

Министерство науки и высшего образования Российской Федерации
Муромский институт
федерального государственного бюджетного образовательного учреждения
высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)**

Факультет _____ ИТ _____

Кафедра _____ ИС _____

ПРАКТИЧЕСКАЯ РАБОТА №5

по _____ Специальным главам математики _____

Тема _____ Статистическая обработка экспериментальных данных _____

Руководитель

Щаников С. А.
(фамилия, инициалы)

(подпись) (дата)

Студент ИСм-121
(группа)

Минеев Р. Р.
(фамилия, инициалы)

(подпись) (дата)

Муром 2021

Практическая работа №5.

Тема: Статистическая обработка экспериментальных данных.

Задание на работу: найти численной значение производной функции и рассчитать интеграл функции численным методом.

```
import numpy
import matplotlib.pyplot as plt
import scipy.stats

laplas = lambda x: scipy.stats.norm.cdf(x) - 0.5

my_data = [
    26, 26, 35, 19, 27, 16, 32, 19, 23, 26, 26, 19, 15,
    28, 30, 27, 19, 20, 29, 14, 22, 20, 24, 26, 26, 24,
    25, 28, 28, 30, 29, 21, 27, 25, 18, 20, 24, 21, 18,
    31, 25, 20, 32, 25, 30, 22, 23, 28, 22, 32, 25, 21,
    21, 28, 19, 26, 22, 31, 34, 30, 24, 32, 28, 13, 29,
    20, 26, 23, 28, 29, 30, 27, 27, 15, 23, 18, 29, 16,
    17, 25, 22, 30, 22, 18, 33, 20, 17, 19, 36, 23, 27,
    27, 22, 24, 25, 22, 23, 20, 24, 28, 20, 23, 34, 26,
]

n = int(numpy.sqrt(len(my_data)))
h = (max(my_data) - min(my_data)) / n
X_Y = {i: 0 for i in numpy.arange(min(my_data), max(my_data), h)}
for value in my_data:
    for key in reversed(X_Y.keys()):
        if key < value:
            X_Y[key] += 1
            break

x, y = zip(*sorted([(key, X_Y[key]) for key in X_Y.keys()], key=lambda x: x[0]))
print(x, y)

plt.hist(my_data, bins=int(numpy.sqrt(len(my_data))))
plt.show()

print(numpy.mean(my_data))
print(numpy.var(my_data))
print(numpy.std(my_data))

# Расчёт средней взвешенной
x_ = sum([xf[0]*xf[1] for xf in zip(x, y)]) / sum(y)
# Размах вариации
R = max(x) - min(x)
# Дисперсия
D = sum([(xf[0]-x_)**2 * xf[1] for xf in zip(x, y)]) / sum(y)
# Несмещённая оценка дисперсии
```

					МИВУ 09.04.02-05.001					
Изм	Лист	№ докум.	Подп.	Дата	Практическая работа №5 Статистическая обработка экспериментальных данных			Литера	Лист	Листов
Студент	Минеев Р. Р.		08.01.					У	2	3
Руков.	Щаников С.А.							МИ ВлГУ ИСм-121		
Конс										
Н.контр.										
Утв.										

```

S_2 = sum([(xf[0]-x_)**2 * xf[1] for xf in zip(x, y)]) / (sum(y) - 1)
std = D ** 0.5
# Оценка среднеквадратического отклонения
s = S_2 ** 0.5

# Критерий согласия Пирсона
P = list()
for i in range(len(x)):
    x1 = (x[i] - x_) / s
    if i == len(x) - 1:
        x2 = (max(my_data) - x_) / s
    else:
        x2 = (x[i+1] - x_) / s
    F_x1 = laplas(x1)
    F_x2 = laplas(x2)
    P.append(abs(F_x2 - F_x1))

K = sum([(y[i] - n*P[i])**2 / n*P[i] for i in range(len(P))])
print(K, 'Pirson\'s coef')

(13.0,      15.3,      17.6,      19.900000000000002,      22.200000000000003,      24.500000000000004,
26.800000000000004, 29.100000000000005, 31.400000000000006, 33.7) (3, 4, 10,
20, 13, 16, 20, 8, 5, 4)
24.451923076923077
24.61307322485207
4.961156440271973

17.130309594100634 Pirson's coef

```

Так как Наблюдаемое значение критерия меньше критического значения (17.130309594100634 < 19.3), нулевую гипотезу о нормальном распределении можно принять при данном уровне значимости.

Вывод: в данной практической работе были получены навыки агрегирования данных и проверки данных на законы распределения.

					МИВУ 09.04.02-05.001	Лист
						3
Изм	Лист	№ докум.	Подп.	Дата		