

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ 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 №2

Группа: ФН11-53Б

Вариант №15

Студент: Пунегов Д.Е.

Преподаватель: Облакова Т.В.

Москва 2022

Задача 2. Моделирование и обработка выборки из дискретного закона распределения.**Задание.**

1. Для данного n смоделируйте выборку из биномиального закона распределения: $P(\xi = j) = p_j = C_k^j p^j (1 - p)^{k-j}, j = \overline{0, k}$.
2. Для полученной выборки постройте статистический ряд. Найдите эмпирическую функцию распределения $\hat{F}_n(x)$. Постройте на одном рисунке графики $F(x)$ и $\hat{F}_n(x)$. Вычислите статистику Колмогорова.
3. Вычислите выборочное среднее и выборочную дисперсию и сравните с истинными значениями этих характеристик.

Данные задачи 2

15	12	0,5	170
----	----	-----	-----

Моделирование и обработка выборки из дискретного закона распределения.

1. Для данного n смоделируйте выборку из биномиального закона распределения

1.1 Загружаем все нужные библиотеки

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from IPython.display import Markdown as md
from IPython.display import display, Math, Latex
import math

pd.set_option('display.max_rows', 20)
```

1.2 Задание начальных данных

```
In [2]: k = 12
p = 0.5
n = 170

def partial_sums(iterable):
    total = 0
    for i in iterable:
        total += i
    yield total
```

1.3 Генерация выборки

```
[3]: import random
from collections import Counter

x = []
pk = [round(math.comb(k, j) * p ** j * (1 - p) ** (k - j), 8) for j in range(k + 1)]
kum = list(round(i, 8) for i in partial_sums(pk))
print('Вероятность: {}'.format(pk))
print('Накопленная вероятность: {}'.format(kum))

rand_list = []

counter = []

for i in range(n):
    ran = random.uniform(0, 1)
    rand_list.append(ran)
    if ran < kum[0]:
        counter.append(0)
    for j in range(k - 1):
        if kum[j] <= ran and kum[j + 1] > ran:
            counter.append(j + 1)
            break
    if (kum[k - 1] <= ran and kum[k] >= ran):
        counter.append(k)

num_set = pd.DataFrame()
num_set['Случайные числа'] = rand_list
num_set
```

Вероятность: [0.00024414, 0.00292969, 0.01611328, 0.05371094, 0.12084961, 0.19335938, 0.22558594, 0.19335938, 0.12084961, 0.05371094, 0.01611328, 0.00292969, 0.00024414]
Накопленная вероятность: [0.00024414, 0.00317383, 0.01928711, 0.07299805, 0.19384766, 0.38720704, 0.61279298, 0.80615236, 0.92700197, 0.98071291, 0.99682619, 0.99975588, 1.00000002]

```
[3]:
```

	Случайные числа
0	0.913611
1	0.106358
2	0.934840
3	0.808262
4	0.532953
...	—
165	0.310419
166	0.056566
167	0.753357
168	0.261515
169	0.576931

170 rows × 1 columns

```
[4]: df = pd.DataFrame()
num_collection = pd.DataFrame({'Выборка': counter})
df['Значения СВ'] = list(range(k + 1))
counter = dict(Counter(counter))
for i in range(k + 1):
    if i not in counter:
        counter[i] = 0
quantities = [pair[1] for pair in sorted(counter.items())]
if len(quantities) < k + 1:
    quantities.insert(0, 0)
df['Частоты'] = quantities
df['Относительные частоты'] = df['Частоты'] / n
df['Накопленные частоты'] = list(partial_sums(df['Относительные частоты']))
num_collection
```

```
[4]:
```

Выборка	
0	8
1	4
2	9
3	8
4	6
...	...
165	5
166	3
167	7
168	5
169	6

170 rows × 1 columns

```
[5]: df
```

```
[5]:
```

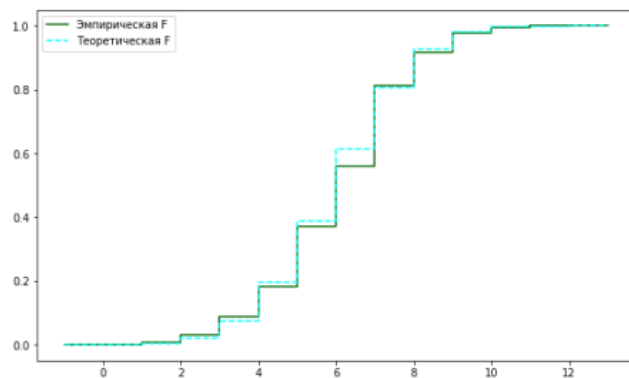
	Значения СВ	Частоты	Относительные частоты	Накопленные частоты
0	0	0	0.000000	0.000000
1	1	1	0.005882	0.005882
2	2	4	0.023529	0.029412
3	3	10	0.058824	0.088235
4	4	16	0.094118	0.182353
5	5	32	0.188235	0.370588
6	6	32	0.188235	0.558824
7	7	43	0.252941	0.811765

8	8	18	0.105882	0.917647
9	9	10	0.058824	0.976471
10	10	3	0.017647	0.994118
11	11	1	0.005882	1.000000
12	12	0	0.000000	1.000000

2. Эмпирическая и теоретическая функция распределения

```
[6]: plt.figure(figsize=(10,6))
plt.step([-1, 0, *(df['Значения СВ'] + 1)], [0, 0, *df['Накопленные частоты']], color = 'darkgreen', label='Эмпирическая F')
plt.step([-1, 0, *(df['Значения СВ'] + 1)], [0, 0] + kum, color = 'cyan', label='Теоретическая F', linestyle='dashed')
plt.legend(loc="upper left")
```

```
[6]: <matplotlib.legend.Legend at 0x1910096c2b0>
```



3. Статистика Колмогорова

```
[7]: md('$\delta = \sup(|F(z) - \overset{\wedge}{F}_{170}(z)|) = {}$'.format('{{{}}}'.format(n), max(abs(df['Накопленные частоты'] - kum))))
```

```
[7]: 
$$\delta = \sup(|F(z) - \overset{\wedge}{F}_{170}(z)|) = 0.053969450588235324$$

```

4. Вывод

Полученная статистика Колмогорова достаточно маленькая, что говорит нам о том, что полученная нами выборка взята из нашего исходного закона распределения.