

**TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA KHOA HỌC TỰ NHIÊN
BỘ MÔN TOÁN HỌC**



BÀI THU HOẠCH

CHUYÊN ĐỀ THỐNG KÊ NÂNG CAO

Sinh viên thực hiện
TRẦN TRUNG TÍN
NGÀNH LTXS & TKTH - Khóa 31
MSHV: M1824006

CẦN THƠ - NĂM 2025

TRƯỜNG ĐẠI HỌC CẦN THƠ
KHOA KHOA HỌC TỰ NHIÊN
BỘ MÔN TOÁN HỌC

<><><><><><><><>

BÀI THU HOẠCH

CHUYÊN ĐỀ THỐNG KÊ NÂNG CAO

Sinh viên thực hiện
TRẦN TRUNG TÍN
NGÀNH LTXS & TKTH - Khóa 31
MSHV: M1824006

CẦN THƠ - NĂM 2025

Mục lục

Mục lục	i
DANH SÁCH HÌNH VẼ	v
DANH SÁCH BẢNG	vii
DANH SÁCH KÝ HIỆU VÀ VIẾT TẮT	viii
LỜI CẢM ƠN	1
Chương 1. Kiến thức chuẩn bị	2
1.1 Không gian xác suất và biến ngẫu nhiên	2
1.1.1 Không gian xác suất	2
1.1.2 Biến ngẫu nhiên	3
1.1.3 Hàm phân phối tích luỹ (CDF)	3
1.1.4 Kỳ vọng và phương sai	3
1.2 Xác suất và kỳ vọng có điều kiện	4
1.2.1 Xác suất có điều kiện	4
1.2.2 Kỳ vọng có điều kiện	4
1.3 Một số phân phối xác suất quan trọng	4
1.3.1 Phân phối chuẩn	4
1.3.2 Phân phối Chi-bình phương	5
1.3.3 Phân phối Student	5
1.3.4 Một vài phân phối cơ bản khác	5

1.4	Định lý giới hạn trung tâm và luật số lớn	6
1.4.1	Định lý giới hạn trung tâm (CLT)	6
1.4.2	Luật số lớn	6
1.5	Cơ sở của kiểm định giả thuyết	6
1.5.1	Khái niệm chung	6
1.5.2	Sai lầm và lực kiểm định	7
1.5.3	Ví dụ minh họa: kiểm định trung bình với phương sai biết trước	7
1.6	Khoảng tin cậy và ước lượng	8
1.6.1	Khái niệm khoảng tin cậy	8
1.6.2	Khoảng tin cậy cho trung bình	9
1.6.3	Khoảng tin cậy cho phương sai	9
Chương 2.	Một số dạng kiểm định thống kê	10
2.1	Kiểm định Pearson	10
2.1.1	Cơ sở lý thuyết	10
2.1.2	Ví dụ GOF khác hoàn toàn	11
2.1.3	Ví dụ kiểm định độc lập khác hoàn toàn	11
2.1.4	Thực nghiệm số (MATLAB minh họa)	11
2.2	Bộ hàm MATLAB kèm theo để chạy	12
2.2.1	Sinh mẫu rời rạc và kiểm định Pearson (GOF)	12
2.2.2	Kiểm định Pearson độc lập cho bảng chéo	13
2.2.3	ECDF thủ công và kiểm định K–S một mẫu tổng quát	13
2.2.4	Kiểm định K–S hai mẫu tổng quát	14
2.3	Kiểm định Kolmogorov–Smirnov (K–S)	14
2.3.1	Cơ sở lý thuyết	14
2.3.2	Ví dụ 1 mẫu (khác hoàn toàn)	15
2.3.3	Ví dụ 2 mẫu (khác hoàn toàn)	15
2.3.4	Thực nghiệm số (MATLAB minh họa)	15

