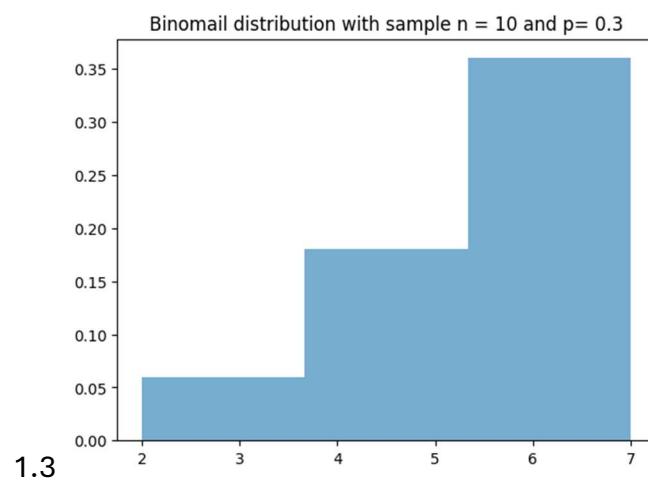
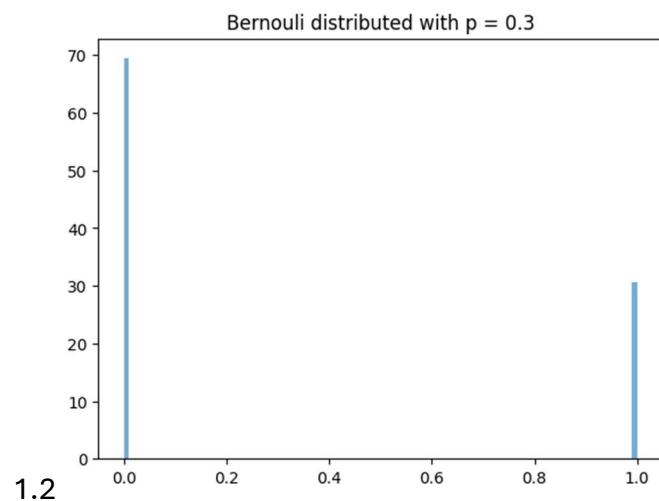
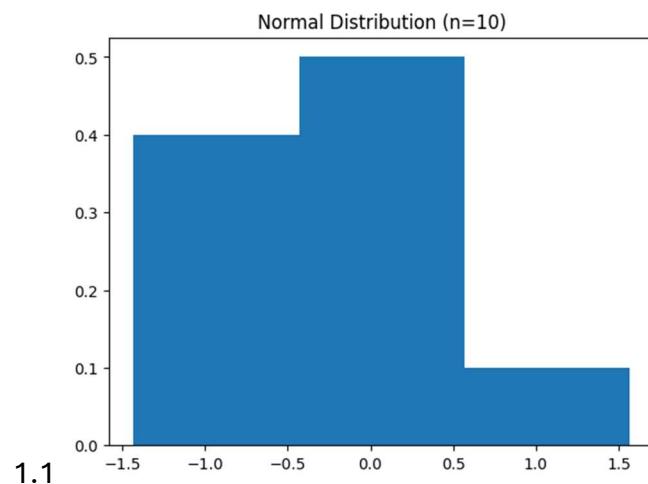
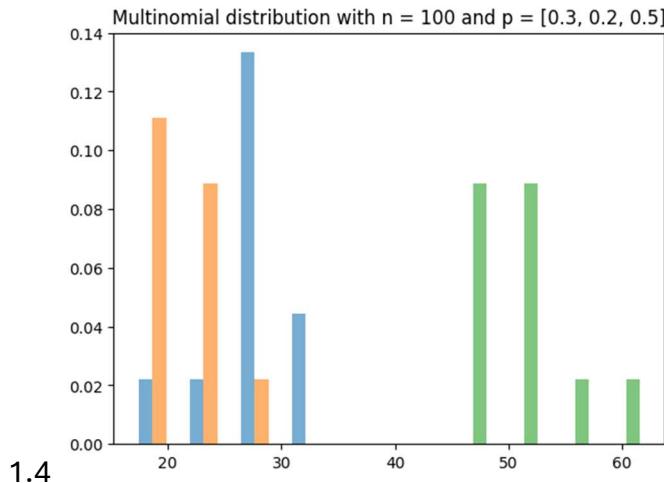
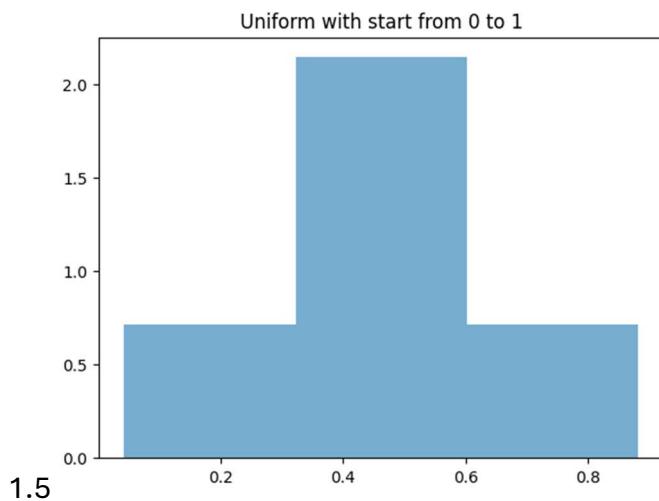


