

Государственное бюджетное профессиональное образовательное учреждение города Москвы
"Московский колледж управления, гостиничного бизнеса и информационных технологий
"Царицыно" Отделение управления и информационных технологий.

Теория вероятностей и математическая статистика

Индивидуальный проект по теме 2:

Построение эмпирических распределений случайных данных

Студент: Уколов Алексей

Группа: П2-3

Преподаватель: к.ф.-м.н. Мещеряков В.В.

Москва 2019

Аннотация

Пояснительная записка содержит теоретический материал, примеры решений задач по теме "Построение графиков эмпирических распределений случайных данных" дисциплины "Теория вероятностей и математическая статистика".

Оглавление

1	Определение эмпирических распределений	4
1.1	Функция эмпирического кумулятивного распределения	4
1.2	Гистограмма распределенных данных	9
1.3	Эмпирическое распределение вероятностей	17
2	Графическое отображение случайных данных на Scilab	20
2.1	Эмпирическое кумулятивное распределение	20
2.2	Высокоуровневые реализации эмпирических распределений	23
3	Высокоуровневые средства Python.....	23
3.1	Непрерывные распределения	25
3.2	Равномерное распределение	25
3.3	Равномерное распределение	27

Определение эмпирических распределений

Функция эмпирического кумулятивного распределения

Для заданного одномерного массива случайных данных можно определить функцию эмпирического кумулятивного распределения

$$F(x) = \frac{1}{N} \sum_{i=1}^N 0(x - x_i)$$

где x – случайная переменная, x_i – значение наблюдаемой величины, полученной в i -ом опыте и записанной в массив данных, N – полное число наблюдений, функция Хевисайда.

$$\theta(\varepsilon) = \begin{cases} 0, & \varepsilon > 0. \\ 1, & \varepsilon \leq 0, \end{cases}$$

Из определения формулы следует, что в геометрическом смысле функция эмпирического кумулятивного распределения первой формулы представляет собой огибающую суммы N ступенек, каждая из которых расположена в точке с координатой x_i и имеет высоту $1/N$. Высоты ступенек огибающей определяют относительную частоту величин данных, которые меньше или равны текущей величине.

По ссылке https://en.wikipedia.org/wiki/Empirical_distribution_function

можно убедиться, что слово «кумулятивного» в названии этого распределения можно опустить. Построение графика кумулятивной функции требует частого обращения к функции Хевисайда.

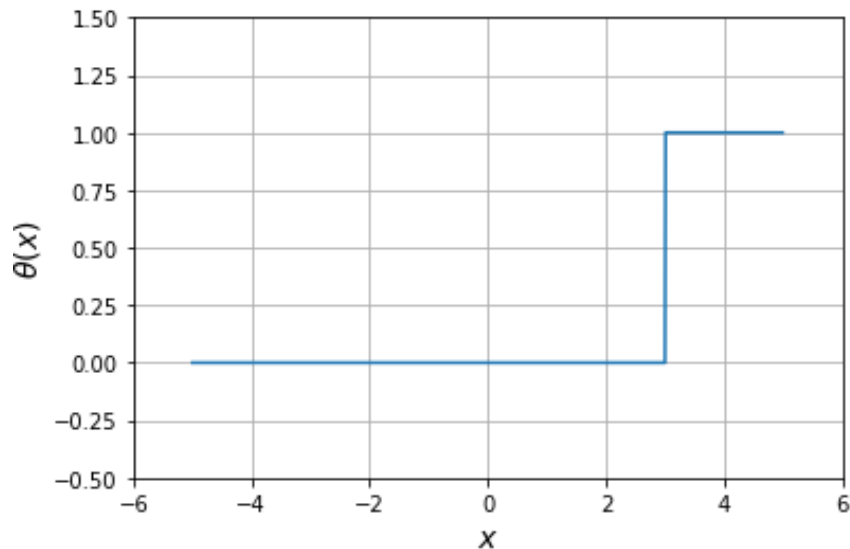
Задача №1 Построить в среде Spyder программную функцию heaviside, основываясь на алгебраическом определении функции Хевисайда (по второй формуле). Проверить корректность работы функции heaviside построением графика функции $\theta(x - 2)$.

Модуль heaviside.py

```
def heavyside(x, a):  
    return 1 * (x > a)
```

Скрипт heaviside_run.py

```
import matplotlib.pyplot as plt  
import numpy as np  
import heaviside as step  
a = 3  
xmin = -5; xmax = 5; dx = 1e-2;  
xi = np.arange(xmin, xmax, dx)  
f = step.heavyside(xi, a); plt.plot(xi, f)  
plt.axis([xmin-1, xmax+1, -0.5, 1.5])  
plt.xlabel(r'$x$', fontsize=14, color='black')  
plt.ylabel(r'$\theta(x)$', fontsize=14, color='black')  
plt.grid(True)
```



Функция Хевисайда $\theta(x - 2)$.

Задача 2. Построить в двух горизонтально расположенных подокнах график случайных данных $\text{data} = [1.5, 5.2, 8.6, 3.0, 5.6, 6.3, 2.7, 9.5, 5.8, 6.8, 5.7, 2, 4.3, 5.6, 1.5]$ и их кумулятивное распределение.

Скрипт `ecdf_example_1.py`

```
import matplotlib.pyplot as plt
import numpy as np
import heaviside as step
# Случайные данные
data = np.array([1.5, 5.2, 8.6, 3.0, 5.6, 6.3, 2.7, 9.5, 5.8, 6.8,
5.7, 2, 4.3, 5.6, 1.5])
N = len(data)
fig = plt.gcf()
fig.set_size_inches(10,5)
plt.subplot(121)
n = np.arange(1,N+1,1)
plt.plot(n,data,'ro--')
plt.axis([0,N+1,0,max(data)+1])
plt.xlabel('$n$', fontsize=14, color='black');
plt.ylabel('$data$', fontsize=14, color='black')
plt.xticks(np.arange(0,N+1,1))
plt.grid()
plt.subplot(122)
m = 1000;
x = np.linspace(0,max(data)+1,m);
sumtheta = np.zeros(m)
for n in range(0,N):
    sumtheta = sumtheta + step.heavyside(x,data[n])
F = sumtheta/N
plt.plot(x,F)
```

```
plt.axis([0, max(data)+2, 0, 1.2])
plt.xlabel('$x$', fontsize=14, color='black');
plt.ylabel('$F$', fontsize=14, color='black')
plt.xticks(np.arange(0, max(data)+2, 1))
plt.grid()
```

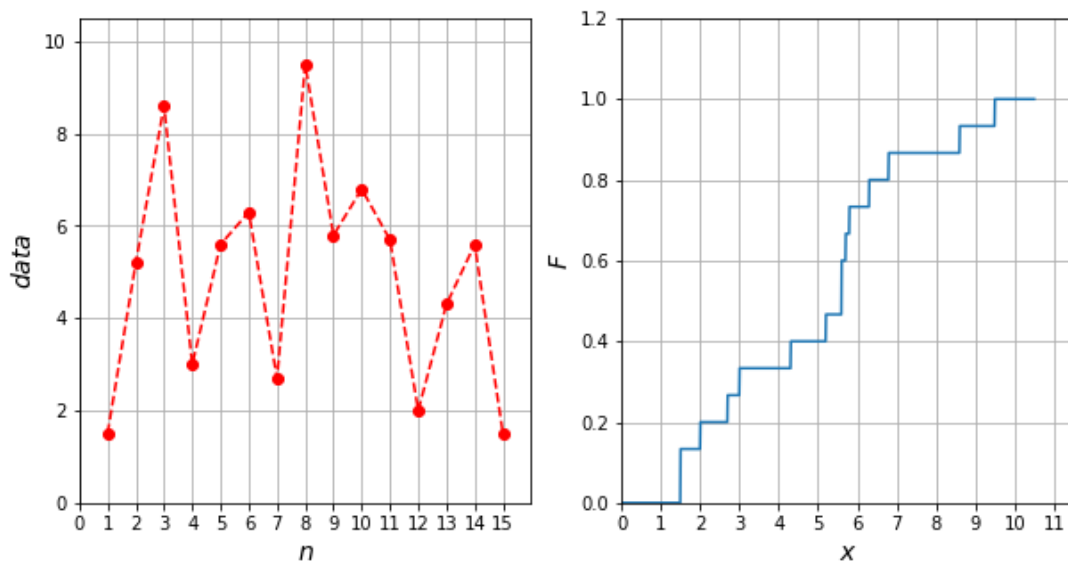


График данных и их кумулятивное распределение в задаче

Задача 3 Найти аналитический вид функции эмпирического кумулятивного распределения чисел $data = [29, 56, 32, 16, 56]$ электронных заказов в интернет - магазине за 5 дней торгов. Построить график кумуляты по ее явному виду и, основываясь на программном коде решения предыдущей задачи в той задаче, дать графическую иллюстрацию числам заказов и проверить аналитическое решение.

