МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДАНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«Московский государственный технический

университет имени Н.Э. Баумана»

(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ \_\_\_ФН\_\_\_

КАФЕДРА  
«ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И МАТЕМАТИЧЕСКАЯ ФИЗИКА»

Направление: Математика и компьютерные науки

Дисциплина: Теория вероятности и математическая статистика

Домашняя работа №1

Группа: \_ФН11-53Б\_

Вариант №9

Студент: Кривяков И. В.

Преподаватель: Облакова Т. В.

Москва 2022

Произведем группировку массива, заданного в виде матрицы .

Находим крайние члены вариационного ряда и размах выборки:

Рассчитываем количество интервалов группировки l по формуле (1), используя встроенную функцию math.trunc , возвращающую целую часть числа, размер выборки и ширину интервалов:

Далее используем встроенную функцию histogram(data,l), аргументами которой служат массив исходных данных data и число интервалов группировки l. Эта функция возвращает два столбца: в первом содержатся частоты попадания элементов из X в каждый из l интервалов, во втором массив с границами каждой ячейки. Для получения средних значений интервалов используем массив binEdgesAverage и произведём вычисления

***Таблица 1 – Интервальный вариационный ряд***

|  |  |  |  |
| --- | --- | --- | --- |
| Интервал | Середина интервала | Частота | Относительная частота |
|  | 3.49 | 12 | 0.1 |
|  | 5.72 | 20 | 0.16667 |
|  | 7.95 | 18 | 0.15 |
|  | 10.18 | 18 | 0.15 |
|  | 12.41 | 17 | 0.14167 |
|  | 14.64 | 20 | 0.16667 |
|  | 16.87 | 15 | 0.125 |

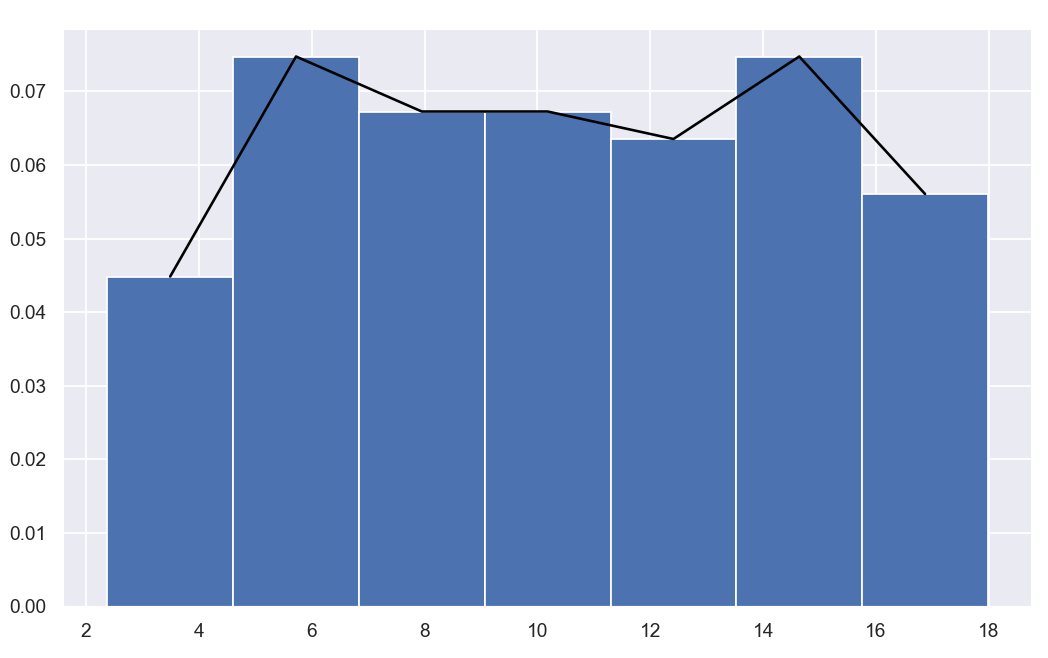
Следующим шагом в обработке выборки является ее графическое представление.

**Определение 4**. *Гистограммой относительных частот* называется фигура, состоящая из прямоугольников с основанием и высотой, равной отношению (*плотность* относительной частоты).

При этом площади прямоугольников равны соответствующим относительным частотам. А поскольку в силу закона больших чисел эти частоты приближают вероятности попадания в интервалы, то гистограмма таким образом графически изображает площади под графиком плотности теоретического закона.

Иногда середины верхних сторон прямоугольников соединяют ломаной и получают *полигон относительных частот*.