## Sampling 1.

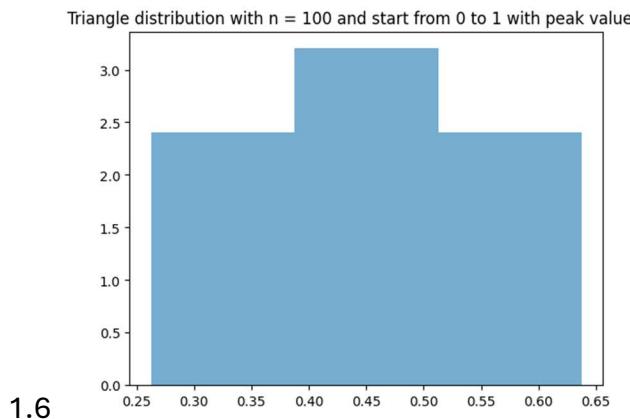




1.4

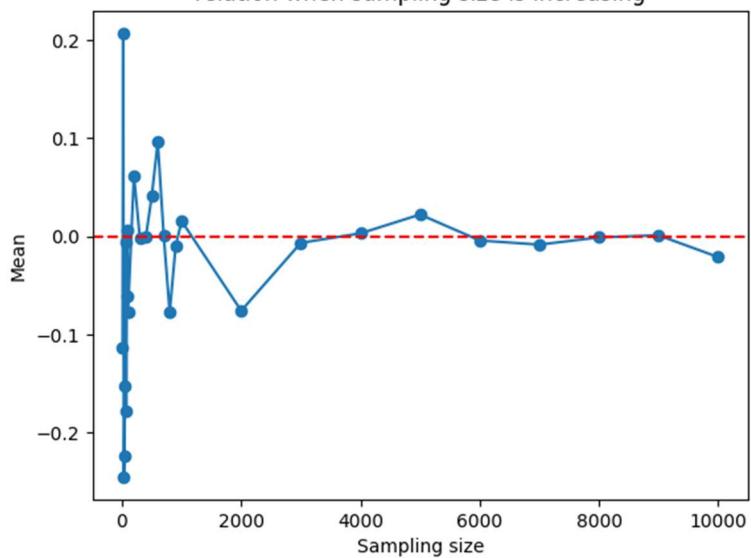


