*Федеральное государственное бюджетное образовательное учреждение* *высшего образования*

|  |  |
| --- | --- |
| **Изображение выглядит как эмблема, герб, нашивка, символ  Автоматически созданное описание** | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(национальный исследовательский университет)***  ***(МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** ФУНДАМЕНТАЛЬНЫЕ НАУКИ\_\_\_\_\_\_\_\_

**КАФЕДРА** ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА И МАТЕМАТИЧЕСКАЯ ФИЗИКА (ФН11)\_

**НАПРАВЛЕНИЕ ПОДГОТОВКИ** МАТЕМАТИКА И КОМПЬЮТЕРНЫЕ НАУКИ (02.03.01)

**Отчет**

**по лабораторной работе № \_2\_**

**Название лабораторной работы: Моделирование и обработка выборки из дискретного закона распределения.**

**Вариант № 2**

**Дисциплина:** Теория вероятности и математическая статистика

Студент группы ФН11-52Б **\_\_\_\_\_\_\_\_\_\_\_\_\_ Кожемякин Г.А.**

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_ Облакова Т.В.**

(Подпись, дата) (И.О. Фамилия)

Москва, 2024

**Задание**

1. Для данного смоделируйте выборку из биномиального закона распределения: .

2. Для полученной выборки постройте статистический ряд. Найдите эмпирическую функцию распределения Постройте на одном рисунке графики и . Вычислите статистику Колмогорова.

3. Вычислите выборочное среднее и выборочную дисперсию и сравните с истинными значениями этих характеристик.

**Исходные данные варианта**

Количество испытаний k = 10

Вероятность успеха в одном испытании p = 0,4

Объём выборки n = 120

**Выполнение задания**

Работа выполнена при помощи языка программирования Python с использованием библиотек для анализа данных: Numpy, Matplotlib.

Для заданных значений k, p, n смоделируем выборку из биноминального распределения .

Находим теоретический закон распределения по формуле Бернулли:

Теоретический закон распределения:

|  |  |
| --- | --- |
| Значение случайной величины | Вероятность |
| 0 | 0.006047 |
| 1 | 0.040311 |
| 2 | 0.120932 |
| 3 | 0.214991 |
| 4 | 0.250823 |
| 5 | 0.200658 |
| 6 | 0.111477 |
| 7 | 0.042467 |
| 8 | 0.010617 |
| 9 | 0.001573 |
| 10 | 0.000105 |

Полученный вероятностный вектор:

p = (0.006047, 0.040311, 0.120932, 0.214991, 0.250823, 0.200658, 0.111477, 0.042467, 0.010617, 0.001573, 0.000105)

Кумулятивные вероятности:

u = (0.00605, 0.04636, 0.16729, 0.38228, 0.6331, 0.83376, 0.94524, 0.98771, 0.99832, 0.9999, 1.0)

Моделирование вектора Y из n случайных величин:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 0.561700 | 0.656707 | 0.202685 | 0.919441 | 0.091606 | … |

YT =

По вектору Y разыгрываем вектор X в соответствии с алгоритмом:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 3 | 6 | 2 | … |

XT =

Построение статистического ряда на основании разыгранного вектора X:

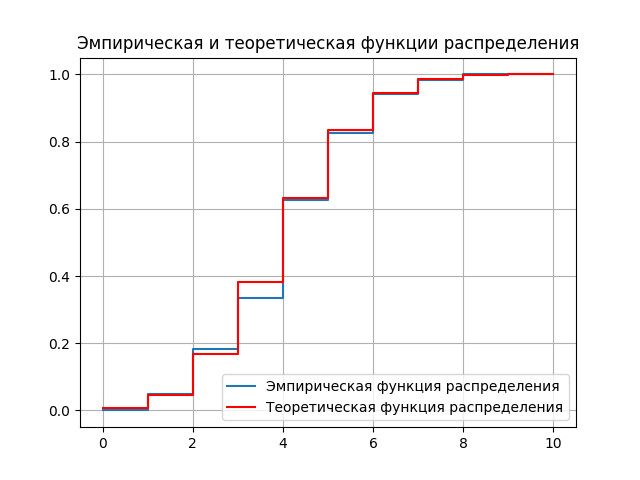
|  |  |  |  |
| --- | --- | --- | --- |
| Значение СВ | Частоты | Относительные частоты | Накопленные частоты |
| 0 | 0 | 0 | 0 |
| 1 | 6 | 0.05 | 0.05 |
| 2 | 16 | 0.133 | 0.183 |
| 3 | 18 | 0.15 | 0.333 |
| 4 | 35 | 0.292 | 0.625 |
| 5 | 24 | 0.2 | 0.825 |
| 6 | 14 | 0.117 | 0.942 |
| 7 | 5 | 0.042 | 0.983 |
| 8 | 2 | 0.017 | 1 |
| 9 | 0 | 0 | 1 |
| 10 | 0 | 0 | 1 |