Для числа данных $N=5$ по первой и второй формуле получим:

$$F(x) = \frac{1}{5} \left\{ \frac{1}{2} \left(\frac{|x-16|}{(x-16)} + 1 \right) + \frac{2}{2} \left(\frac{|x-29|}{(x-29)} + 1 \right) + \frac{1}{2} \left(\frac{|x-32|}{(x-32)} + 1 \right) + \frac{1}{2} \left(\frac{|x-56|}{(x-56)} + 1 \right) \right\} =$$

$$= \frac{|x-16|}{10(x-16)} + \frac{|x-29|}{5(x-29)} + \frac{|x-32|}{10(x-32)} + \frac{|x-56|}{10(x-56)} + \frac{1}{2}$$

Более кратко и более просто искомая функция выражается через функцию Хевисайда:

$$F(x) = \frac{\theta(x-16)}{5} + \frac{2\theta(x-29)}{5} + \frac{\theta(x-32)}{5} + \frac{\theta(x-56)}{5}$$

Программное решение:

Скрипт `ecdf_example_2.py`

```
import matplotlib.pyplot as plt
import numpy as np
import heaviside as step
# Случайные данные
data = np.array([29, 56, 32, 16, 56])
N = len(data)
fig = plt.gcf()
fig.set_size_inches(10,5)
plt.subplot(121)
n = np.arange(1,N+1,1)
plt.plot(n,data,'ro--')
plt.axis([0,N+1,0,max(data)+10])
plt.xlabel('$n$', fontsize=14, color='black');
plt.ylabel('$data$', fontsize=14, color='black')
plt.xticks(np.arange(0,N+1,1))
plt.grid()
plt.subplot(122)
m = 1000;
x = np.linspace(0,max(data)+13,m);
sumtheta = np.zeros(m)
for n in range(0,N):
    #print(n)
    sumtheta = sumtheta + step.heaviside(x,data[n])
F = sumtheta/N
plt.plot(x,F,color = 'yellow',linewidth = 6)
#Явный вид кумулятивной функции
F = (step.heaviside(x,16)/5 + (2*step.heaviside(x,29))/5 +
step.heaviside(x,32)/5 + step.heaviside(x,56)/5)
plt.plot(x,F,color = 'red',linewidth = 1)
plt.axis([0, max(data)+13, 0 ,1.2])
plt.xlabel('$x$', fontsize=14, color='black');
plt.ylabel('$F$', fontsize=14, color='black')
plt.xticks(np.arange(0,110,10))
plt.grid()
```

В правом подокне желтая линия определяет график эмпирической кумулятивной функции, красная линия соответствует графику кумулятивной функции, найденной по ее явному аналитическому виду.

Рис.2.1

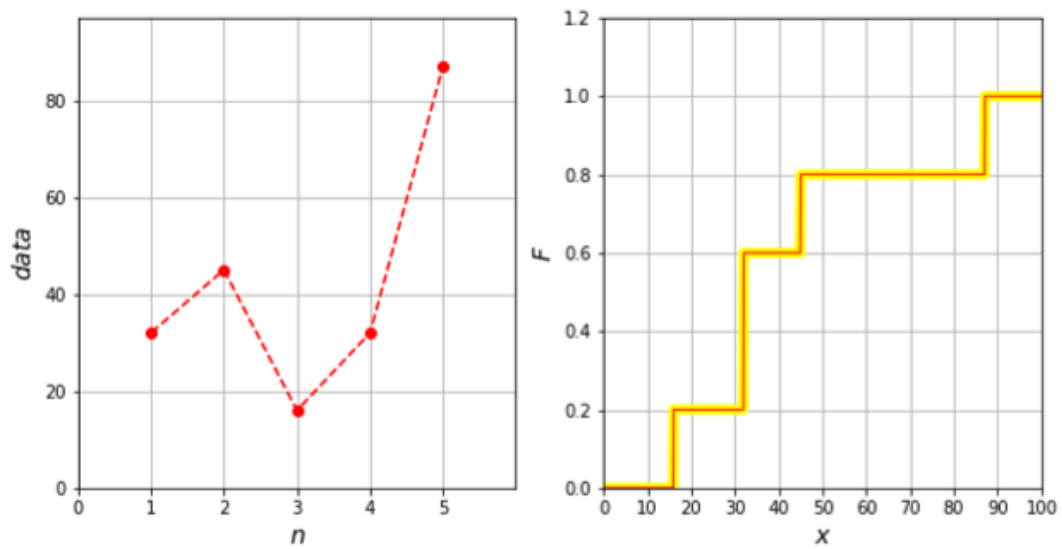
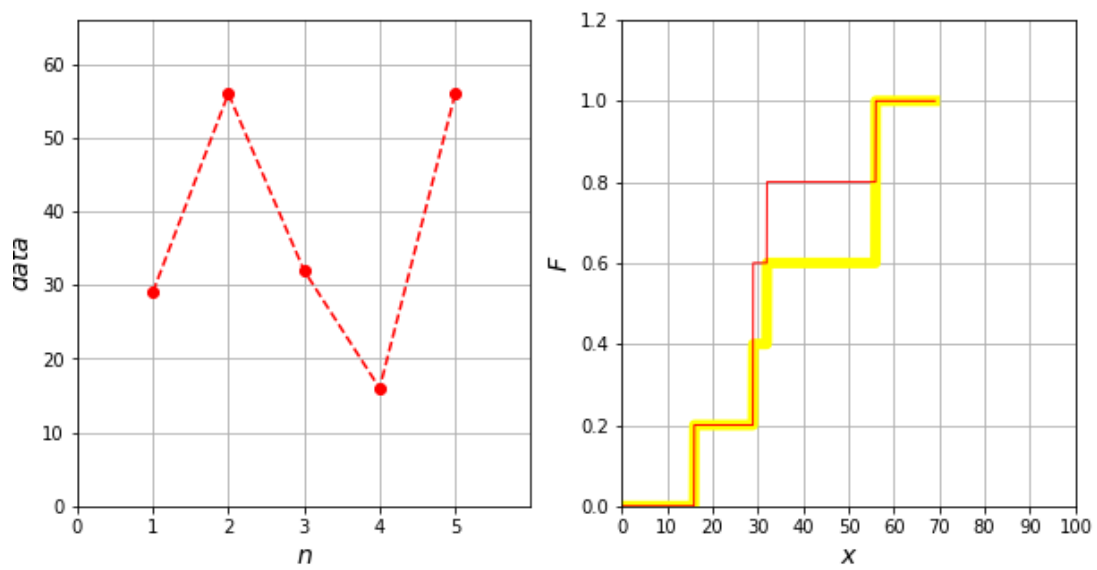


Рис 2.2. График данных и их кумулятивное распределение в задаче №6.



```
data = np.array([29, 56, 32, 16, 56])
```


Гистограмма распределенных данных

Гистограммой распределения данных называют столбчатый график частот, относительных частот или плотности относительных частот величин случайных данных, принимаемых в зависимости от интервалов значений случайной переменной x . Для построения гистограммы диапазон наблюдаемых величин x разбивают на $nbins$ интервалов (бинов) длиной Δx по правилу

$$\Delta x = \frac{x_{max} - x_{min}}{nbins}$$

где x_{max} и x_{min} – соответственно максимальное и минимальное значение наблюдаемой величины. Затем подсчитывают частоту n_i значений данных, попавших в i -ый интервал, и откладывают либо саму величину n_i , либо величину f_i называют плотностью относительной частоты

$$f_i = \frac{n_i}{N\Delta x}$$

Задача 4.