1.5



1.6

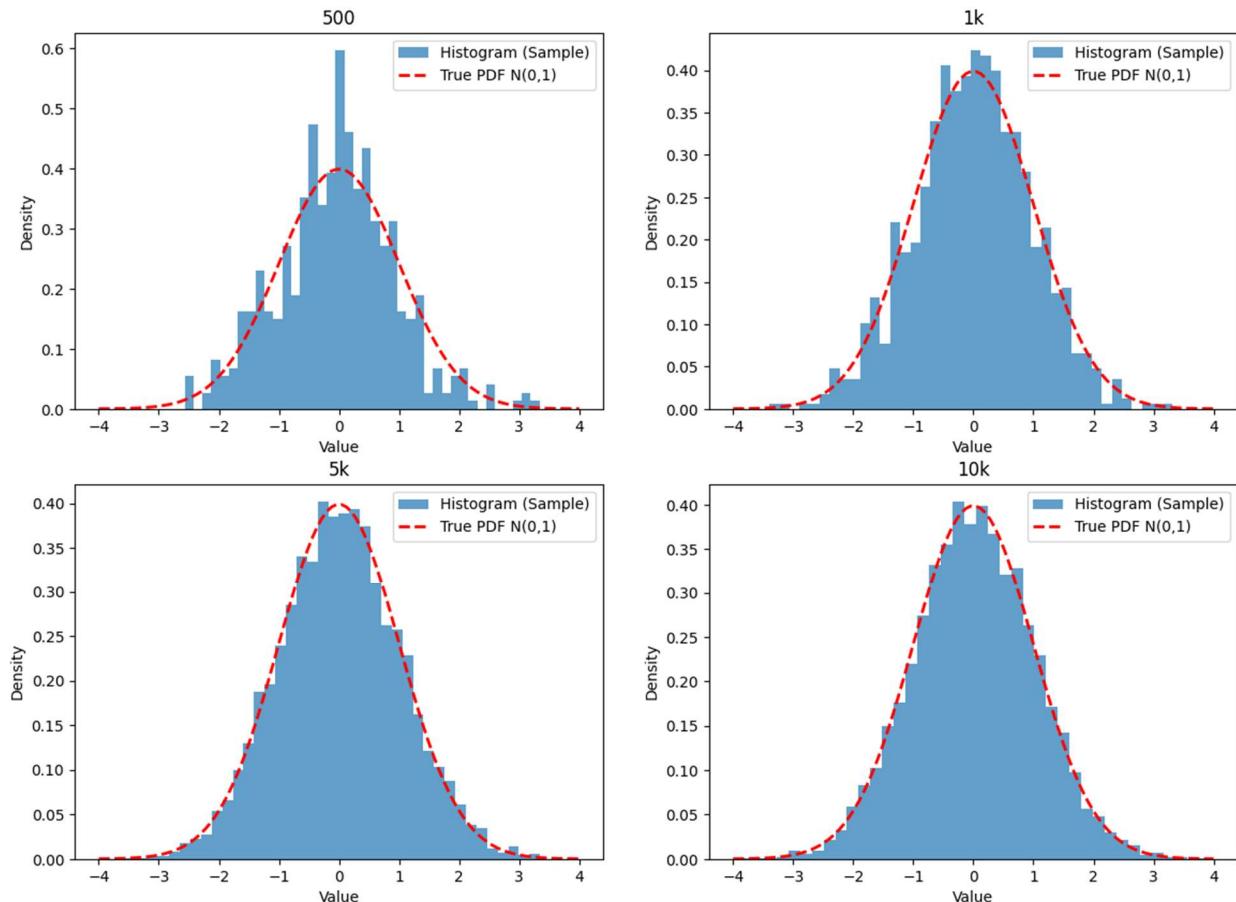
relation when sampling size is increasing



2.

3.