Вектор накопленных частот содержит ненулевые значения эмпирической функции распределения. Для вычисления статистики Колмогорова данные удобно свести в таблицу. Статистика Колмогорова вычисляется по формуле:

Вычисление статистики Колмогорова для выборки из дискретного закона.

|  |  |  |  |
| --- | --- | --- | --- |
| Интервал | Эмпирическая ф-я распределения | Теоретическая ф-я распределения | Модуль разности |
| (-∞,0] | 0 | 0 | 0 |
| (0,1] | 0 | 0.00605 | 0.00605 |
| (1,2] | 0.05 | 0.04636 | 0.00364 |
| (2,3] | 0.183 | 0.16729 | 0.01571 |
| (3,4] | 0.333 | 0.38228 | 0.04928 |
| (4,5] | 0.625 | 0.6331 | 0.0081 |
| (5,6] | 0.825 | 0.83376 | 0.00876 |
| (6,7] | 0.942 | 0.94524 | 0.00324 |
| (7,8] | 0.983 | 0.98771 | 0.00471 |
| (8,9] | 1 | 0.99832 | 0.00168 |
| (9,10] | 1 | 0.9999 | 0.0001 |
| (10,+ ∞) | 1 | 1 | 0 |

Из данных таблицы следует, что максимальное различие теоретической и эмпирической функции распределения наблюдается на полуинтервале (3,4]. Значение статистики Колмогорова будет равно 0.04928. Значение статистики невелико, что говорит о приемлемом результате моделирования. Снизу приведены совмещенные графики эмпирической и теоретической функций распределения.



Вычисление эмпирических и теоретических характеристик распределений.

Выборочное среднее:

Выборочная дисперсия:

Математической ожидание:

Дисперсия: 2,4

На основании полученных результатов можно сделать вывод, что, так как абсолютная величина разности математического ожидания и выборочного среднего мала (0,05833), а отношение выборочной дисперсии к ее теоретическому значению близко к единице, то результаты моделирования можно считать удовлетворительными.

**Выводы:** в ходе проделанной лабораторной работы было проведено моделирование и обработка выборки из дискретного закона распределения. Для полученной выборки построен статистический ряд и эмпирическая функция распределения, вычислена статистика Колмогорова. На основании значений выборочного среднего и выборочной дисперсии был сделан вывод о степени качества моделирующей дискретный закон выборки.

**Приложение**

В приложении представлен программный код на языке Python, с помощью которого была проведена работа над задачей лабораторной работы.

**import** math  
**import** random  
**import** matplotlib.pyplot **as** plt  
**import** numpy **as** np  
  
  
**def** c\_j\_k(j, k):  
 **return** math.factorial(k) / (math.factorial(j) \* math.factorial(k - j))  
  
  
**def** check\_value(cumul, y):  
 X = []  
 **for** item **in** y:  
 tmp = 0  
 **for** prob **in** cumul:  
 **if** item < prob:  
 X.append(tmp)  
 **break** tmp += 1  
 **return** X  
  
  
*# Количество испытаний*k = 10  
*# Вероятность успеха в одном испытании*p = 0.4  
*# Объем выборки*n = 120  
  