2.4	Mở rộng mô hình và thí nghiệm trên dữ liệu	15
2.4.1	Thiết kế thí nghiệm Monte Carlo	15
2.5	Ứng dụng thực tế: Kiểm định với dữ liệu Singleton	16
2.5.1	Hàm tính phân phối của biến tổng hợp	18
2.5.2	Kiểm định Pearson cho dữ liệu Singletons	21
2.5.3	Kiểm định Kolmogorov-Smirnov cho dữ liệu Singletons	23
2.5.4	Áp dụng cho bộ dữ liệu Binaires	28
2.5.5	Kiểm định Pearson và Kolmogorov-Smirnov kết hợp	33
2.6	Ứng dụng với các tập dữ liệu khác	35
2.6.1	Phân tích dữ liệu Binaires	35
2.6.2	Phân tích dữ liệu Meteo	36
2.6.3	So sánh kết quả kiểm định	36
2.6.4	Ứng dụng trong phân tích dữ liệu thực tế	36
2.6.5	Kiểm định trên mô hình tổng hợp	37
2.7	Kết luận chương	38
Chương 3.	Phân tích nhiều chiều dữ liệu thang đo định lượng	40
3.1	Khái niệm cơ bản về dữ liệu nhiều chiều	40
3.1.1	Vector ngẫu nhiên và phân phối nhiều chiều	40
3.1.2	Vector kỳ vọng và ma trận hiệp phương sai	41
3.1.3	Phân phối chuẩn nhiều chiều	41
3.2	Ma trận tương quan và các đặc trưng mô tả	41
3.2.1	Ma trận tương quan	41
3.2.2	Ước lượng mẫu	42
3.3	Phân tích thành phần chính (Principal Component Analysis - PCA) .	42
3.3.1	Động lực và ý tưởng cơ bản	42
3.3.2	Định nghĩa toán học	43
3.3.3	Tính chất quan trọng	43

3.3.4	Thuật toán thực hiện PCA	43
3.3.5	Tiêu chí lựa chọn số thành phần	44
3.3.6	Ví dụ minh họa với MATLAB	44
3.4	Phân tích nhân tố (Factor Analysis)	45
3.4.1	Mô hình nhân tố	45
3.4.2	Giả định của mô hình	45
3.4.3	Phân tích ma trận hiệp phương sai	45
3.4.4	Phương pháp ước lượng	46
3.4.5	Xoay nhân tố (Factor Rotation)	46
3.5	Ứng dụng thực tế: Phân tích dữ liệu với mô hình tổng hợp	46
3.5.1	Phân tích dữ liệu kinh tế - xã hội	46
3.5.2	Bước 1: Đọc và xử lý dữ liệu	47
3.5.3	Bước 2: Phân tích tương quan	49
3.5.4	Bước 3: Kiểm định tính phù hợp cho phân tích nhân tố	50
3.5.5	Bước 4: Phân tích thành phần chính (PCA)	53
3.5.6	Bước 5: Phân tích nhân tố khám phá (EFA)	55
3.5.7	Bước 6: Đánh giá độ tin cậy Cronbach's Alpha	59
3.5.8	Quy trình phân tích tổng hợp	59
3.5.9	Đánh giá hiệu quả mô hình	63
3.6	Phân tích nhân tố khám phá (EFA)	63
3.6.1	Mô hình phân tích nhân tố	64
3.6.2	Kiểm định độ tin cậy Cronbach's Alpha	64
3.7	Kết luận chương	65
KẾT LUẬN		67
TÀI LIỆU THAM KHẢO		73
Tài liệu tham khảo		74
Phụ lục A. Hướng dẫn trích dẫn tài liệu		76