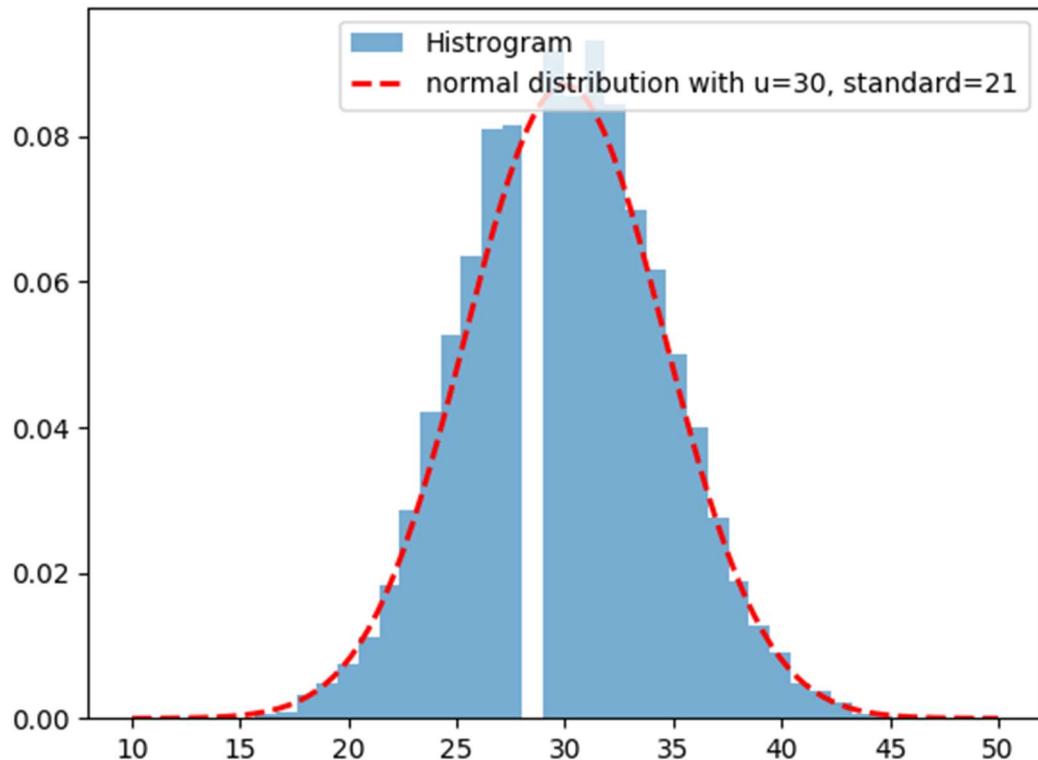
Histogram Approximation vs True PDF (bins=40)



#### 4.

```
n = 100
p = 0.3
# we will use sample_binomial to create distribution of tossed coin with p = 0.3 and 100
# assume sample_size considered high value because it easy to see what kind of distribution it is.
sample_tosses = sample_binomial(sample_size=10000,n=n, p=p)
# sample ในแต่ละครั้งก็คือจำนวนหัวที่เกิดขึ้นในแต่ละ n ทำให้เข้าสู่สูตร  $S_n = X_1 + X_2 + \dots + X_n$  ใน CLT
x_normal = np.linspace(10,50,100)
# mean of binomial distribution is  $n*p$  and standard derivation is  $\sqrt{np*(1-p)}$ 
y_normal = norm.pdf(x_normal, n*p, np.sqrt(n*p*(1-p)))
plt.hist(sample_tosses, bins=40, density=True, alpha=0.6, label="Histogram")
plt.plot(x_normal, y_normal, 'r--', linewidth=2, label="normal distribution with u=30, standard=21")
plt.legend()
plt.show()

# จากภาพจะเห็นว่า Histogram จะมีความเหมือน Normal distribution เป็นอย่างมาก
# โดยค่า P จะเปลี่ยนรูปแบบการกระจายของกราฟให้ค่าเข้ามาแต่เมื่อ n มากขึ้นจะทำให้
# ผลลัพธ์ของการนับ จะหายไป เพราะค่าจากทั้งสองฟังก์ชันมีค่านาก
```



5.

```
▶ # ข้อ 5
# เราจะใช้การเปิดตาราง Z เพราะว่าเราได้ว่ากราฟมีความเป็น normal distribution (ถ้าทำมือ)
# แต่ scipy.stats สามารถหา cdf จาก method cdf ได้ เช่น norm.cdf

u = n*p
s = round(np.sqrt(n*p*(1-p)),4)
z_value = (40.5-u)/s
# นิ้ 40.5 เพราะเรามองว่า Area ของ >= 41 คือเริ่มตั้งแต่ 40.5 เพราะแต่ละค่าที่เป็น distribut กว้าง 1
# print(z_value)
prob_from_left = norm.cdf(z_value)
prob_gte_41 = 1-prob_from_left
print(round(prob_gte_41,4))
```

