

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий
Кафедра “Прикладная математика”
Дисциплина “Стохастический анализ”

Отчет по лабораторной работе №1

Вариант 4

Выполнила:
студент гр. 16-ПМ
Горенкова А. В.

Проверила:
Шерстнева Л.В.

Нижний Новгород
2019

Задание

1. Вычислить числовые характеристики выборки: средневывборочное, дисперсию, среднеквадратичное отклонение, медиану, моду, эксцесс, асимметрию
2. Сделать предварительную проверку выборки на нормальность распределения.
3. Построить эмпирическую функцию распределения, гистограмму и полигон частот.

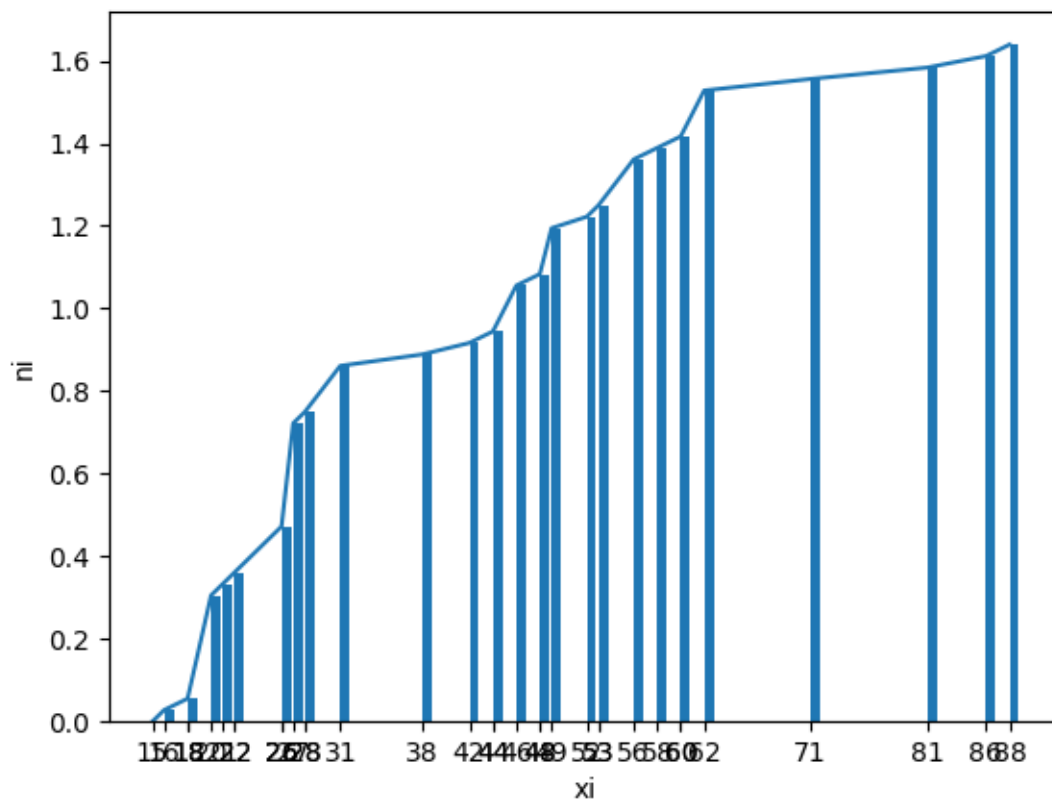
Исходная выборка:

4. 35 42; 21; 31; 18; 27; 28; 48; 26; 28; 62; 56; 26; 18; 22; 26;
52; 16; 53; 20; 46; 48; 81; 38; 88; 86; 58; 49; 53; 18; 22; 60; 44; 60; 44;
71.