Danh sách hình vẽ

2.1	Kết quả kiểm định Pearson cho dữ liệu Singletons	12
2.2	Biểu đồ ECDF và CDF lý thuyết trong kiểm định Kolmogorov-Smirnov	16
2.3	Hình minh họa cho các biến trạng thái của bộ dữ liệu Singletons . . .	18
2.4	So sánh phân phối tích lũy thực nghiệm và lý thuyết	24
2.5	Sơ đồ quyết định cho dữ liệu Singleton	24
2.6	So sánh phân phối tích lũy thực nghiệm và lý thuyết với K-S test . . .	28
2.7	Sơ đồ mô hình Binaires	31
2.8	Sơ đồ mô hình Meteo	33
2.9	Kết quả kiểm định Kolmogorov-Smirnov cho dữ liệu Singletons . . .	34
2.10	Kết quả kiểm định Pearson cho dữ liệu Singletons	35
2.11	Sơ đồ phụ thuộc cho dữ liệu Binaires	35
2.12	Sơ đồ phụ thuộc cho dữ liệu khí tượng	36
2.13	So sánh kết quả kiểm định Pearson giữa các biến X và Y	36
2.14	Kết quả kiểm định thống kê cho dữ liệu thực tế	37
2.15	Phân tích dữ liệu meteorological với nhiều phương pháp kiểm định .	38
3.1	Minh họa nguyên lý PCA: Tìm hướng phương sai lớn nhất	42
3.2	Ma trận tương quan giữa các biến	51
3.3	Minh họa thành phần phương sai đóng góp	56
3.4	Top 10 các biến đóng góp	56
3.5	Biểu đồ EFA tổng quát	60
3.6	Độ tin cậy Factor 1	60

3.7 Độ tin cậy Factor 2	61
3.8 Độ tin cậy Factor 3	61
3.9 Độ tin cậy Factor 4	61
3.10 Độ tin cậy Factor 5	61
3.11 Độ tin cậy Factor 6	61
3.12 Độ tin cậy Factor 7	61
3.13 Quy trình phân tích dữ liệu nhiều chiều	63
3.14 Kết quả phân tích nhân tố khám phá với phương pháp Maximum Likelihood	65
3.15 Độ tin cậy nhân tố 1	65
3.16 Độ tin cậy nhân tố 2	65
3.17 Độ tin cậy nhân tố 3	66
3.18 Độ tin cậy nhân tố 4	66
3.19 Độ tin cậy nhân tố 5	66

Danh sách bảng

DANH SÁCH KÝ HIỆU VÀ VIẾT TẮT

Ký hiệu toán học

X, Y, Z	Biến ngẫu nhiên
x, y, z	Giá trị của biến ngẫu nhiên
$f(x)$	Hàm mật độ xác suất
$F(x)$	Hàm phân phối tích lũy
$E[X]$	Kỳ vọng của biến ngẫu nhiên X
$\text{Var}(X)$	Phương sai của biến ngẫu nhiên X
σ	Độ lệch chuẩn
μ	Trung bình mẫu
n	Kích thước mẫu
α	Mức ý nghĩa
β	Xác suất sai lầm loại II
H_0	Giả thuyết không
H_1	Giả thuyết đối
χ^2	Phân phối Chi-bình phương
t	Phân phối Student
F	Phân phối Fisher
p -value	Giá trị p

Viết tắt

ĐSTT	Đại số thống kê
HH	Hình học
LTXS	Lý thuyết xác suất
TKTH	Thống kê toán học
CDF	Cumulative Distribution Function
PDF	Probability Density Function
MATLAB	Matrix Laboratory
CTU	Can Tho University

LỜI CẢM ƠN

Em xin chân thành cảm ơn TS. Trần Văn Lý đã tận tình hướng dẫn và giúp đỡ em trong quá trình thực hiện bài thu hoạch này. Thầy đã truyền đạt những kiến thức quý báu về thống kê nâng cao và các phương pháp phân tích dữ liệu, giúp em hiểu sâu sắc hơn về các kỹ thuật kiểm định thống kê và phân tích đa biến.

Em cũng xin gửi lời cảm ơn đến các thầy cô trong Bộ môn Toán học, Khoa Khoa học Tự nhiên, Trường Đại học Cần Thơ đã tạo điều kiện thuận lợi cho việc học tập và nghiên cứu.

Cuối cùng, em xin cảm ơn gia đình và bạn bè đã động viên và hỗ trợ em trong suốt quá trình học tập.