→ 0.011

6.

```
# ข้อ 6
prob_le_40 = binom.cdf(k=40, n=n, p=p)
prob_gt_40 = 1 - prob_le_40
print(round(prob_gt_40,4))
# จะเห็นว่าค่าที่ได้ใกล้เคียงกับ CDF โดยใช้ normal distribution มากๆ
```

0.0125

7.

```
▶ from scipy.integrate import quad
def fx(x):
    if -2 <= x and x <= 0:
        return 0.1
    if 0 < x and x <= 2:
        return 0.4
    return 0
def fy(y):
    if y < 3 or y > 5:
        return 0
    return 1/2

def intergrate_z(x,z):
    return fx(x)*fy(z-x)

def fz(z):
    result = quad(func=intergrate_z, a=-np.inf,b=np.inf,args=(z,))
    # argument a, b จะป้อนเข้า intergratee ที่ parameter ตัวแรกโดยอัตโนมัติ และ args(z,) เป็นการบอกว่า parameter ตัวที่สอง ให้มีค่า z เข้าไป
    return result[0]
prob = quad(func=fz, a=3,b=5)
print(f"ความน่าจะเป็น P(3 < Z < 5) คือ: {prob[0]:.4f}")
# ใช้หลักการ sum of random variables ที่ใช้งาน Convolution ของสอง fx และ fy
# fx+fy(z) = (fx * fy)(z)
# ช่วงแรก z < 1 มีค่าน้อยกว่า 0 เพราะไม่ทับกัน
# ช่วงของ 1 <= z < 3
# ช่วงสาม 3 <= z < 5
# ช่วง 4 z>=5
```

→ ความน่าจะเป็น P(3 < Z < 5) คือ: 0.5000

8.

```
# ข้อ 8
# correlation = Cov(x,y)/sqrt(var(x)*var(y))
size = 1000
x = sample_uniform(sample_size=size, from_x=-1,to_x=1)
a1 = sample_uniform(sample_size=size, from_x=-1,to_x=1)
a2 = sample_uniform(sample_size=size, from_x=-10,to_x=10)
a3 = sample_uniform(sample_size=size, from_x=-100,to_x=100)

y1 = x + 10
correlation_1 = np.corrcoef(x,y1)
# print(correlation)
# [[pxx, pxy],[pyx,ppy]]
print(correlation_1[0,1])

y2 = x+a1
correlation_2 = np.corrcoef(x,y2)
print(correlation_2[0,1])

y3 = x+a2
correlation_3 = np.corrcoef(x,y3)
print(correlation_3[0,1])

y4 = x+a3
correlation_4 = np.corrcoef(x,y4)
print(correlation_4[0,1])
```

```
→ 0.9999999999999998
0.6912092074340993
0.1106737006832169
0.05643575482166692
```

9.

```
# ข้อ 9
print(f'9.1 ตอบ ใช่')

y5 = x + 10000
correlation_5 = np.corrcoef(x,y5)
print(f'correlation ของ x+ 10000: {correlation_5[0,1]}')
print(f'correlation ของ x+ U(-10,10): {correlation_3[0,1]}')

y6 = y5 + a2
correlation_6 = np.corrcoef(x,y6)
print(f'correlation ของ x+ U(-10,10) + 10000: {correlation_6[0,1]}')
print('จะเห็นว่า correlation ของ x+U(-10,10) กับ x+ U(-10,10) + 10000 แทนไม่ต่างกันเพราะค่าคงที่แทบจะไม่ส่งผลกับCorrelation')
print('อีกทั้งค่าคงที่ทำหน้าที่เพียงแค่ shift ค่าออกไปแต่ไม่ได้ส่งผลต่อ Variance และ Covariance ทำให้ correlation ไม่เปลี่ยนแปลงมาก')
```

```
→ 9.1 ตอบ ใช่
correlation ของ x+ 10000: 1.0
correlation ของ x+ U(-10,10): 0.1106737006832169
correlation ของ x+ U(-10,10) + 10000: 0.11067370068321104
จะเห็นว่า correlation ของ x+U(-10,10) กับ x+ U(-10,10) + 10000 แทนไม่ต่างกันเพราะค่าคงที่แทบจะไม่ส่งผลกับCorrelation
อีกทั้งค่าคงที่ทำหน้าที่เพียงแค่ shift ค่าออกไปแต่ไม่ได้ส่งผลต่อ Variance และ Covariance ทำให้ correlation ไม่เปลี่ยนแปลงมาก
```