Построить гистограмму случайных чисел $data = [1.5, 5.2, 8.6, 3.0, 5.6, 6.3, 2.7, 9.5, 5.8, 6.8, 2.5, 5.6]$

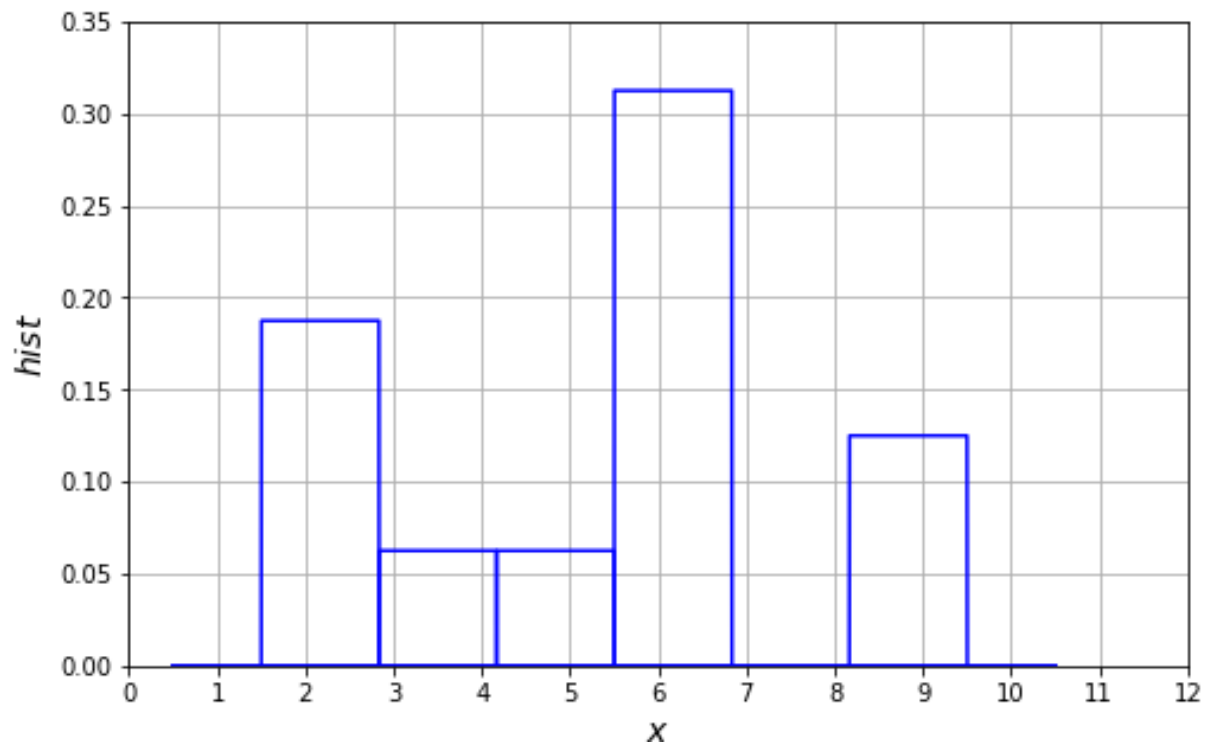
с числом бинов $nbins = 6$. Предусмотреть вывод числа данных и относительного числа данных, попавших в соответствующий бин. Для вычисления частот n , величины бина $lbin$, минимального $xmin$ и максимального $xmax$ значения данных и их числа N построим программную функцию.

Модуль hist1D.py

```
def hist1D(nbins,data):
    import numpy as np
    N = len(data)
    xmin = np.min(data);
    xmax = np.max(data)
    lbin = (xmax-xmin)/nbins
    sortdata = np.sort(data)
    n = np.zeros(nbins,int);
    x1 = np.zeros(nbins);
    x2 = np.zeros(nbins);
    for p in range(nbins):
        for k in range(N):
            # Границы бинов
            x1[p] = xmin + (p)*lbin; x2[p] = xmin + (p+1)*lbin;
            if sortdata[k] >= x1[p]:
                if sortdata[k] <= x2[p]:
                    n[p] = n[p] + 1;
    f = n/N/lbin;
    return n,f,lbin,xmin,xmax,N
```

Скрипт hist1D_run.py

```
import matplotlib.pyplot as plt
import numpy as np
import heaviside as step
import hist1D as hist
nbins = 6
data = np.array([1.5, 5.2, 8.6, 3.0, 5.6, 6.3, 2.7, 9.5, 5.8,
6.8, 2.5, 5.6])
n,f,lbin,xmin,xmax,N = hist.hist1D(nbins,data)
print('n = ');
print(n)
print('f = ');
print(f)
print('lbin = ');
print(lbin)
print('xmin = ');
print(xmin)
print('xmax = ');
print(xmax)
print('N = ');
print(N)
fig = plt.gcf()
fig.set_size_inches(8,5)
m = 10**4; x = np.linspace(xmin - 1,xmax + 1,m);
barhist = [np.zeros(m) for k in range(nbins)]
# Вещественная переменная k, так как в python умножение целого i
на
# вещественное lbin приводит к округлению до целого
k = 0.0
for i in range(nbins):
    k = k + 1.0
    barhist[i] = f[i]*(step.heavyside(x, (xmin+(k-1)*lbin)) -
        step.heavyside(x, (xmin+k*lbin)))
    plt.plot(x,barhist[i], color='blue')
    plt.axis([0, 10, 0, 0.35])
    plt.xlabel('$x$', fontsize=14, color='black')
    plt.ylabel('$hist$', fontsize=14, color='black')
    plt.xticks(np.arange(0,N+1,1))
    plt.grid(True)
```



Гистограмма данных $\text{data} = [1.5, 5.2, 8.6, 3.0, 5.6, 6.3, 2.7, 9.5, 5.8, 6.8, 2.5, 5.6]$.

IPython console:

```
n = [3 1 1 5 0 2]
```

```
f = [0.1875 0.0625 0.0625 0.3125 0. 0.125 ]
```

```
lbin = 1.3333333333333333
```

```
xmin = 1.5
```

```
xmax = 9.5
```

```
N = 12
```

Задача № 5

Найти аналитический вид гистограммы чисел $\text{data} = [56, 45, 32, 16, 43, 33, 44, 55]$ электронных заказов и интернет - магазине за 5 дней торгов. Для числа бинов $\text{nbins} = 4$ построить гистограммы частот данных и нормированные гистограммы для случаев численного суммирования функций Хевисайда и по явному виду функций, описывающих эти гистограммы.