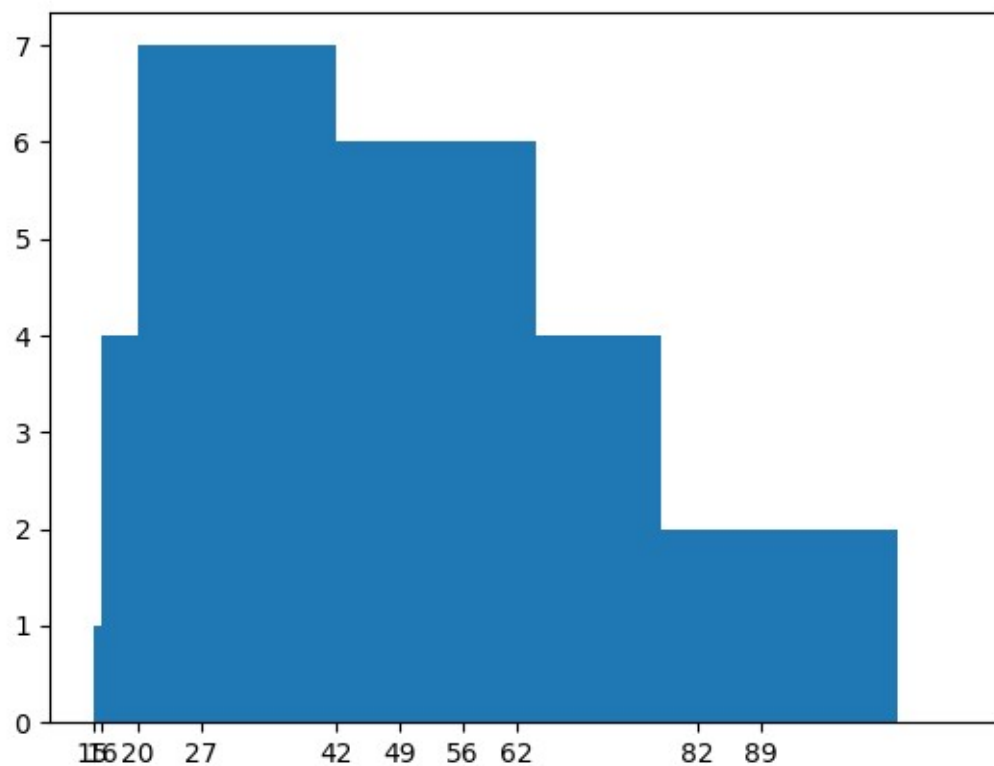
Результат работы

```
[16.0, 18.0, 18.0, 18.0, 20.0, 21.0, 22.0, 22.0, 26.0, 26.0, 26.0, 27.0, 28.0, 28.0, 31.0, 38.0, 42.0, 44.0, 44.0, 46.0, 48.0, 48.0, 49.0, 52.0, 53.0, 53.0, 56.0, 58.0, 60.0, 60.0, 62.0, 71.0, 81.0, 86.0, 88.0]  
n: 35  
Sredne vuborochnoe: 42.457142857142856  
Dispersia: 401.27673469387764  
Sr kv otklonenie: 20.03189293835901  
coeff variacii 47.18144366369888  
Mediana: 44.0  
Moda: 18.0  
Excess: 0.0579813055993883  
Assim: 0.12897904664269658
```

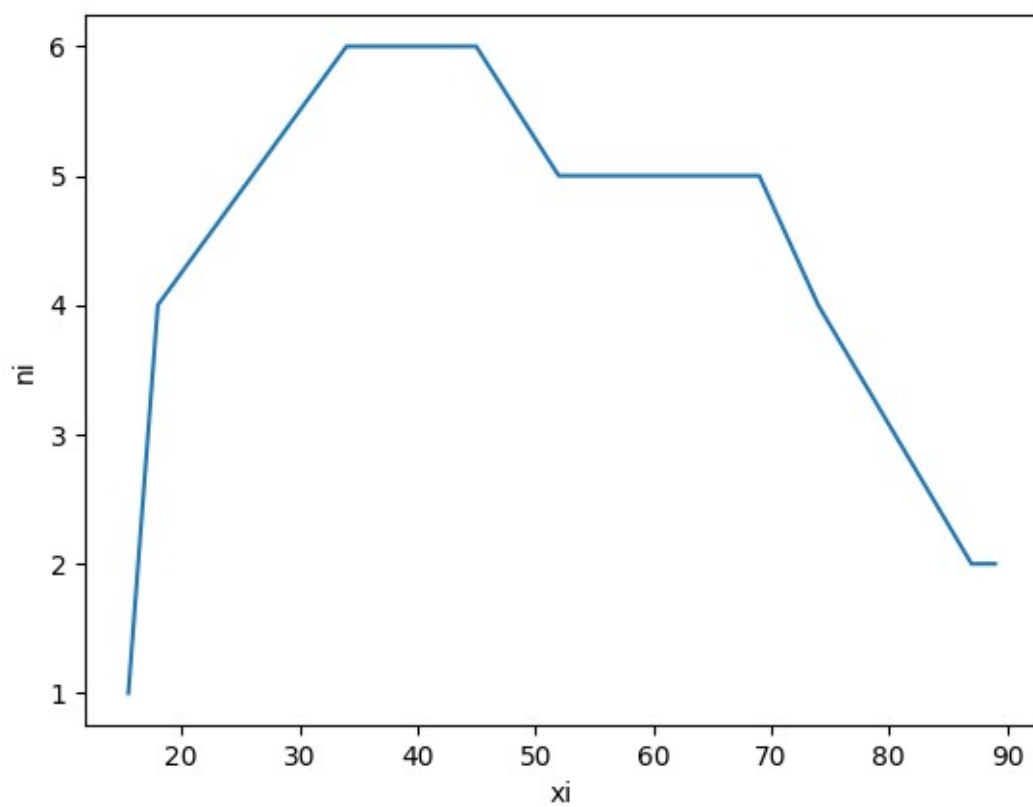
Эмпирическая функция:



Гистограмма:



