

In the name of God



استاد : دکتر تیموری

دانشجو : توحید حقیقی سیس

شماره دانشجویی : 830598021

موضوع : تمرین دهم

## تمرین اول :

1. According to the Mendelian theory of genetics, a certain garden pea plant should produce white, pink, or red flowers, with respective probabilities  $\frac{1}{4}$ ,  $\frac{1}{2}$ ,  $\frac{1}{4}$ . To test this theory a sample of 564 peas was studied with the result that 141 produced white, 291 produced pink, and 132 produced red flowers. Approximate the  $p$ -value of this data set

268

### 11 Statistical Validation Techniques

- (a) by using the chi-square approximation, and
- (b) by using a simulation.

این سوال 2 قسمت دارد در قسمت اول گفته با روش و کتابخانه chi-square مقدار احتمال را محاسبه کنیم و در قسمت دوم گفته که با روش شبیه سازی یعنی با تکرار این روش در تعداد خیلی بالا مقدار جواب اصلی و نزدیک را محاسبه میکنیم .

```
N_Array = [141,291,132]
probability = [0.25,0.5,0.25]
Sum_n=564

T = 0
for i in range(len(N_Array)):
    T += ((N_Array[i]-Sum_n*probability[i])**2)/(Sum_n*probability[i])

P_Value = 1-chi_Fuction.cdf(T,2)
print("p-value :",P_Value)
```

کد بالا جواب قسمت 1 است و با کتابخانه scipy مقدار مورد نظر را محاسبه میکند .

```

# در این تابع ما شروع به ساخت اعداد رندوم میکنیم
# و بعد این اعداد را با شمارنده های هر قسمت شمارش میکنیم
def CounterGenerator(random_count):
    # در روی این سوال 3 مقدار داده شده است
    randN = [0,0,0]
    # به تعداد عدد ورودی عدد رندوم تولید میکنیم
    for item in range(random_count):
        # ساخت عدد رندوم
        x = random.random()
        # شروط رو بر اساس اعدادی که روی سوال داده اعمال میکنیم
        if x < 0.25:
            randN[0]+=1
        elif x < 0.75:
            randN[1]+=1
        else:
            randN[2]+=1

    return randN

#تابع شبیه سازی به صورت زیر است#

```

کد بالا برای تولید اعداد رندوم در احتمالات بالا است.

```

#تابع شبیه سازی به صورت زیر است#
def simulation(pt,Rep,probability):
    for i in range(0, Rep):
        U=[]
        U=CounterGenerator(564)
        Tt = 0
        # print(U)
        for o in range(len(U)):
            Tt += ((U[o] - 564 * probability[o]) ** 2) / (564 * probability[o])
        # print(Tt)
        if Tt > T:
            pt = pt + 1

    pt = pt / Rep
    print('p-value using simulationand 1000 Rep: ',pt)

```

تابع شبیه سازی نیز به صورت کد زیر می باشد

تمرین دوم :

2. To ascertain whether a certain die was fair, 1000 rolls of the die were recorded, with the result that the numbers of times the die landed  $i, i = 1, 2, 3, 4, 5, 6$  were, respectively, 158, 172, 164, 181, 160, 165. Approximate the  $p$ -value of the test that the die was fair

- (a) by using the chi-square approximation, and
- (b) by using a simulation.

محاسبه این روش نیز مثل سوال اول است با این تفاوت که ارایه 6 تایی دارد و با احتمالات متفاوت .

```
N_Array = [158,172,164,181,160,165]
probability = [1/6,1/6,1/6,1/6,1/6,1/6]
Sum_n=1000

T = 0
for i in range(len(N_Array)):
    T += ((N_Array[i]-Sum_n*probability[i])**2)/(Sum_n*probability[i])

P_Value = 1-chi_Fuction.cdf(T,2)
print("p-value :",P_Value)

# simulation function
simulation(0,1000,probability)
# print(T)
```

کد بالا جواب قسمت 1 است و با کتابخانه spicy مقدار مورد نظر را محاسبه میکند .

```

# در این تابع ما شروع به ساخت اعداد رندوم میکنیم
# و بعد این اعداد را با شمارنده های هر قسمت شمارش میکنیم
def CounterGenerator(random_count):
    # در روی این سوال 3 مقدار داده شده است
    randN = [0,0,0,0,0,0]
    # به تعداد عدد ورودی عدد رندوم تولید میکنیم
    for item in range(random_count):
        # ساخت عدد رندوم
        x_rand = random.random()
        # شرط رو بر اساس اعدادی که روی سوال داده اعمال میکنیم
        for k in range(6):
            if x_rand < (k+1)/6:
                randN[k] += 1
                break

    return randN

# تابع شبیه سازی به صورت زیر است
def simulation(pt, Rep, probability):
    for i in range(0, Rep):
        U = []
        U = CounterGenerator(564)
        Tt = 0
        # print(U)
        for o in range(len(U)):
            Tt += ((U[o] - 564 * probability[o]) ** 2) / (564 * probability[o])
        # print(Tt)
        if Tt > T:
            pt = pt + 1

```

تمرین سوم :

3. Approximate the  $p$ -value of the hypothesis that the following 10 values are random numbers: 0.12, 0.18, 0.06, 0.33, 0.72, 0.83, 0.36, 0.27, 0.77, 0.74.