Cần Thơ, tháng 12 năm 2024

Sinh viên

TRẦN TRUNG TÍN

Chương 1

Kiến thức chuẩn bị

1.1 Không gian xác suất và biến ngẫu nhiên

1.1.1 Không gian xác suất

Định nghĩa 1.1 (Không gian xác suất). Một không gian xác suất là bộ (Ω, \mathcal{F}, P) , trong đó Ω là không gian mẫu, \mathcal{F} là σ -đại số các biến cố và P là độ đo xác suất trên \mathcal{F} thỏa:

- $P(A) \geq 0$ với mọi $A \in \mathcal{F}$;
- $P(\Omega) = 1$;
- Với dãy $\{A_i\}$ đôi một rời nhau: $P(\bigcup_{i \geq 1} A_i) = \sum_{i \geq 1} P(A_i)$.

Tính chất 1.1. *Hệ quả cơ bản: $P(A^c) = 1 - P(A)$; nếu $A \subseteq B$ thì $P(A) \leq P(B)$.*

Sự độc lập

Định nghĩa 1.2 (Độc lập của biến cố). Hai biến cố $A, B \in \mathcal{F}$ được gọi là *độc lập* nếu $P(A \cap B) = P(A)P(B)$. Một hệ $\{A_i\}$ là độc lập nếu mọi giao hữu hạn đều có xác suất bằng tích các xác suất thành phần.

Xác suất toàn phần và công thức Bayes

Định lí 1.1 (Xác suất toàn phần). *Nếu $\{B_i\}_{i \geq 1}$ là một phân hoạch của Ω với $P(B_i) > 0$ thì với mọi biến cố A ta có*

$$P(A) = \sum_i P(A | B_i) P(B_i).$$

Định lí 1.2 (Công thức Bayes). *Với ký hiệu như trên, với mỗi j thỏa $P(B_j) > 0$ và $P(A) > 0$,*

$$P(B_j | A) = \frac{P(A | B_j) P(B_j)}{\sum_i P(A | B_i) P(B_i)}.$$

1.1.2 Biến ngẫu nhiên

Định nghĩa 1.3 (Biến ngẫu nhiên). Biến ngẫu nhiên $X : \Omega \rightarrow \mathbb{R}$ là hàm đo được (tức $\{\omega : X(\omega) \leq x\} \in \mathcal{F}$ với mọi $x \in \mathbb{R}$).

Hai lớp thường gặp:

- **Rời rạc:** X nhận các giá trị $\{x_1, x_2, \dots\}$ với hàm khối xác suất (PMF)

$$p_X(x) = P(X = x), \quad p_X(x) \geq 0, \quad \sum_x p_X(x) = 1.$$

- **Liên tục:** tồn tại hàm mật độ (PDF) $f_X(x) \geq 0$ sao cho với mọi $a < b$,

$$P(a \leq X \leq b) = \int_a^b f_X(x) dx, \quad \int_{-\infty}^{\infty} f_X(x) dx = 1.$$

1.1.3 Hàm phân phối tích luỹ (CDF)

Định nghĩa 1.4 (Hàm phân phối). Hàm phân phối của X là $F_X(x) = P(X \leq x)$. Với X rời rạc: $F_X(x) = \sum_{x_i \leq x} p_X(x_i)$. Với X liên tục có mật độ f_X , ta có $F'_X(x) = f_X(x)$ hầu khắp nơi.

1.1.4 Kỳ vọng và phương sai

$$\mathbb{E}[X] = \begin{cases} \sum_x x p_X(x), & \text{rời rạc,} \\ \int_{-\infty}^{\infty} x f_X(x) dx, & \text{liên tục,} \end{cases} \quad \text{Var}(X) = \mathbb{E}[(X - \mathbb{E}X)^2].$$

Tính chất 1.2. $\mathbb{E}(aX + b) = a\mathbb{E}X + b$; nếu X, Y độc lập thì $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$.

Hiệp phương sai và một số công thức quan trọng

Định nghĩa 1.5 (Hiệp phương sai). Với hai đại lượng ngẫu nhiên khả tích X, Y , đặt $\text{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}X)(Y - \mathbb{E}Y)]$. Khi X và Y độc lập thì $\text{Cov}(X, Y) = 0$.