Полигон:



Исходный код

main.py

```
from func import *
```

```
v = [42.0, 21.0, 31.0, 18.0, 27.0, 28.0, 48.0, 26.0, 28.0, 62.0, 56.0, 26.0, 18.0, 22.0,
26.0,
52.0, 16.0, 53.0, 20.0, 46.0, 48.0, 81.0, 38.0, 88.0, 86.0, 58.0, 49.0, 53.0, 18.0, 22.0,
60.0, 44.0, 60.0, 44.0,
71.0]
```

```
sortedV = sorted(v)
```

```
print(sortedV)
```

```
n = len(v)
```

```
print("n: ", n)
```

```
Xsr = find_Xsr(v)
```

```
print("Sredne vuborochnoe: ", Xsr)
```

```
S2 = find_Dispers(v, Xsr)
```

```
print("Dispersia: ", S2)
```

```
S = S2**(0.5)
```

```
print("Sr kv otklonenie: ", S)
```

```
V = 100*S/Xsr
```

```
print("coeff variacii", V)
```

```
Me = find_Median(v)
```

```
print("Mediana: ", Me)
```

```
ni = find_ni(v)
```

```
Mo = find_Moda(v, ni)
```

```
print("Moda: ", Mo)
```

```
Ex = find_Ex(v, ni, Xsr, S)
```

```
print("Excess: ", Ex)
```

```
As = find_As(v, ni, Xsr, S)
```

```
print("Assim: ", As)
```

```
poligon(v)
```

```
gistogramm(v)
```

```
empir(v)
```

func.py

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

```
def find_Xsr(v):
    Xsr = 0
    n = len(v)
    for i in range(n):
        Xsr += v[i]/n
    return Xsr
```

```
def find_Dispers(v, Xsr):
    S2 = 0
    n = len(v)
    for i in range(n):
        S2 += ((v[i] - Xsr)**2)/n
    return S2
```

```
def find_Median(v):
    n = len(v)
    sortedV = sorted(v)
    if n%2 == 0:
        return (sortedV[int(n/2)] + sortedV[int(n/2 + 1)])/2
    else:
        return sortedV[int(n/2 + 1)]
```

```
def find_ni(v):
    ni = []
    for i in range(len(v)):
        ni.append(1.0)
        for j in range(i):
            if v[j] == v[i]:
                ni[j] += 1.0
                ni[i] += 1.0
    # for i in range(len(ni)):
    #     ni[i] = ni[i]/(len(v))
    return ni
```

```
def find_ni_int(v, vint):
    ni = []
    for i in range(1, len(vint)):
        ni.append(0)
        for j in range(len(v)):
            if v[j] == vint[i]:
                ni[i] += 1
```

```

        if vint[i-1] < v[j] <= vint[i]:
            ni[i-1] += 1
    return ni

def find_Moda(v, ni):
    Moi = 0
    for i in range(len(v)):
        if ni[i] > ni[Moi]:
            Moi = i
    return v[Moi]

def find_Ex(v, ni, Xsr, S):
    m4 = 0
    n = len(v)
    for i in range(n):
        m4 += ((v[i] - Xsr)**4.0)*ni[i]/n
    Ex = (m4/S**4.0) - 3.0

    # if Ex < 0:
    #     print("Ne norm. raspr.")
    # else:
    #     print("Norm. raspr.")
    return Ex

def find_As(v, ni, Xsr, S):
    m3 = 0
    n = len(v)
    for i in range(n):
        m3 += ((v[i] - Xsr)**3.0)*ni[i]/n
    return m3/S**3

def Fn(x, v, ni):
    Sum = 0.0
    n = len(v)
    for i in range(n):
        if v[i] < x:
            Sum += float(ni[i])/float(n)
    return Sum

def empir(v):
    v = sorted(v)
    ni = find_ni(v)
    F = []
    for i in range(len(v)):
        F.append(Fn(v[i], v, ni))

```

```
ax = plt.gca()
ax.bar(v, F, align='edge')
ax.set_xticks(v)
plt.plot(v, F)
plt.ylabel('ni')
plt.xlabel('xi')
plt.show()
```

```
def poligon(v):
    v = sorted(v)
    vint = []
    vi = [15.5, 18, 34, 45, 52, 69, 74, 87, 89]
    vint = [15, 16, 20, 26, 42, 48, 56, 62, 82, 89]
    ni = find_ni_int(v, vint)
    plt.plot(vi, ni)
    # plt.label("poligon")
    plt.ylabel('ni')
    plt.xlabel('xi')
    plt.show()
```

```
def gistogramm(v):
    v.append(15)
    v = sorted(v)
    vint = [15, 16, 20, 27, 42, 49, 56, 62, 82, 89]
    ni = find_ni_int(v, vint)
    ni.append(0)
    plt.bar(vint, ni, width = 22, align = 'edge')
    plt.xticks(vint)
    plt.show()
```