При построении гистограммы по оси абсцисс указывается столбец , а по оси ординат – столбец . При этом гистограмма и полигон отличаются только типом графика (гистограмма получается при выборе типа «сплошные столбики», а полигон - это линия)

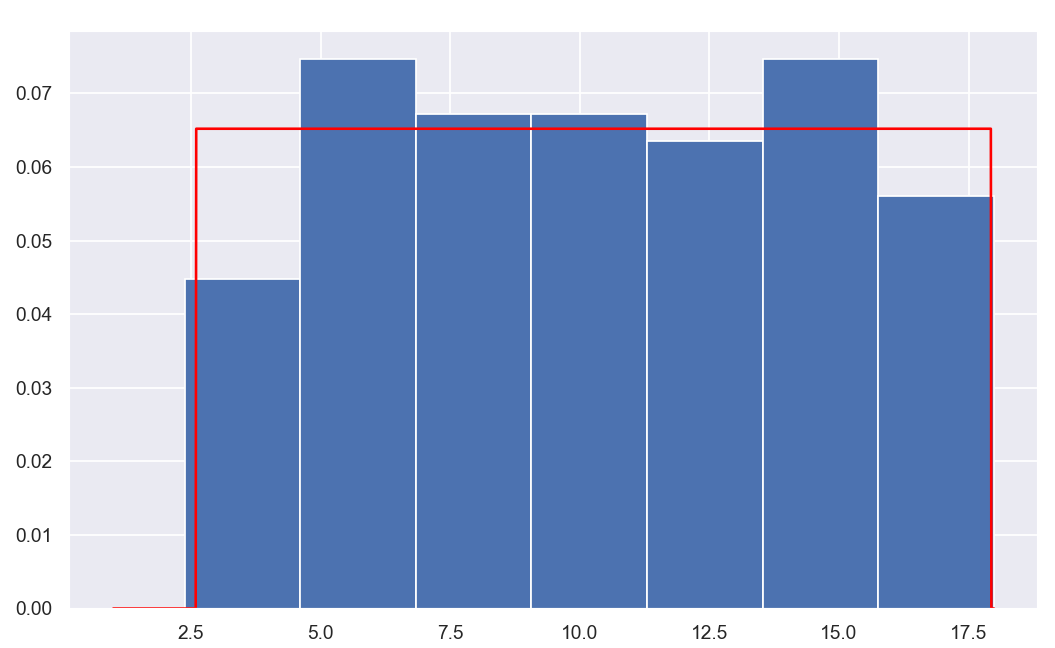


**Замечание**. По виду гистограммы мы можем заключить, что частоты не имеют чёткого максимума для некоторого значения. Это наблюдение позволяет предположить, что теоретический закон распределения выборки является равномерным.

В предположении, что заданная выборка подчиняется равномерному закону с плотностью , найдём параметры a, b этого закона.



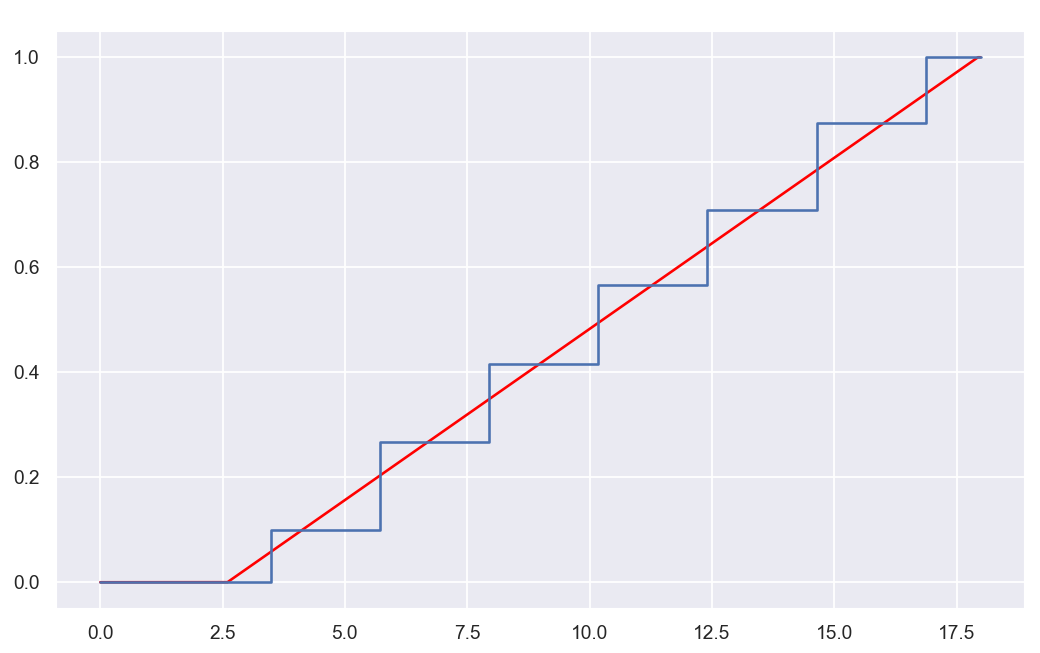
Учитывая и , выборочное среднее и выборочную дисперсию получим, оценки параметров a и b 2.6 и 17.9 соответственно.

Построим совмещенные графики гистограммы и теоретической плотности равномерного закона с найденными параметрами. 