Скрипт hist1D_run2.py

```
import matplotlib.pyplot as plt
import numpy as np
import heaviside as step
import hist1D as hist
nbins = 4
data = np.array([56, 45, 32, 16, 43, 33, 44, 55])
n,f,lbin,xmin,xmax,N = hist.hist1D(nbins,data)
print('n = ');
print(n)
print('f = ');
print(f)
print('lbin = ');
print(lbin)
print('xmin = ');
print(xmin)
print('xmax = ');
print(xmax)
print('N = ');
print(N)
fig = plt.gcf()
fig.set_size_inches(8,8)
plt.subplot(211) # Гистограмма частот
m = 10**4; x = np.linspace(xmin - 1,xmax + 1,m);
barhist = [np.zeros(m) for k in range(nbins+1)]
#barhist = zeros(nbins,m);
meanbin = np.zeros(nbins)
k = 0.0
for i in range(nbins):
    k = k + 1.0
    meanbin[i] = xmin+(2*k - 1)*lbin/2;
    # Вектор середин бинов
    barhist[i] = n[i]*(step.heavyside(x,meanbin[i] -
lbin/2) -      step.heavyside(x,meanbin[i] + lbin/2));
    #Hn(i,:) = n(i)*(heaviside(x - (xmin+(i-1)*lbin))
- ...
    #heaviside(x - (xmin+i*lbin)));
    plt.plot(x,barhist[i], color='blue')
    plt.ylim([0, 6.5])
    plt.xlabel('$x$', fontsize=14, color='black')
    plt.ylabel('$n$', fontsize=14, color='black')
```

```

plt.grid(True)
print('meanbin = ');
print(meanbin)
plt.subplot(212) # Нормированная гистограмма
m = 10**4; x = np.linspace(xmin - 1, xmax + 1, m);
barhist = [np.zeros(m) for k in range(nbins+1)]
meanbin = np.zeros(nbins)
k = 0.0
for i in range(nbins):
    k = k + 1.0
    meanbin[i] = xmin+(2*k - 1)*lbin/2;
    barhist[i] = f[i]*(step.heavyside(x, meanbin[i] -
lbin/2) -
    step.heavyside(x, meanbin[i] + lbin/2));
    #Hn(i,:) = n(i)*(heaviside(x - (xmin+(i-1)*lbin)) -
...
    #heaviside(x - (xmin+i*lbin)));
plt.plot(x, barhist[i], color='blue')
plt.ylim([0, 0.04])
plt.xlabel('$x$', fontsize=14, color='black')
plt.ylabel('$f$', fontsize=14, color='black')
plt.grid(True)

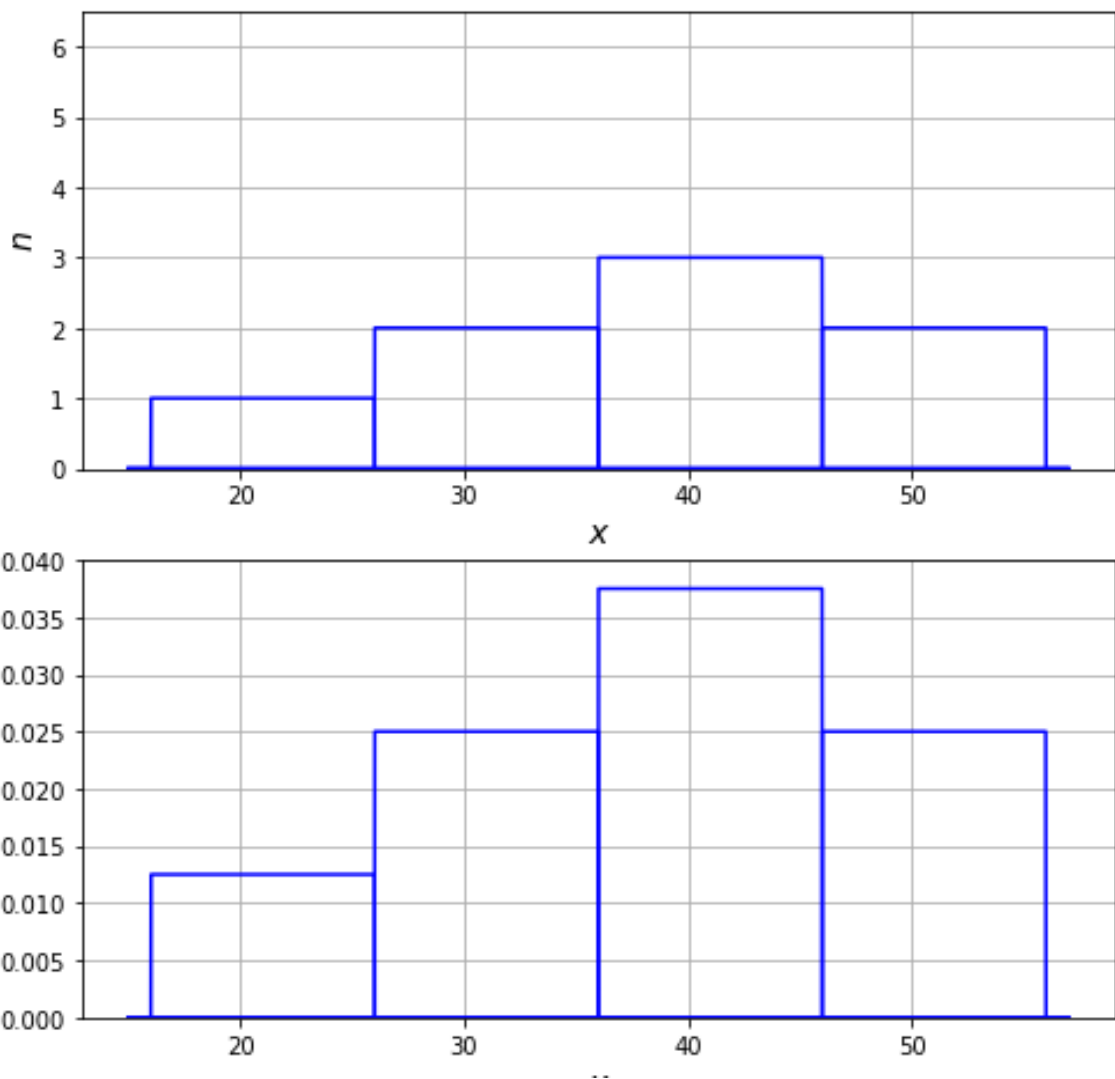
```

IPython console:

```

n = [1 2 3 2]
f = [0.0125 0.025 0.0375 0.025 ]
lbin = 10.0
xmin = 16
xmax = 56
N = 8
meanbin = [21. 0. 0. 0.]
meanbin = [21. 31. 0. 0.]
meanbin = [21. 31. 41. 0.]
meanbin = [21. 31. 41. 51.]

```



Гистограммы данных $\text{data} = [56, 45, 32, 16, 43, 33, 44, 55]$, полученные программным вычислением частот.

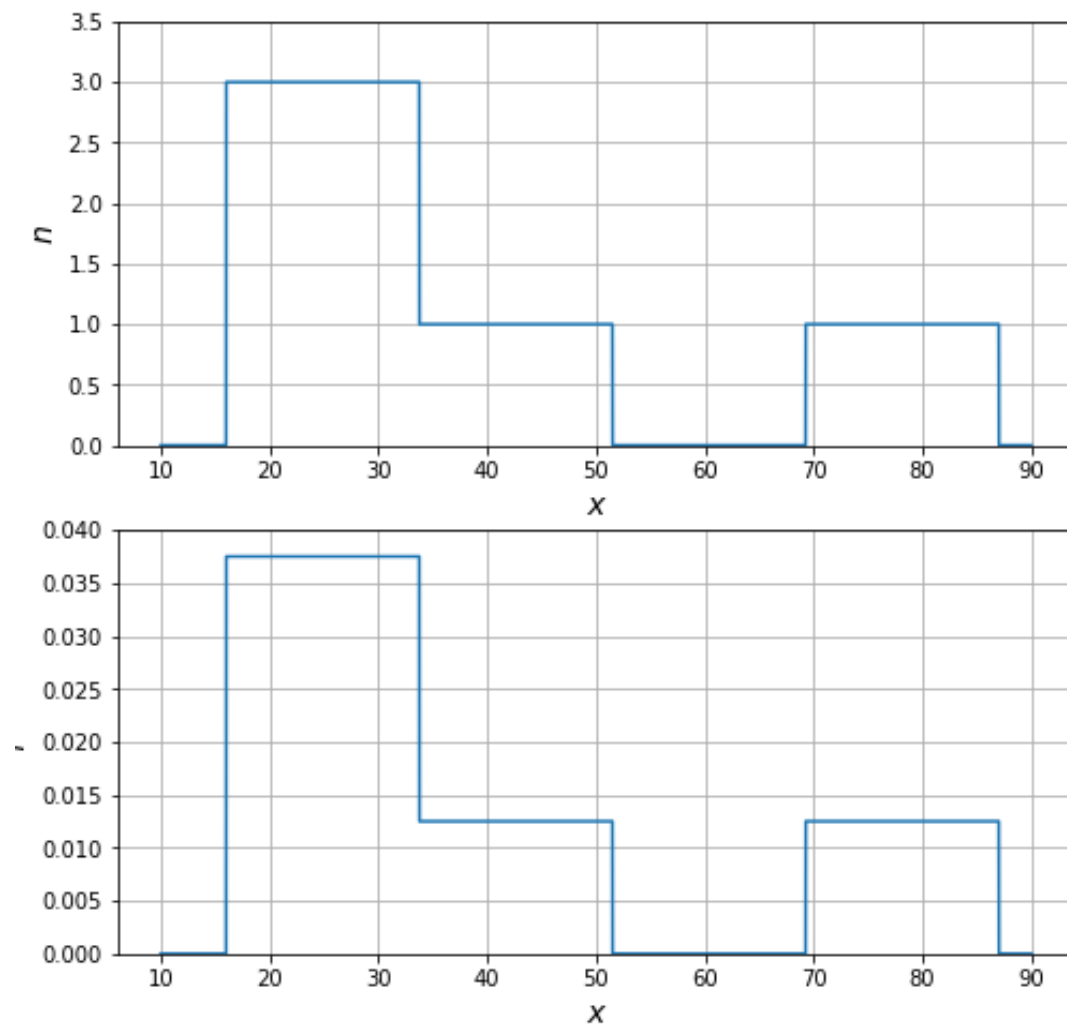
Формула для случая нормирования величины нормированного коэффициента

$$N\Delta_x = 5 * \frac{71}{4} = \frac{355}{4}$$

Программирование полученных формул приводит к следующему скрипту.