## 10.1 และ 10.2.

```
▶ # ข้อ 1 ตอบ u = 15 เพราะ จะทำให้ t มีค่าเฉลี่ยเท่ากับ 15 ซึ่งทำให้ term แรกของ P มีค่าเป็น 0 ทำให้ P มีค่า 0.001  
print('ข้อ 1. ตอบ u = 15 เพราะ จะทำให้ t มีค่าเฉลี่ยเท่ากับ 15 ซึ่งทำให้ term แรกของ P มีค่าเป็น 0 ทำให้ P มีค่า 0.001')  
# ข้อ 2 หา expected value ที่จะ fail โดยใช้ intergrate(fn*gn)  
from scipy.integrate import quad  
def fn(x):  
    return ((0.97/2250)*(x-15)**2 + 0.001)*0.5  
  
p = quad(a=14,b=16,func=fn)  
print(f'ข้อ 2. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: {round(p[0],4)}')
```

⇒ ข้อ 1. ตอบ u = 15 เพราะ จะทำให้ t มีค่าเฉลี่ยเท่ากับ 15 ซึ่งทำให้ term แรกของ P มีค่าเป็น 0 ทำให้ P มีค่า 0.001  
ข้อ 2. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: 0.0011

## 10.3

```
▶ # ข้อ 3 จาก all disk failur at the same time ถึงจะ read fail จะได้ว่า p = p^N โดย n คือจำนวน disk ที่กำลังหาย  
# ให้ n = 10000 คือจำนวน request ที่เข้ามา  
idx = 1  
result = 0  
while(True):  
    pi = p[0]**idx  
    idx+=1  
    at_least_fail = binom.cdf(1, 10000,pi)  
    # argument แรกคือค่าที่จะสำเร็จ 1 ครั้งคือมี fail 1 ครั้ง  
    check = 1 - at_least_fail  
    if (check < 0.0001):  
        # < 0.01% = 0.0001  
        result = check  
        break  
print(f'จำนวน disk ที่ต้องใช้คิดเป็นจำนวน {idx}')  
print(f'ข้อ 3. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: {round(result,6)}')
```

⇒ จำนวน disk ที่ต้องใช้คิดเป็นจำนวน 3  
ข้อ 3. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: 8.5e-05

## 10.4-1 และ 10.4-2

```
▶ # ข้อ 4.  
# ข้อ 4.1 ให้ u = 15 เพราะ probability ที่ u = 15 มีค่าสูงที่สุด  
print('ข้อ 4.1. ตอบ u = 15 เพราะ จะทำให้ t มีค่าเฉลี่ยเท่ากับ 15 ซึ่งทำให้ term แรกของ P มีค่าเป็น 0 ทำให้ P มีค่า 0.001 ซึ่งหักออกส่วนที่ t = 15 มีค่าสูงที่สุดจาก normal distribution')  
# ข้อ 4.2  
x = np.linspace(10,30,100)  
# gn = norm.pdf(x,15,9)  
# plt.plot(x,gn, 'r--', linewidth=2)  
def gn(x):  
    return norm.pdf(x,15,3)  
def fn_1(x):  
    return ((0.97/2250)*(x-15)**2 + 0.001)*gn(x)  
p = quad(func=fn_1, a=10,b=30)  
print(f'ข้อ 4.2. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: {round(p[0],6)}')
```