Построим эмпирическую функцию распределения по сгруппированным данным из таблицы 1.

Для этого необходимо рассчитать *накопленные частоты*. Они определяются путем последовательного суммирования относительных частот интервалов. Составим столбец накопленных частот . Первый элемент столбца равен нулю, следующие элементы находятся по формуле при . Следовательно, последний элемент является суммой всех относительных частот и должен быть равен 1.





На рисунке 3 ломаная линия изображает *полигон накопленных частот*. Вершинами этой ломаной являются точки, координаты которых соответствуют границам интервалов и накопленным частотам.

**Вывод**: Первоначальная обработка позволяет предварительно отнести выборку к равномерному закону с параметрами:  
 a = 2.6 и b = 17.9

import numpy as np

import math

import scipy

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

def print\_matrix(x):

    for i in range(0, len(x)):

        for i2 in range(0, len(x[i])):

            print(x[i][i2], end=' ')

        print()

data = [

    [14.495,4.715,7.175,8.428,11.093,3.375,12.906,8.415,8.916,13.4],

    [5.343,17.985,15.992,13.89,9.838,13.924,9.012,9.458,17.69,6.542],

    [14.396,8.592,8.206,14.237,7.357,10.821,12.767,16.058,12.959,4.354],

    [12.888,10.268,9.182,5.647,8.282,2.903,15.988,12.959,14.919,6.339],

    [2.375,17.921,9.097,15.85,11.449,11.095,9.493,12.175,7.479,13.535],

    [9.234,6.078,4.964,6.355,13.957,12.911,15.694,14.286,9.869,5.175],

    [5.811,7.241,5.814,3.086,6.875,3.878,5.333,15.134,12.924,9.159],

    [4.727,4.646,15.535,9.919,17.117,10.351,16.892,12.423,10.511,4.942],

    [4.843,9.927,15.864,3.635,17.963,8.25,5.14,6.734,12.622,13.325],

    [3.377,16.195,12.04,12.768,2.744,14.186,9.354,15.439,14.612,15.649],

    [8.681,5.006,3.608,2.867,12.177,15.506,7.683,14.022,17.103,8.905],

    [12.173,17.757,6.883,2.666,9.861,5.743,16.175,15.308,7.039,15.238]

    ]

data = np.array(data, float)

print("Выборка вариант 9")

print\_matrix(data)

##

print("\n\nНаходим крайние члены вариационного ряда и размах выборки")

max = np.amax(data)

min = np.amin(data)

print("Крайние члены вариационного ряда:\nmax = ", max, "min = ", min)

w = max - min

print("Размах выборки:\nw = ", w)

n = np.size(data)

l = math.trunc(1 + math.log2(n))

h = w / l

print("Размер n = ",n,"\nчисло интервалов l=",l,"ширина интервалов h = ",h)

histogram,binEdges  = np.histogram(data,l)

binEdgesAverage = np.zeros(binEdges[:l].size)

for i in range (l):

    binEdgesAverage[i] = (binEdges[i] + binEdges[i + 1]) / 2

print("Средние значения интервалов binEdgesAverage =",binEdgesAverage)

print("Итнтервалы", binEdges)

p = histogram / n

print("Относительные частоты p = ",np.around(p,5))

print("Частоты histogram = ", histogram)

f1 = p / h

print("f1 = ",np.around(f1,7))

##

sns.set\_theme()

plt.figure(figsize=(10,6))

x = binEdgesAverage

y = f1

plt.bar(x, y, width=h)

plt.plot(x, y, color = 'black')

plt.show()

##

xm = 0

for i in range(data.shape[0]):

    for j in range(data.shape[1]):

        xm += data[i, j]

xm /= n

print("\n\nDыборочное среднее xm = ", xm)

s2 = 0

for i in range(data.shape[0]):

    for j in range(data.shape[1]):

        s2 += (data[i, j] - xm) \*\* 2

s2 /= (n - 1)

print("Выборочная дисперсия s2 = ", s2)

##

plt.figure(figsize=(10,6))

x = binEdgesAverage

y = f1

plt.bar(x, y, width=h)

x1 = np.arange(1,18,0.01)

y1 = scipy.stats.uniform.pdf(x1, loc = 2.598023243281459, scale = 17.93683675671854 - 2.598023243281459)

print(y)

plt.plot(x1, y1, color = 'red')

plt.show()

##

sum = 0

kum =  np.zeros(binEdges[:l+1].size)

for i in range(l):

    sum += p[i]

    kum[i+1] = sum

print(np.around(kum,5), "\n")

##

plt.figure(figsize=(10,6))

x1 = np.arange(0,18,0.01)

y1 = scipy.stats.uniform.cdf(x1, loc = 2.598023243281459, scale = 17.93683675671854 - 2.598023243281459)

plt.plot(x1, y1, color = 'red')

plt.step([0] + binEdgesAverage.tolist()+[18],[0] + kum.tolist())

plt.show()