Скрипт hist1D_run3.py

```
import matplotlib.pyplot as plt
import numpy as np
import heaviside as step
nbins = 4
data = np.array([56, 45, 32, 16, 43, 33, 44, 55])
fig = plt.gcf()
fig.set_size_inches(8,8)
x1 = 10; x2 = 90;
m = 10**4;
x = np.linspace(x1,x2,m);
plt.subplot(211)
n = (3*step.heavyside(x,16) - 3*step.heavyside(x,135/4) +
step.heavyside(x,135/4) - step.heavyside(x,103/2) +
step.heavyside(x,277/4) - step.heavyside(x,87))
plt.plot(x,n);
plt.ylim([0, 3.5])
plt.xlabel('$x$', fontsize=14, color='black')
plt.ylabel('$n$', fontsize=14, color='black')
plt.grid(True)
plt.subplot(212)
N = len(data)
xmin = np.min(data);
xmax = np.max(data)
lbin = (xmax-xmin)/nbins
f = n/N/lbin
plt.plot(x,f)
plt.ylim([0, 0.04])
plt.xlabel('$x$', fontsize=14, color='black')
plt.ylabel('$f$', fontsize=14, color='black')
plt.grid(True)
```



Гистограммы данных $\text{data} = [32, 45, 16, 32, 87, 15, 25, 45]$, полученные по их аналитическим формам.

Эмпирическое распределение вероятностей

Частный случай гистограмм представляют распределения случайных величин, которые могут принимать не более, чем счетное число m , значений $a_n = 1, m$. В этом случае столбчатую диаграмму

$$p(a_i) = (p_i), \text{ где } \sum_{i=1}^{\infty} P_i = 1$$

называют *дискретным распределением вероятностей*, а случайную величину *дискретной*.

Задача 9.

. Дискретная случайная величина, которая может принимать значения 1, 2, 3, 4, 5, в восьми опытах 4 раза приняла значение 1, 1 раз значение 2, 2 раза значение 3 и 1 раз значение 5 как это прописано в массиве

`data = [1,1,2,1,3,3,1,5]`.

Написать модуль для программного подсчёта числа значений (частот) n_i принятых случайной величиной. Используя графическую функцию `bar`, построить график эмпирического распределения числа значений n_i и график эмпирического распределения вероятностей p_i .

Модуль `epmf.py`

```
def epmf(data):
    import numpy as np
    N = len(data)
    xmin = min(data); xmax = max(data)
    unique_data = np.unique(data)
    unique_data = np.sort(unique_data)
    unique_n = len(unique_data)
    n = np.zeros(unique_n, int)
    for k in range(unique_n):
        n[k] = data.count(unique_data[k])
    p = n/N
    return n, p, xmin, xmax, N, unique_data
```

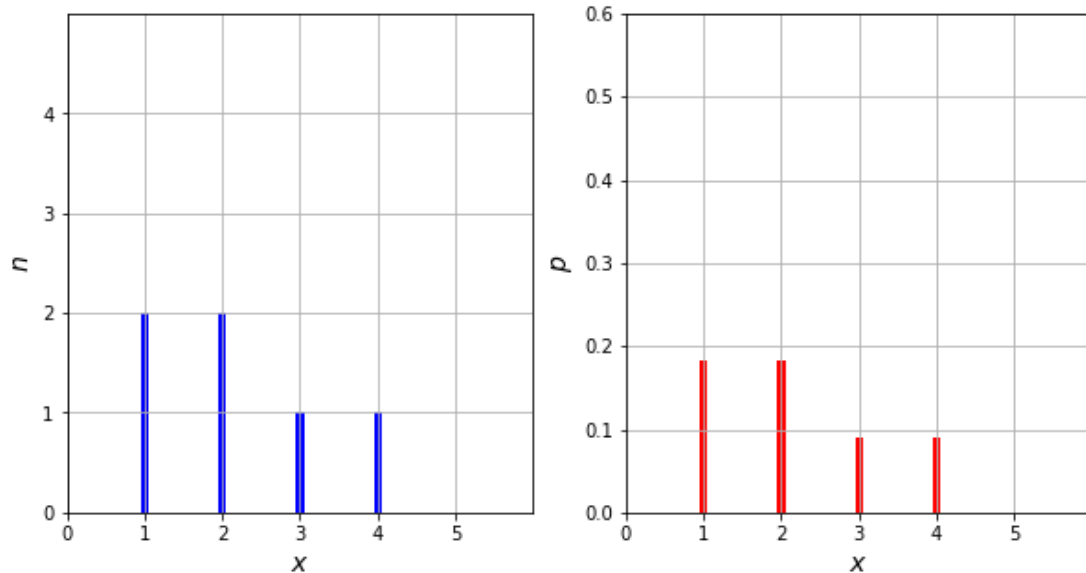
Скрипт epmf_run.py

```
import numpy as np
import matplotlib.pyplot as plt
import epmf as pmf
data=[1,1,2,2,4,13,28,51,54,28,3]
n,p,xmin,xmax,N,unique_data = pmf.epmf(data)
print('unique_data =',unique_data)
print('n =',n)
round_p = np.round(p*1e2)/1e2
print('round_p =',round_p)
sum_p = np.sum(p); print('sum_p =', sum_p)
fig = plt.gcf()
fig.set_size_inches(10,5)
width = 0.1
plt.subplot(121)
plt.bar(unique_data, n, width, color='blue')
plt.xlabel('$x$', fontsize=14)
plt.xticks(np.arange(0,6,1))
plt.ylabel('$n$', fontsize=14)
plt.yticks(np.arange(0,5,1))
plt.axis([0,6,0,5])
plt.grid(True)
plt.subplot(122)
plt.bar(unique_data,p,width,color='red')
plt.xlabel('$x$', fontsize=14)
plt.xticks(np.arange(0,6,1))
plt.ylabel('$p$', fontsize=14)
plt.axis([0,6,0,0.6])
plt.grid(True)
```

Исполнение скрипта `erfmf_run.py` приводит к следующим численным результатам и графикам

IPython console:

```
unique_data = [ 1  2  3  4 13 28 51 54]
n = [2 2 1 1 1 2 1 1]
round_p = [0.18 0.18 0.09 0.09 0.09 0.18 0.09 0.09]
sum_p = 1.0
```



Кумулятивное распределение дискретной случайной величины определяется формулой и, следовательно, программное построение его графика одинаково как для непрерывных случайных величин, так и для дискретных.

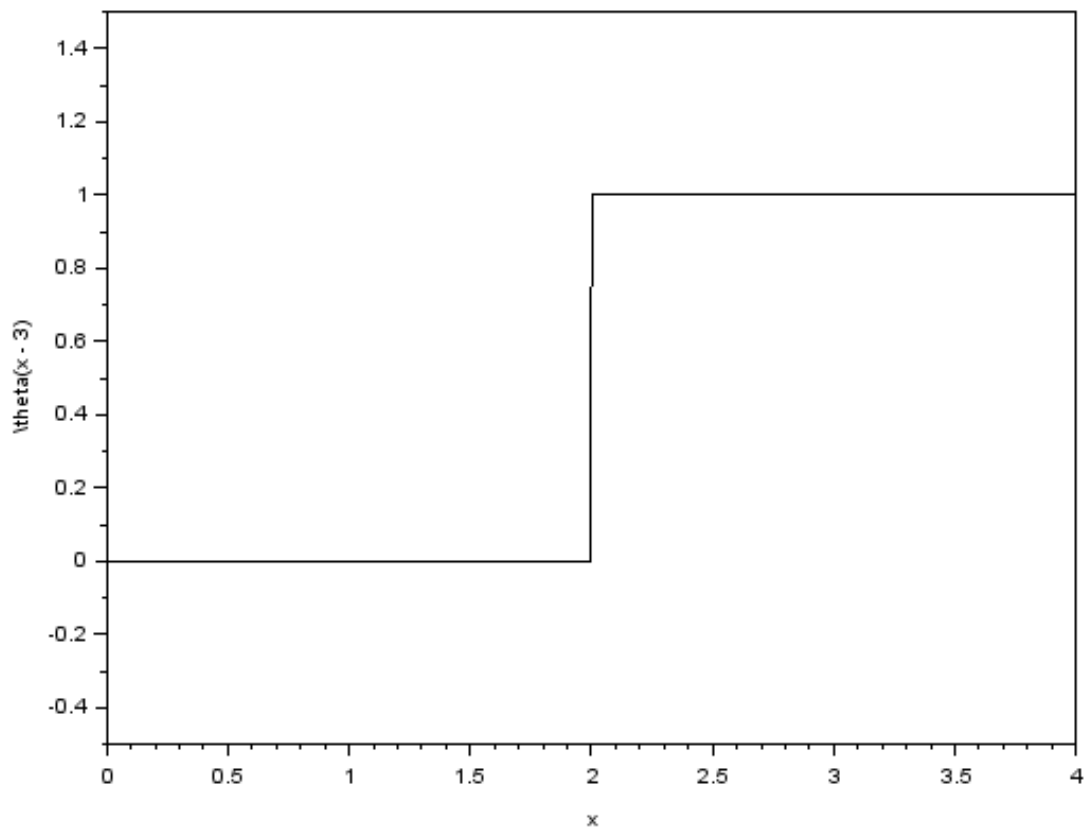
Графическое отображение случайных данных на Scilab

Эмпирическое кумулятивное распределение

Задача 7.