Định lí 1.3 (Các công thức phương sai). (i) $\text{Var}(X) = \mathbb{E}(X^2) - [\mathbb{E}(X)]^2$.

(ii) $\text{Var}(aX) = a^2 \text{Var}(X)$ và $\text{Var}(X + a) = \text{Var}(X)$ với mọi hằng số a .

(iii) $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y) + 2\text{Cov}(X, Y)$; đặc biệt, nếu X và Y độc lập thì $\text{Var}(X + Y) = \text{Var}(X) + \text{Var}(Y)$.

1.2 Xác suất và kỳ vọng có điều kiện

1.2.1 Xác suất có điều kiện

Định nghĩa 1.6 (Xác suất có điều kiện). Với các biến cỗ A, B và $P(A) > 0$, xác suất có điều kiện của B khi biết A đã xảy ra được định nghĩa bởi:

$$P(B | A) = \frac{P(A \cap B)}{P(A)}$$

Tính chất 1.3. Các tính chất cơ bản của xác suất có điều kiện:

- $0 \leq P(B | A) \leq 1$ với mọi biến cỗ B
- $P(A | A) = 1$ (tính phản xạ)
- $P(B^c | A) = 1 - P(B | A)$ (tính bù)
- Nếu B_1, B_2, \dots là các biến cỗ đôi một rời nhau thì $P(\bigcup_{i=1}^{\infty} B_i | A) = \sum_{i=1}^{\infty} P(B_i | A)$

Quy tắc nhân xác suất

Định lí 1.4 (Quy tắc nhân). Với các biến cỗ A_1, A_2, \dots, A_n sao cho $P(A_1 \cap A_2 \cap \dots \cap A_{n-1}) > 0$, ta có:

$$P(A_1 \cap A_2 \cap \dots \cap A_n) = P(A_1) \cdot P(A_2 | A_1) \cdot P(A_3 | A_1 \cap A_2) \cdots P(A_n | A_1 \cap A_2 \cap \dots \cap A_{n-1})$$

Độc lập có điều kiện

Định nghĩa 1.7 (Độc lập có điều kiện). Hai biến cỗ A và B được gọi là độc lập có điều kiện với biến cỗ C (với $P(C) > 0$) nếu:

$$P(A \cap B | C) = P(A | C) \cdot P(B | C)$$

1.2.2 Kỳ vọng có điều kiện

Định nghĩa 1.8 (Kỳ vọng có điều kiện với biến cỗ). Với biến cỗ A có $P(A) > 0$ và biến ngẫu nhiên khả tích Y , kỳ vọng có điều kiện của Y khi biết A đã xảy ra được định nghĩa bởi:

$$\mathbb{E}(Y | A) = \frac{1}{P(A)} \int_A Y dP$$

Định nghĩa 1.9 (Kỳ vọng có điều kiện với biến ngẫu nhiên). Với biến ngẫu nhiên X và Y khả tích, kỳ vọng có điều kiện $\mathbb{E}(Y | X)$ là một biến ngẫu nhiên đo được đối với $\sigma(X)$ thỏa mãn:

$$\int_A \mathbb{E}(Y | X) dP = \int_A Y dP$$

với mọi $A \in \sigma(X)$.

Tính chất 1.4. Các tính chất quan trọng của kỳ vọng có điều kiện:

- **Tính tuyến tính:** $\mathbb{E}(aY_1 + bY_2 | X) = a\mathbb{E}(Y_1 | X) + b\mathbb{E}(Y_2 | X)$
- **Tính tháp:** $\mathbb{E}[\mathbb{E}(Y | X)] = \mathbb{E}(Y)$
- **Độc lập:** Nếu X và Y độc lập thì $\mathbb{E}(Y | X) = \mathbb{E}(Y)$
- **Tính chất đo được:** $\mathbb{E}(Y | X)$ là hàm đo được của X