⇒ ข้อ 4.1. ตอบ u = 15 เพราะ วุฒิทำให้ t มีค่าเฉลี่ยเท่ากับ 15 ซึ่งทำให้ term แรกของ P มีค่าเป็น 0 ทำให้ P มีค่า 0.001 ซึ่งหักออกส่วนที่ t = 15 มีค่าสูงที่สุดจาก normal distribution  
ข้อ 4.2. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: 0.004003

### 10.4-3

```
▶ # ข้อ 4.3
idx = 1
result = 0
while(True):
    pi = p[0]**idx
    idx+=1
    at_least_fail = binom.cdf(1,10000,pi)
    check = 1 - at_least_fail
    if (check < 0.0001):
        result = check
        break
print(f'ข้อ 4.3. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: {round(result,7)}')
print(f'จำนวน disk ที่ต้องใช้คือเป็นจำนวน {idx}'')
```

☞ ข้อ 4.3. ตอบ มีความน่าจะเป็นที่จะ Fail เท่ากับ: 2e-07  
จำนวน disk ที่ต้องใช้คือเป็นจำนวน 4

## 11.1

```
# จาก sigma ที่ให้มา (covariance matrix)
print(f'ข้อ 1 ตอบ จะเห็นว่า cov(b,a), cov(b,c), cov(b,d) มีค่าเท่ากัน 0 ทำให้ correlation เป็น 0 ด้วยท่าให้เป็นเหตุที่ Independent กันเห็นญื่น')
# ข้อ 1 ตอบ จะเห็นว่า cov(b,a), cov(b,c), cov(b,d) มีค่าเท่ากัน 0 ทำให้ correlation เป็น 0 ด้วยท่าให้เป็นเหตุที่ Independent กันเห็นญื่น
```

## 11.2

```
▶ days_choices = [30, 180]
answer = dict()
count = 10000
for day in days_choices:
    answer[day] = get_expected_return(day, count)
    print(f'จำนวนวัน={day}')
    for i in range(len(coin_name)):
        # นำอัปของเดือนที่รับยูของแต่ละช่วงเวลาที่กำหนด
        coin_return = answer[day][i]
        # นำค่าของแต่ละ coin และลงมาหาผลลัพธ์
        print(f'{coin_name[i]}, มีexpected_return {round(np.mean(coin_return),4)}, มีความน่าจะเป็นเท่ากับ: {round(get_profit_prob(coin_return),4)}')

☞ จำนวนวัน=30
a, มีexpected_return 0.8622, มีความน่าจะเป็นเท่ากับ: 0.4541
b, มีexpected_return 0.5727, มีความน่าจะเป็นเท่ากับ: 0.5148
c, มีexpected_return 1.2119, มีความน่าจะเป็นเท่ากับ: 0.4616
d, มีexpected_return 1.2603, มีความน่าจะเป็นเท่ากับ: 0.4421
จำนวนวัน=180
a, มีexpected_return 6.9187, มีความน่าจะเป็นเท่ากับ: 0.3995
b, มีexpected_return 4.4333, มีความน่าจะเป็นเท่ากับ: 0.5562
c, มีexpected_return 10.2743, มีความน่าจะเป็นเท่ากับ: 0.4037
d, มีexpected_return 10.377, มีความน่าจะเป็นเท่ากับ: 0.3568
```

## 11.3 และ 11.4

```
print('ข้อ 3 ตอบ เหรียญ b มี probability ในการได้กำไรมากที่สุด')
print('ข้อ 4 ตอบ เพราเม็คความน่าจะเป็นจะต่ำกว่า 0.5 แต่ยังค่าที่เราได้กลับมามากการลงทุน (เมื่อชนะได้ 1 ครั้งมีมูลค่าสูงมากจนทำให้ Expected return value เป็นมากได้')

# ข้อ 3 ตอบ เหรียญ b มี probability ในการได้กำไรมากที่สุด
# ข้อ 4 ตอบ เพราเม็คความน่าจะเป็นจะต่ำกว่า 0.5 แต่ยังค่าที่เราได้กลับมามากการลงทุน เมื่อชนะได้ 1 ครั้งมีมูลค่าสูงมากจนทำให้ Expected return value เป็นมากได้
```