Построить в среде Scilab программную функцию heaviside, основываясь на алгебраическом определении функции Хевисайда по 2 формуле

```
function theta=heaviside(xi)
theta = abs(xi)./xi/2+1/2;
endfunction
n = 1e4;x = linspace(0,8,n);
a = 2; theta = heaviside(x - a)
plot2d(x,theta,rect = [0 -0.5 4 1.5])
ylabel('\theta(x - 3)'), xlabel('x')
a = gca();a.box = "on"
```



Функция Хевисайда при $x=2$

По графику, что функция Хевисайда представляет собой ступеньку единичной высоты, расположенную в точке с координатами $x=2$ оси абсцисс

Задача 9

Построить в двух горизонтально расположенных подокнах график случайных данных $\text{data} = [9\ 5\ 3\ 3\ 5\ 6\ 2\ 2\ 5\ 6\ 4\ 4\ 8]$ и их кумулятивное распределение.

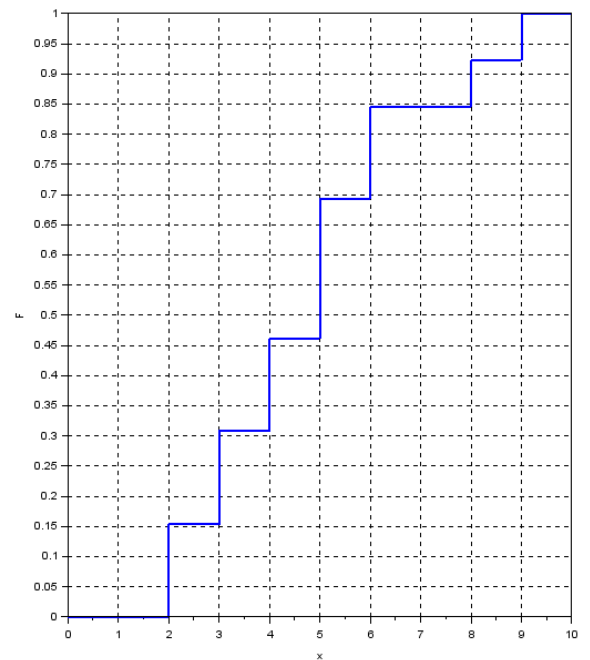
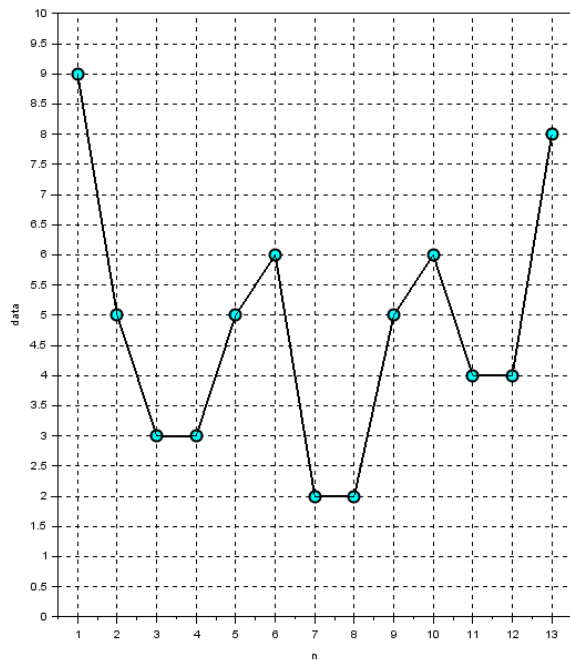


График данных и их кумулятивное распределение в задаче 9.

Код графика

```
f=get("current_figure");
f.figure_position = [70,30];
f.figure_size = [600,300];

data = [9 5 3 3 5 6 2 2 5 6 4 4 8]; N = length(data);
subplot(1,2,1) // График случайных данных
plot(1:length(data),data,'k','LineWidth',2, ...
'Marker','o','MarkerSize',10,'MarkerFaceColor','c')
plot2d(1:length(data),data,rect = [0.5 0 N+0.5 10])
xgrid, ylabel('data'), xlabel('n'),
subplot(1,2,2) // График эмпирического кумулятивного распределения
m = 1e4; x = linspace(0,10,m);
sumtheta = zeros(1,m);
for k = 1:N
sumtheta = sumtheta + heaviside(x - data(k));
end
F = sumtheta/N; plot(x,F,'LineWidth',2)
xgrid, ylabel('F'), xlabel('x')
```

задача № 10

Найти аналитический вид функции эмпирического кумулятивного распределения чисел $\text{data} = [13, 40, 23, 83, 40, 86]$ заказов в интернет - магазине за 7 дней торгов. Построить график кумуляты по ее явному виду и, основываясь на программном коде решения предыдущей задачи, дать графическую иллюстрацию числам заказов и проверить решение.

```
f=get("current_figure");
f.figure_position = [70,40];
f.figure_size = [500,300];
data = [13, 40, 23, 83, 40, 86];
N = length(data);
subplot(1,2,1)
plot(1:N,data,'k','LineWidth',2, ...
'Marker','o','MarkerSize',10,'MarkerFaceColor','c')
plot2d(1:length(data),data,rect = [0.5 0 N+0.5 100])
xgrid,ylabel('data'), xlabel('n'),
subplot(1,2,2)
m = 1e5;x = linspace(0,100,m);
sumtheta = zeros(1,m); // Численное суммирование кумулятивной функции for k = 1:N
sumtheta = sumtheta + heaviside(x - data(k)); end F = sumtheta/N;
plot(x,F,'g','LineWidth',4) // Явный вид кумулятивной функции
F = heaviside(x - 16)/5 + (2*heaviside(x - 32))/5 + ...
heaviside(x - 45)/5 + heaviside(x - 87)/5;
plot(x,F,'k','LineWidth',1);
xgrid, ylabel('F'), xlabel('x')
```

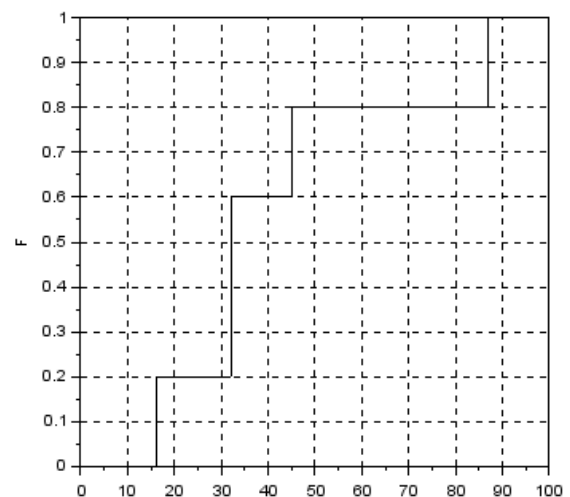
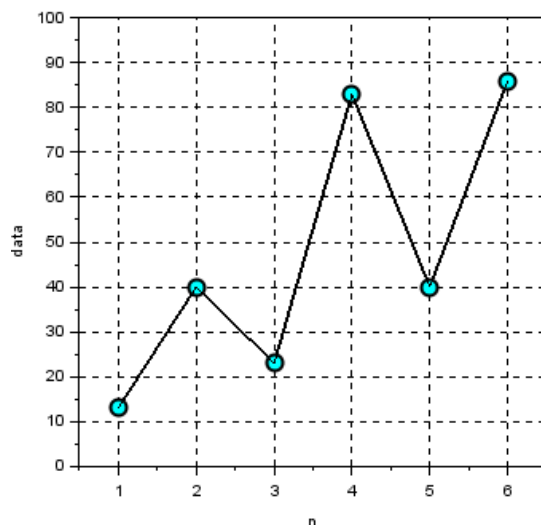


