.72 <= x < 0.85
F(x) = 0.65: 0.85 <= x < 0.96
F(x) = 0.7: 0.96 <= x < 1.08
F(x) = 0.75: 1.08 <= x < 1.15
F(x) = 0.8: 1.15 <= x < 1.24
F(x) = 0.85: 1.24 <= x < 1.54
F(x) = 0.9: 1.54 <= x < 1.57
F(x) = 0.95: 1.57 <= x < 1.63
F(x) = 1.0: 1.63 <= x
```



Гистограмма приведенных частот группированной выборки

In [ ]:

```
num_bins = 6
hist, bins = np.histogram(data, bins=bins_num)
plt.hist(variation_series, bins=num_bins, edgecolor='black')
plt.title('Гистограмма частот')
plt.xlabel('x', ha='right', x=1.0)
plt.ylabel('n/h', va='top', y=1.0)
plt.gca().yaxis.get_label().set_rotation(0)
plt.show()
for i in range(len(bins)-1):
  print(f"Начало: {bins[i]}\t Конец:{bins[i+1]}")
```

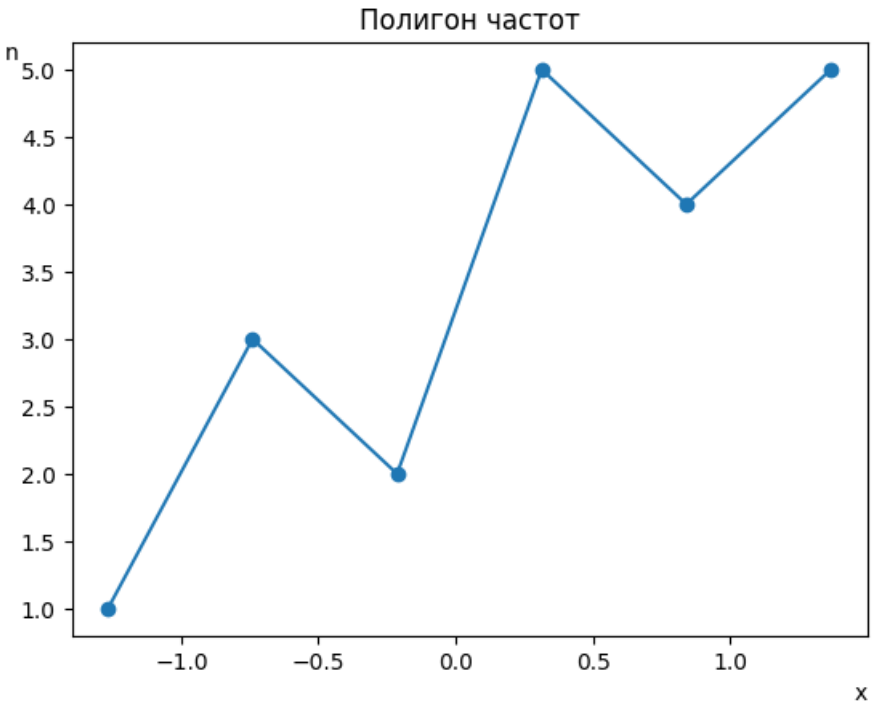



```
Начало: -1.53  Конец:-1.0033333333333334
Начало: -1.0033333333333334  Конец:-0.47666666666666657
Начало: -0.47666666666666657  Конец:0.050000000000000044
Начало: 0.050000000000000044  Конец:0.5766666666666669
Начало: 0.5766666666666669  Конец:1.1033333333333337
Начало: 1.1033333333333337  Конец:1.63
```

Полигон приведенных частот группированной выборки

In [ ]:

```
bin_centers = (bins[:-1] + bins[1:]) / 2
plt.plot(bin_centers, hist, marker='o', linestyle='-')
plt.title('Полигон частот')
plt.xlabel('x', ha='right', x=1.0)
plt.ylabel('n', va='top', y=1.0)
plt.gca().yaxis.get_label().set_rotation(0)
plt.show()
```


Полигон частот
n
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
−1.0
−0.5
0.0
0.5
1.0
x