Phương sai có điều kiện

Định nghĩa 1.10 (Phương sai có điều kiện). Phương sai có điều kiện của Y khi biết X được định nghĩa bởi:

$$\text{Var}(Y | X) = \mathbb{E}[(Y - \mathbb{E}(Y | X))^2 | X] = \mathbb{E}(Y^2 | X) - [\mathbb{E}(Y | X)]^2$$

Định lí 1.5 (Công thức phân rã phương sai). *Với các biến ngẫu nhiên X và Y có phương sai hữu hạn:*

$$Var(Y) = \mathbb{E}[Var(Y | X)] + Var[\mathbb{E}(Y | X)]$$

Ví dụ minh họa

Xét một hộp đựng 3 quả bóng đỏ và 2 quả bóng xanh. Lấy ngẫu nhiên 2 quả bóng không hoàn lại. Gọi X là số quả bóng đỏ trong lần lấy đầu tiên và Y là tổng số quả bóng đỏ sau 2 lần lấy.

Tính toán:

- $P(X = 1) = \frac{3}{5}, P(X = 0) = \frac{2}{5}$
- $\mathbb{E}(Y | X = 1) = 1 + \frac{2}{4} = 1.5$ (1 quả đỏ đã lấy + xác suất lấy thêm 1 quả đỏ từ 4 quả còn lại)
- $\mathbb{E}(Y | X = 0) = 0 + \frac{3}{4} = 0.75$ (0 quả đỏ đã lấy + xác suất lấy 1 quả đỏ từ 4 quả còn lại)
- $\mathbb{E}(Y) = \mathbb{E}[\mathbb{E}(Y | X)] = 1.5 \cdot \frac{3}{5} + 0.75 \cdot \frac{2}{5} = 1.2$

1.3 Một số phân phối xác suất quan trọng

Phân phối xác suất là nền tảng của thống kê toán học, mô tả cách các giá trị của biến ngẫu nhiên được phân bố. Dưới đây là các phân phối quan trọng nhất thường gặp trong thực tế và lý thuyết thống kê.

1.3.1 Phân phối chuẩn (Normal Distribution)

Định nghĩa 1.11 (Phân phối chuẩn). Biến ngẫu nhiên X được gọi là có phân phối chuẩn với tham số vị trí μ và tham số tỷ lệ σ^2 , ký hiệu $X \sim \mathcal{N}(\mu, \sigma^2)$, nếu hàm mật độ xác suất của nó có dạng:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), x \in \mathbb{R}$$

Tính chất 1.5. Các tính chất quan trọng của phân phối chuẩn:

- **Đối xứng:** Phân phối đối xứng quanh giá trị trung bình μ
- **Đặc trưng số:** $\mathbb{E}X = \mu$, $Var(X) = \sigma^2$
- **Quy tắc 68-95-99.7:** Khoảng $[\mu - \sigma, \mu + \sigma]$ chứa 68.27% dữ liệu, $[\mu - 2\sigma, \mu + 2\sigma]$ chứa 95.45%, $[\mu - 3\sigma, \mu + 3\sigma]$ chứa 99.73%
- **Tính chất tuyến tính:** Nếu $X \sim \mathcal{N}(\mu, \sigma^2)$ thì $aX + b \sim \mathcal{N}(a\mu + b, a^2\sigma^2)$
- **Tổng các biến chuẩn độc lập:** Nếu $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$ độc lập thì $\sum_{i=1}^n X_i \sim \mathcal{N}(\sum_{i=1}^n \mu_i, \sum_{i=1}^n \sigma_i^2)$

Phân phối chuẩn chuẩn hóa

Khi $\mu = 0$ và $\sigma^2 = 1$, ta có phân phối chuẩn hóa $Z \sim \mathcal{N}(0, 1)$ với hàm mật độ:

$$\phi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$$

và hàm phân phối tích lũy:

$$\Phi(z) = \int_{-\infty}^z \phi(t) dt$$

Ứng dụng thực tế

Phân phối chuẩn xuất hiện trong nhiều hiện tượng tự nhiên:

- Chiều cao, cân nặng của con người
- Điểm số trong các bài kiểm tra
- Sai số đo lường trong thí nghiệm
- Lợi nhuận của các khoản đầu tư tài chính

1.3.2 Phân phối Chi-bình phương (χ^2)

Định nghĩa 1.12 (Phân phối Chi-bình phương). Nếu $Z_1, \dots, Z_n \stackrel{i.i.d.}{\sim} \mathcal{N}(0, 1)$ và $X = \sum_{i=1}^n Z_i^2$ thì $X \sim \chi^2(n)$. Hàm mật độ xác suất:

$$f(x) = \begin{cases} \frac{e^{-x/2} x^{\frac{n}{2}-1}}{2^{n/2} \Gamma(\frac{n}{2})}, & x > 0, \\ 0, & x \leq 0. \end{cases}$$

Tính chất 1.6. Các tính chất của phân phối χ^2 :

- **Tham số:** n được gọi là bậc tự do (degrees of freedom)
- **Đặc trưng số:** $\mathbb{E}(X) = n$, $\text{Var}(X) = 2n$
- **Tính chất cộng tính:** Nếu $X_1 \sim \chi^2(n_1)$ và $X_2 \sim \chi^2(n_2)$ độc lập thì $X_1 + X_2 \sim \chi^2(n_1 + n_2)$
- **Phân phối không âm:** Chỉ nhận giá trị không âm
- **Phân phối lệch phải:** Đầu phân phối kéo dài về phía phải

Phân vị Chi-bình phương Giá trị $\chi_{\alpha,n}^2$ được xác định bởi $P(X \leq \chi_{\alpha,n}^2) = \alpha$ với $X \sim \chi^2(n)$. Các bảng phân vị thường dùng để thiết lập miền bắc bỏ trong kiểm định phuơng sai và xây dựng khoảng tin cậy cho phuơng sai.

Ứng dụng trong thống kê

- **Kiểm định phuơng sai:** So sánh phuơng sai mẫu với phuơng sai tổng thể
- **Phân tích bảng tương quan:** Kiểm định tính độc lập giữa các biến phân loại
- **Khoảng tin cậy cho phuơng sai:** Xây dựng khoảng tin cậy cho phuơng sai tổng thể

1.3.3 Phân phối Student (t-distribution)

Định nghĩa 1.13 (Phân phối Student). Nếu $Z \sim \mathcal{N}(0, 1)$ độc lập với $V \sim \chi^2(n)$ thì $T = \frac{Z}{\sqrt{V/n}} \sim t(n)$. Hàm mật độ:

$$f(t) = \frac{\Gamma(\frac{n+1}{2})}{\sqrt{n\pi}\Gamma(\frac{n}{2})} \left(1 + \frac{t^2}{n}\right)^{-(n+1)/2}, \quad t \in \mathbb{R}$$

Tính chất 1.7. Các tính chất của phân phối Student:

- **Đối xứng:** Phân phối đối xứng quanh 0
- **Đặc trưng số:** Với $n > 1$: $\mathbb{E}(T) = 0$; với $n > 2$: $Var(T) = \frac{n}{n-2}$
- **Hội tụ về chuẩn:** Khi $n \rightarrow \infty$, phân phối $t(n)$ hội tụ về $\mathcal{N}(0, 1)$
- **Đuôi nặng hơn chuẩn:** Với n hữu hạn, đuôi phân phối nặng hơn phân phối chuẩn

Ứng dụng chính

- **Kiểm định trung bình:** Khi phương sai tổng thể chưa biết
- **Khoảng tin cậy cho trung bình:** Với mẫu nhỏ và phương sai chưa biết
- **So sánh hai trung bình:** Kiểm định t cho hai mẫu độc lập hoặc ghép cặp