График данных и их кумулятивное распределение

Высокоуровневые реализации эмпирических распределений

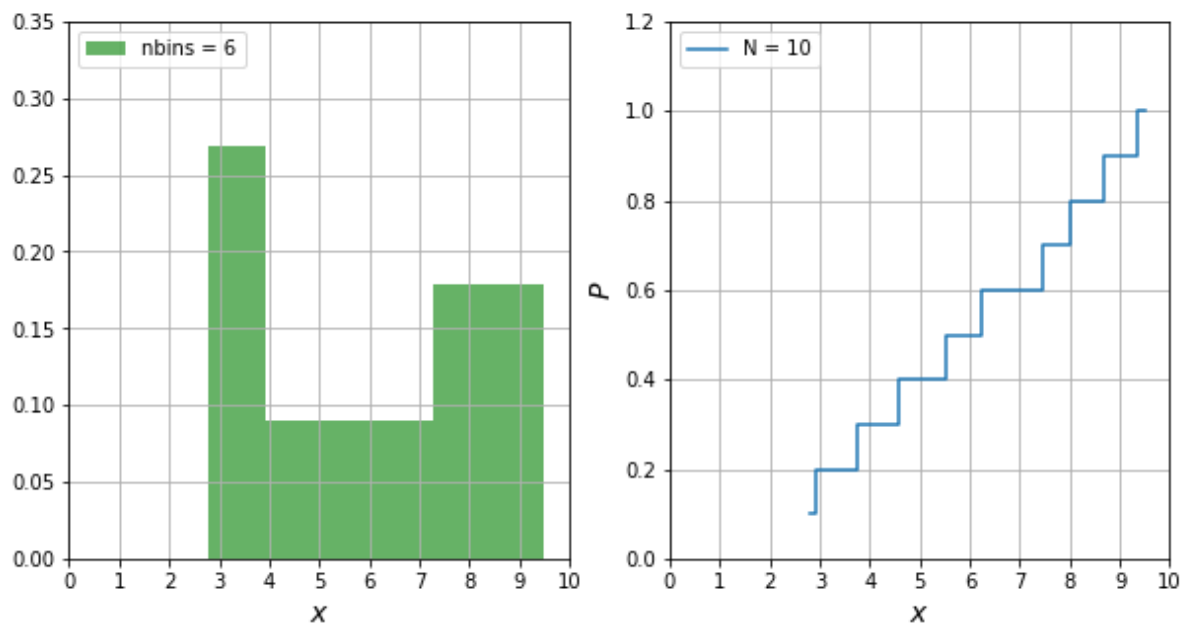
Высокоуровневые средства Python

Задача 10. Используя высокоуровневые средства языка Python, построить гистограмму и график функции эмпирического кумулятивного распределения массива случайных данных
`data = [1.5, 5.2, 8.6, 3.0, 5.6, 6.3, 2.7, 9.5, 5.8, 6.8]`

Скрипт `python_hist_ecdf.py`

```
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
nbins = 6
data =
np.array([7.5, 8.1, 4.6, 3.0, 5.6, 6.3, 2.8, 9.5, 8.8, 3.8])
N = len(data)
fig = plt.gcf()
fig.set_size_inches(10, 5)
# Гистограмма
plt.subplot(121)
plt.hist(data, nbins, normed=True,
facecolor='green', alpha=0.6, label='nbins =
'+str(nbins))
plt.legend( loc='upper left' )
plt.xlabel('$x$', fontsize=14)
plt.ylabel('$H$', fontsize=14)
plt.axis([0, 10, 0, 0.35])
plt.xticks(np.arange(0, N+1, 1))
plt.grid(True)
# Функция эмпирического распределения
plt.subplot(122)
ecdf = sm.distributions.ECDF(data)
x = np.linspace(min(data), max(data))
F = ecdf(x)
plt.step(x, F, label='N = '+str(N))
plt.legend(loc='upper left')
plt.xlabel('$x$', fontsize=14)
plt.ylabel('$P$', fontsize=14)
plt.axis([0, 10, 0, 1.2])
plt.xticks(np.arange(0, N+1, 1))
```

```
plt.grid(True)
```



Гистограмма и функция эмпирического кумулятивного распределения массива случайных данных
data = [7.5,8.1,4.6,3.0,5.6,6.3,2.8,9.5,8.8,3.8]

Непрерывные распределения

Равномерное распределение

Задача №11

Визуализировать $N = 10^{**2}$, 10^{**4} , 10^{**6} равномерно распределённых случайных чисел с параметрами $a = 30$ и $b = 30$. Для всех значений N построить гистограммы с соответствующим числом бинов $nbins = 10, 20, 30$ и графики функций эмпирических кумулятивных распределений

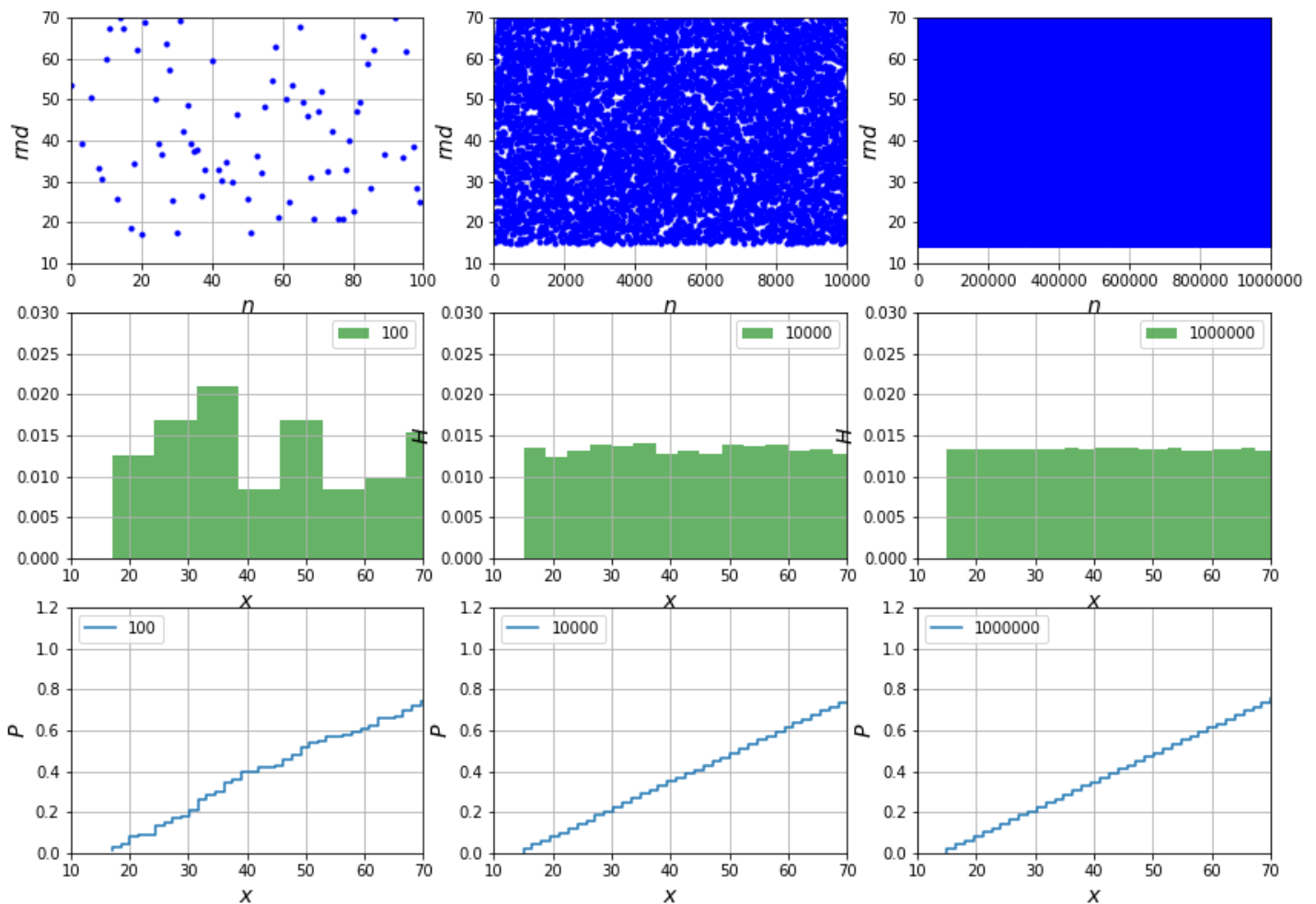
Скрипт uniform_rnd.py

```
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
a = 15; b = 90;
fig = plt.gcf()
fig.set_size_inches(14,10)
for k in range(1,4):
    N = 100**k
    rnd = np.random.uniform(a,b,N)
    # Визуализация случайных чисел
    plt.subplot(3,3,k)
    n = np.arange(0,N,1)
    plt.plot(n,rnd, 'b.' )
    plt.axis([0, N, 10 ,70])
    plt.xlabel('$n$',fontsize=14),
    plt.ylabel('$rnd$', fontsize=14)
    plt.grid(True)
    # Гистограмма
    plt.subplot(3,3,k+3)
    nbins = 10*k
    plt.hist(rnd, nbins, normed=True, facecolor='green',
    alpha=0.6,label=str(N))
    plt.legend( loc='upper right')
    plt.xlabel('$x$',fontsize=14)
    plt.ylabel('$H$',fontsize=14)
    plt.axis([10, 70, 0, 0.03])
    plt.grid(True)
    # Функция эмпирического кумулятивного распределения
    plt.subplot(3,3,k+6)
    ecdf = sm.distributions.ECDF(rnd)
```

```

x = np.linspace(min(rnd), max(rnd))
F = ecdf(x)
plt.step(x,F,label=str(N))
plt.legend(loc='upper left' )
plt.xlabel('$x$',fontsize=14)
plt.ylabel('$P$',fontsize=14)
plt.axis([10, 70, 0, 1.2])
plt.grid(True)