*# Находим теоретический закон по формуле Бернулли*prob = []  
**for** j **in** range(11):  
 prob.append(c\_j\_k(j, k) \* (p \*\* j) \* ((1 - p) \*\* (k - j)))  
  
print(**'Теоретический закон распределения:'**)  
print(**'Значение СВ - Вероятность'**)  
**for** i **in** range(len(prob)):  
 print(**f'{**i**}: {**prob[i]**}'**)  
  
*# Кумулятивные вероятности*u = [prob[0]]  
**for** i **in** range(1, 11):  
 tmp = u[i - 1] + prob[i]  
 u.append(tmp)  
  
print(**'Кумулятивные вероятности'**)  
u\_str = **f'u = ('  
for** item **in** u:  
 **if** item != 1:  
 u\_str += **f'{**round(item, 5)**}, '  
 else**:  
 u\_str += **f'{**round(item, 5)**})'**print(u\_str)  
  
*# Моделируем вектор из n случайных чисел*y = []  
**for** i **in** range(n):  
 y.append(random.random())  
  
print(**'Вектор y:'**)  
print(y)  
  
*# По вектору y разыгрываем вектор X:*X = check\_value(u, y)  
print(**'Вектор x:'**)  
print(X)  
  
*# Построение статистического ряда*count = []  
**for** i **in** range(11):  
 count.append(X.count(i))  
  
freq = [count[i] / n **for** i **in** range(11)]  
add\_freq = [freq[0]]  
**for** i **in** range(1, 11):  
 tmp = add\_freq[i - 1] + freq[i]  
 add\_freq.append(tmp)  
  
print(**'Значения СВ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |'**)  
print(**f'Частоты | {**count[0]**} | {**count[1]**} | {**count[2]**} | {**count[3]**} | {**count[4]**} | {**count[5]**} | {**count[6]**} | '  
 f'{**count[7]**} | {**count[8]**} | {**count[9]**} | {**count[10]**} |'**)  
print(**f'Относительные частоты | {**freq[0]**} | {**freq[1]**} | {**freq[2]**} | {**freq[3]**} | {**freq[4]**} | {**freq[5]**} | {**freq[6]**} '  
 f'| {**freq[7]**} | {**freq[8]**} | {**freq[9]**} | {**freq[10]**} |'**)  
print(**f'Накопленные частоты |{**add\_freq[0]**}|{**add\_freq[1]**}|{**add\_freq[2]**}|{**add\_freq[3]**}|{**add\_freq[4]**}|{**add\_freq[5]**}|'  
 f'{**add\_freq[6]**}|{**add\_freq[7]**}|{**add\_freq[8]**}|{**add\_freq[9]**}|{**add\_freq[10]**}|'**)  
  
*# По накопленным частотам строим эмпирическую функцию распределения*x = np.arange(len(add\_freq))  
  
*# Построение эмпирической функции распределения*plt.step(x, add\_freq, label=**'Эмпирическая функция распределения'**, where=**'post'**)  
  
*# Построение биноминального распределения*plt.step(x, u, label=**'Теоретическая функция распределения'**, color=**'red'**, where=**'post'**)  
  
*# Настройка графика*plt.title(**'Эмпирическая и теоретическая функции распределения'**)  
plt.legend()  
plt.grid(**True**)  
  
*# Отображение графика*plt.show()  
  
*# Вычисление статистики Колмогорова*delta = 0  
tmp = 0  
**for** i **in** range(11):  
 tmp = abs(u[i] - add\_freq[i])  
 **if** tmp > delta:  
 delta = tmp  
  
print(**f'Статистика Колмогорова: {**delta**}'**)  
  
*# Выборочные характеристики, сравнение с истинными значениями*x\_mean = sum(X) / n  
s = sum([(item - x\_mean) \*\* 2 **for** item **in** X]) / (n - 1)  
  
print(**f'Выборочное среднее: {**x\_mean**}'**)  
print(**f'Выборочная дисперсия: {**s**}'**)  
  
m = sum([prob[i] \* i **for** i **in** range(len(prob))])  
d = p \* (1 - p) \* k  
  
print(**f'Математическое ожидание: {**m**}'**)  
print(**f'Дисперсия: {**d**}'**)