|  |  |  |  |
| --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА  ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  **Кафедра програмних систем і технологій**    Дисципліна  **«Ймовірнісні основи програмної інженерії»**  **Лабораторна робота № 5**  **Дискретні розподіли ймовірностей** | | | |
| **Виконав:** | Пономаренко Андрій Сергійович | **Перевірила**: | Вечерковська Анастасія Сергіївна |
| Група | ІПЗ-25мс | Дата перевірки |  |
| Форма навчання | денна | Оцінка |  |
| Спеціальність | 121 |
| 2022 | | | |

**Мета роботи:** навчитись використовувати на практиці набуті знання про центральні тенденції та міри.

**Завдання:**

1. Ймовірність знаходження в кожному прибулому потязі вагонів на дане призначення 0,2. Визначити ймовірність того, що в трьох із п’яти потягів, які прибувають протягом однієї години, будуть вагони на дане призначення.

2. Знайти ймовірність того, що в п’яти незалежних випробуваннях подія А відбудеться: а) рівно 4 рази; б) не менше 4 разів, якщо в кожному випробуванні ймовірність появи події становить 0,8.

3. На кондитерській фабриці 20% всіх цукерок складають льодяники. Знайти ймовірність того, що серед 400 вибраних навмання цукерок буде рівно 80 льодяників.

4. На автомобільному заводі у звичному режимі роботи з конвеєра сходить 100000 автомобілів. Ймовірність бракованого автомобіля дорівнює 0,0001. Знайти ймовірність того, що з конвеєра зійшло 5 бракованих автомобілів.

5. Ймовірність того, що пара взуття, яка взята навмання з виготовленої партії виявиться вищого ґатунку дорівнює 0,4. Чому дорівнює ймовірність того, що серед 600 пар, які поступили на контроль, виявиться від 228 до 252 пар взуття вищого ґатунку?

6. Банк обслуговує 100 клієнтів, від кожного з яких може надійти вимога на проведення фінансової операції на наступний день з ймовірністю 0,4. Знайти найімовірніше число вимог клієнтів кожного дня, та його ймовірність.

7. Завод випускає в середньому 4% нестандартних виробів. Яка ймовірність того, що число нестандартних виробів у партії з 4000 штук не більше 170?.

8. Яка ймовірність того, що при 10000 незалежних киданнях монети герб випаде 5000 разів?

9. Фірма відправила на базу 1000 якісних виробів. Ймовірність того, що вироби в дорозі пошкодяться дорівнює 0,002. Знайти ймовірність того, що на базу прибуде 5 пошкоджених виробів.

10. Нехай ймовірність того, що грошовий приймальник автомату при опусканні монети скидає неправильно дорівнює 0,03. Знайти найімовірніше число випадків правильної роботи автомату, якщо буде кинуто 150 монет.

**Рішення:**

Це все є задачі на біноміали тому всі вони вирішуються досить однаково

n-кількість спроб

x-потрібна кількість

p-шанс кожної

Але в задачах з 3 по 10 дуже великі числа щоб рахувати їх факторіал тому ми використаємо формулу наближеного біноміалу

1.

2.

3.

4.

Для задач 5 та 7 нам треба обчислити занадто багато наближених шансів і для цього використовується формула

a-роздільник

b-другий роздільник

x-потрібна кількість

p-шанс кожної

5.

===0.83147-0.14917=0.6823

6.

max=n\*p

Найймовірніше число 100\*0.4=40.

7.

8.

9.

10.

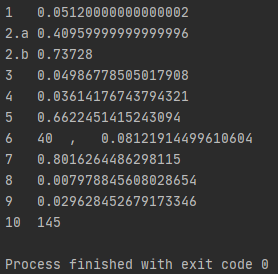
max=n\*p

150\*(1-0.03)= 150\*0.97=145.5

**Код програми:**

from numpy import \*  
  
def cusFact(n):  
 if(n<=1):  
 return 1  
 else:  
 return (n\* cusFact(n-1))  
  
  
def binomial(n, p, x, mode):  
 chances=zeros(n+1)  
 if(x<=n):  
 if(n<150): #при цифровій обробці поріг обчислюємих факторіалів вище ніж в ручну  
 for i in range(n+1):  
 chances[i]=(cusFact(n))\*(p\*\*i)\*((1-p)\*\*(n-i))/(cusFact(n-i)\*cusFact(i)) #звичайний біноміал  
 else:  
 for i in range(n+1):  
 chances[i]=1.0/sqrt(2\*pi\*n\*p\*(1-p)) \* exp(-(i-n\*p)\*\*2/(2\*n\*p\*(1-p))) #біноміал через нормальний розподіл  
 if(mode == '>x'):  
 return sum(chances[(x+1):n+1]) #виводить всі шанси вище x  
 if(mode == '<x'):  
 return sum(chances[0:x]) #виводить всі шанси низще x  
 if(mode == '=x'):  
 return chances[x] #виводить тільки шанс x  
 if(mode == 'max'):  
 return argmax(chances) #виводить індекс найвищого шансу і ігнорує x,можна булоб просто помножити n на p але всеодно ця таблиця уже зроблена  
 if(mode == 'min'):  
 return argmin(chances) #виводить індекс найменшого шансу і ігнорує x  
 else:  
 return 0 #якщо x більше ніж кількість спроб то шанс 0  
#main  
  
print("1\t"+str(binomial(5,0.2,3,'=x')))  
print("2.a\t"+str(binomial(5,0.8,4,'=x')))  
print("2.b\t"+str(binomial(5,0.8,4,'=x')+(binomial(5,0.8,4,'>x'))))  
print("3\t"+str(binomial(400,0.2,80,'=x')))  
print("4\t"+str(binomial(100000,0.0001,5,'=x')))  
print("5\t"+str(binomial(600,0.4,228,'>x')-binomial(600,0.4,251,'>x')))  
print("6\t"+str(binomial(100,0.4,0,'max'))+"\t,\t"+str(binomial(100,0.4,binomial(100,0.4,0,'max'),'=x')))  
print("7\t"+str(binomial(4000,0.04,171,'<x')))  
print("8\t"+str(binomial(10000,0.5,5000,'=x')))  
print("9\t"+str(binomial(1000,0.002,5,'=x')))  
print("10\t"+str(binomial(150,(1-0.03),0,'max')))

**Результат виконання програми**



**Висновок:**

Я вирішив 10 задач на біноміальний розподіл та автоматизував їх за допомогою python3.8.

Нам довелось використати наближення нормального розподілу тому не вийшло отримати ідеальну точність,також як завжди через те як python обчислює десятинні числа інколи губляться дуже глибокі біти і виходить результат по типу тих що в 1 та 2а.В 5тій та 7 задачі результат не співпадає більше всього швидше за все через різні методи знаходження величин та те що таблиця z score взагалі не дуже точна. Але в основному всі результати більш менш співпадають