```



Равномерно распределенные случайные числа.

Равномерное распределение

Задача №12

Визуализировать $N=10^{**2}, 10^{**4}, 10^{**6}$ нормально распределенных случайных чисел с параметрами $\vartheta=2$ и $\delta = 0.5$. Для всех значений построить гистограммы с соответствующим числом бинов и графики функций эмпирических кумулятивных распределений.

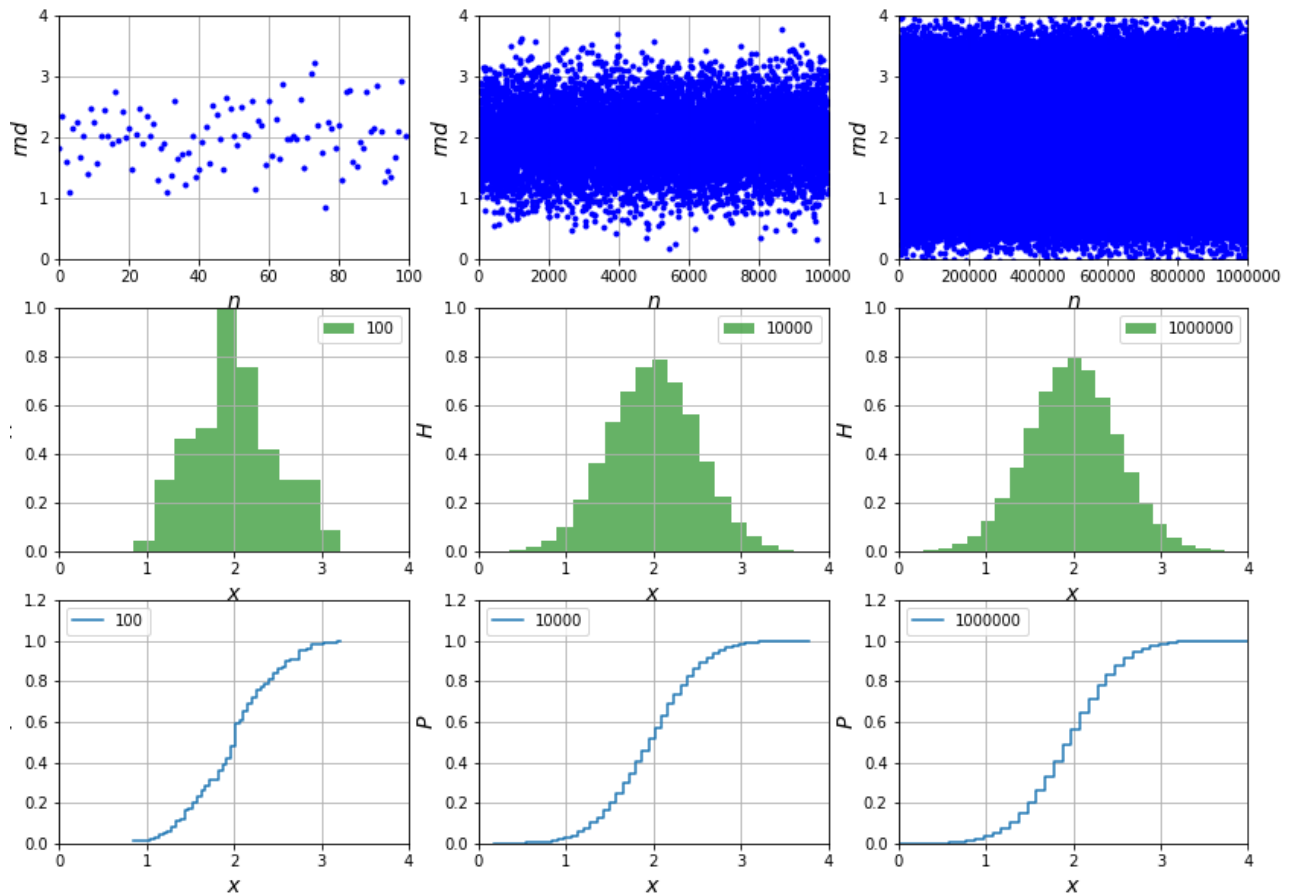
Скрипт normal_rnd.py

```
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
mu = 2; sigma = 0.5
fig = plt.gcf()
fig.set_size_inches(14,10)
for k in range(1,4):
    N = 100**k
    rnd = np.random.normal(mu,sigma,N)
    # Визуализация случайных чисел
    plt.subplot(3,3,k)
    n = np.arange(0,N,1)
    plt.plot(n,rnd, 'b.' )
    plt.axis([0, N, mu-4*sigma, mu+4*sigma])
    plt.xlabel('$n$', fontsize=14),
    plt.ylabel('$rnd$', fontsize=14)
    plt.grid(True)
    # Гистограмма
    plt.subplot(3,3,k+3)
    #оуровневые реализации эмпирических распределений
    nbins = 10*k
    plt.hist(rnd, nbins, normed=True,
facecolor='green',
            alpha=0.6, label=str(N))
    plt.legend( loc='upper right' )
    plt.xlabel('$x$', fontsize=14)
    plt.ylabel('$H$', fontsize=14)
    plt.axis([mu-4*sigma, mu+4*sigma, 0, 1.0])
    plt.grid(True)
    # Функция эмпирического кумулятивного распределения
    plt.subplot(3,3,k+6)
```

```

ecdf = sm.distributions.ECDF(rnd)
x = np.linspace(min(rnd), max(rnd))
F = ecdf(x)
plt.step(x,F,label=str(N))
plt.legend(loc='upper left' )
plt.xlabel('$x$',fontsize=14)
plt.ylabel('$P$',fontsize=14)
plt.axis([mu-4*sigma, mu+4*sigma, 0, 1.2])
plt.grid(True)

```



Нормально распределённые случайные числа.

Литература и интернет источники

https://ru.wikipedia.org/wiki/Визуализация_данных

<https://studfiles.net/preview/1392752/page:8/>

https://ru.wikipedia.org/wiki/Эмпирическая_формула

[https://ru.wikipedia.org/wiki/Гистограмма_\(статистика\)](https://ru.wikipedia.org/wiki/Гистограмма_(статистика))

<https://economy-ru.info/info/22211/>

https://ru.wikipedia.org/wiki/Распределение_вероятностей

http://wikiredia.ru/wiki/Кумулятивная_функция_распределения

<https://riptutorial.com/ru/r/example/24824/empirical-cumulative-distribution-function>

https://ru.wikipedia.org/wiki/Выборочная_функция_распределения