## 11.5

### ขายหลังจาก 30 วัน

Strategy	Buy a	Buy b	Buy c	Buy d	Expected[return]	Variance[return]	Probability of having profit
1	100%	0%	0%	0%	0.8621995485655338	39.12259119722992	0.4541
2	0%	100%	0%	0%	0.5726676487150836	10.344334112191925	0.5148
3	0%	0%	100%	0%	1.2119389069946793	53.09450250167658	0.4616
4	0%	0%	0%	100%	1.260326763097639	72.81678658150535	0.4421
5	50%	50%	0%	0%	0.7174335986403086	12.306824623890858	0.5149
6	50%	0%	50%	0%	1.0370692277801066	30.60093470757088	0.4911
7	50%	0%	0%	50%	1.0612631558315866	37.54445050073845	0.4751

### ขายหลังจาก 180 วัน

Strategy	Buy a	Buy b	Buy c	Buy d	Expected[return]	Variance[return]	Probability of having profit
1	100%	0%	0%	0%	6.91871839169123	1194.6231569384317	0.3995
2	0%	100%	0%	0%	4.43329679272481	146.19787448389314	0.5562
3	0%	0%	100%	0%	10.274338152137293	2415.5765247381496	0.4037
4	0%	0%	0%	100%	10.376991155276293	2885.3938212818084	0.3568
5	50%	50%	0%	0%	5.67600759220802	336.1190057658521	0.5541
6	50%	0%	50%	0%	8.596528271914261	1028.5312623442328	0.4695
7	50%	0%	0%	50%	8.647854773483763	1211.5234495111822	0.4354

## 11.6,11.7, 11.8 และ 11.9

▶ ข้อ 6 ตอบ แผนที่ 4 ให้ผล expected\_return ที่ดีสุดเมื่อขายเหรียญ 30 วัน และ 180 วัน  
 print('ข้อ 6 ตอบ แผนที่ 4 ให้ผล expected\_return ที่ดีสุดเมื่อขายเหรียญ 30 วัน และ 180 วัน')  
 print('ข้อ 7 ตอบ แผนที่ 5 ให้ probability สูงสุดเมื่อขายหลัง 30 วันแต่ แผน 2 ให้ probability ต่ำสุดเมื่อขายหลังจาก 180 วัน')  
 print(''ข้อ 8 ตอบ เมื่อจงหา var(x,y) = var(x) + var(y) + 2\*cov(x,y) ทำให้ยัง cov มากยิ่งกว่าให้ var โดยจาก table ที่โจทย์กำหนดให้ จะเห็นว่า Cov(a,d) มีค่านากกว่า Cov(a,c) ทำให้ Var ของแผนที่ 7 มากกว่าแผนที่ 6'')  
 print(''ข้อ 9 ตอบ ใช้งาน Broker ถ้าคุณไม่เก่งในก็ลงทุนสิ่งที่มีความเสี่ยงต่ำ แต่ถ้าอยากรวยไว้ก็ลงทุนแบบ All in'')

◀ ข้อ 6 ตอบ แผนที่ 4 ให้ผล expected\_return ที่ดีสุดเมื่อขายเหรียญ 30 วัน และ 180 วัน  
 ข้อ 7 ตอบ แผนที่ 5 ให้ probability สูงสุดเมื่อขายหลัง 30 วันแต่ แผน 2 ให้ probability ต่ำสุดเมื่อขายหลังจาก 180 วัน  
 ข้อ 8 ตอบ เมื่อจงหา var(x,y) = var(x) + var(y) + 2\*cov(x,y) ทำให้ยัง cov มากยิ่งกว่าให้ var โดยจาก table ที่โจทย์กำหนดให้ จะเห็นว่า Cov(a,d) มีค่านากกว่า Cov(a,c) ทำให้ Var ของแผนที่ 7 มากกว่าแผนที่ 6  
 ข้อ 9 ตอบ ใช้งาน Broker ถ้าคุณไม่เก่งในก็ลงทุนสิ่งที่มีความเสี่ยงต่ำ แต่ถ้าอยากรวยไว้ก็ลงทุนแบบ All in

+ ตอบ + ข้อความ