1.

1. (1) P(Type 1 error)

$= P(\text{Reject } H_0 / H_0 \text{ is true})$

$= P(\sum_{i=1}^{5} X_i = 5 / H_0 : p = \frac{1}{2})$

$= \binom{5}{5}\left(\frac{1}{2}\right)^5\left(\frac{1}{2}\right)^0 = \frac{1}{32}$ #

(2) P(Type 2 error)

$= P(\text{Accept } H_0 / H_a \text{ is true})$

$= P(\sum_{i=1}^{5} X_i < 5 / H_a : p = \frac{3}{4})$

$= \sum_{k=0}^{4} \binom{5}{k}\left(\frac{3}{4}\right)^k\left(\frac{1}{4}\right)^{5-k}$

$= 0.7627$ #

2.

tomy

2. $\begin{cases} H_0: \sigma^2 = 0.81 \\ H_a: \sigma^2 > 0.81 \end{cases}$

$$\chi^2 = \frac{9\cdot(1.2)^2}{(0.9)^2} = 16$$

H_0 is rejected when $\chi^2 > 16.919$

A: accept $\sigma = 0.9$

3(1)

Given an undirected graph **G=(V,E)**, a set of random variables $X = (X_v)_{v \in V}$ indexed by V form a Markov random field with respect to G if they satisfy the local Markov properties:

- Pairwise Markov property: Any two non-adjacent variables are conditionally independent given all other variables: $X_u \perp\!\!\!\perp X_v \mid X_{V \setminus \{u,v\}}$
- Local Markov property: A variable is conditionally independent of all other variables given its neighbors: $X_v \perp\!\!\!\perp X_{V \setminus \mathrm{N}'[v]} \mid X_{\mathrm{N}(v)}$ 非鄰居
  where $\mathrm{N}(v)$ is the set of neighbors of v, $\mathrm{N}'[v] = v \cup \mathrm{N}(v)$ is the closed neighbourhood of v.
- Global Markov property: Any two subsets of variables are conditionally independent given a separating subset: $X_A \perp\!\!\!\perp X_B \mid X_S$
  where every path from a node in A to a node in B passes through S

3(2)

Local Markov property (for DAG)

- Each variable is conditionally independent of its non-descendants given its parent variables:

$$X_v \perp\!\!\!\perp X_{V \setminus \mathrm{de}(v)} \mid X_{\mathrm{pa}(v)} \quad \text{for all } v \in V$$

  where de(v) denotes the set of descendants of v (thus V \ de(v) is the set of non-descendants of v)

3(3)

Definition: Markov blanket

- The Markov blanket of a node is the set of nodes consisting of its parents, its children, and any other parents of its children.

4.

optimal projection vector:
(0.99130435   -0.13158907)

corresponding eigenvalue:
8.6783

```python
import numpy as np

a = np.array([[5, 3], [3, 5], [3, 4], [4, 5], [4, 7], [5, 6]])
b = np.array([[9, 10], [7, 7], [8, 5], [8, 8], [7, 2], [10, 8]])

a_centered = a - np.mean(a, axis=0)
b_centered = b - np.mean(b, axis=0)

s_1 = np.zeros((2, 2))
for i in range(a.shape[0]):
    tmp_a = a_centered[i].reshape(2, 1)
    s_1 += np.matmul(tmp_a, tmp_a.T)

s_1 /= (a.shape[0] - 1)
# print(s_1)

s_2 = np.zeros((2, 2))
for i in range(b.shape[0]):
    tmp_b = b_centered[i].reshape(2, 1)
    s_2 += np.matmul(tmp_b, tmp_b.T)

s_2 /= (b.shape[0] - 1)
# print(s_2)

s_w = s_1 + s_2

tmp = (mean_a - mean_b).reshape(2, 1)
s_b = np.matmul(tmp, tmp.T)

fld = np.matmul(np.linalg.inv(s_w), s_b)
e_val, e_vec = np.linalg.eig(fld)

print(e_val[0])
print(e_vec.T[0])
```