1.3.4 Phân phối F (Fisher-Snedecor)

Định nghĩa 1.14 (Phân phối F). Nếu $X_1 \sim \chi^2(n_1)$ và $X_2 \sim \chi^2(n_2)$ độc lập thì $F = \frac{X_1/n_1}{X_2/n_2} \sim F(n_1, n_2)$. Hàm mật độ:

$$f(x) = \frac{\Gamma(\frac{n_1+n_2}{2})}{\Gamma(\frac{n_1}{2})\Gamma(\frac{n_2}{2})} \left(\frac{n_1}{n_2}\right)^{n_1/2} \frac{x^{n_1/2-1}}{(1+\frac{n_1x}{n_2})^{(n_1+n_2)/2}}, \quad x > 0$$

Tính chất 1.8. Các tính chất của phân phối F:

- **Tham số:** n_1, n_2 là các bậc tự do
- **Đặc trưng số:** Với $n_2 > 2$: $\mathbb{E}(F) = \frac{n_2}{n_2-2}$; với $n_2 > 4$: $Var(F) = \frac{2n_2^2(n_1+n_2-2)}{n_1(n_2-2)^2(n_2-4)}$
- **Tính chất nghịch đảo:** Nếu $F \sim F(n_1, n_2)$ thì $\frac{1}{F} \sim F(n_2, n_1)$

1.3.5 Các phân phối rời rạc quan trọng

Phân phối Bernoulli

Định nghĩa 1.15 (Phân phối Bernoulli). Biến ngẫu nhiên X có phân phối Bernoulli với tham số p (ký hiệu $X \sim \text{Bernoulli}(p)$) nếu:

$$P(X = 1) = p, \quad P(X = 0) = 1 - p$$

Tính chất 1.9. $\mathbb{E}[X] = p$, $\text{Var}(X) = p(1 - p)$. Phân phối này mô tả kết quả của một thí nghiệm với hai kết cục: thành công (1) hoặc thất bại (0).

Phân phối nhị thức

Định nghĩa 1.16 (Phân phối nhị thức). Biến ngẫu nhiên X có phân phối nhị thức với tham số n và p (ký hiệu $X \sim \text{B}(n, p)$) nếu:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}, \quad k = 0, 1, \dots, n$$

Tính chất 1.10. $\mathbb{E}[X] = np$, $\text{Var}(X) = np(1 - p)$. Phân phối này mô tả số lần thành công trong n thí nghiệm Bernoulli độc lập.

Điều kiện áp dụng: Khi n lớn và p không quá gần 0 hoặc 1, phân phối nhị thức có thể xấp xỉ bằng phân phối chuẩn $\mathcal{N}(np, np(1 - p))$.

Phân phối Poisson

Định nghĩa 1.17 (Phân phối Poisson). Biến ngẫu nhiên X có phân phối Poisson với tham số λ (ký hiệu $X \sim \text{Poisson}(\lambda)$) nếu:

$$P(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}, \quad k = 0, 1, 2, \dots$$

Tính chất 1.11. $\mathbb{E}[X] = \text{Var}(X) = \lambda$. Phân phối Poisson mô tả số sự kiện xảy ra trong một khoảng thời gian hoặc không gian cố định.

Ứng dụng: Số cuộc gọi đến tổng đài, số tai nạn giao thông, số lỗi trong sản xuất, số khách hàng đến cửa hàng.

1.3.6 Các phân phối liên tục khác

Phân phối đều

Định nghĩa 1.18 (Phân phối đều). Biến ngẫu nhiên X có phân phối đều trên đoạn $[a, b]$ (ký hiệu $X \sim U(a, b)$) nếu hàm mật độ:

$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{trường hợp khác} \end{cases}$$

Tính chất 1.12. $\mathbb{E}[X] = \frac{a+b}{2}$, $Var(X) = \frac{(b-a)^2}{12}$. Phân phối đều mô tả sự kiện ngẫu nhiên có xác suất đồng đều trên một khoảng.

Phân phối mũ

Định nghĩa 1.19 (Phân phối mũ). Biến ngẫu nhiên X có phân phối mũ với tham số $\lambda > 0$ (ký hiệu $X \sim Exp(\lambda)$) nếu hàm mật độ:

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Tính chất 1.13. $\mathbb{E}[X] = \frac{1}{\lambda}$, $Var(X) = \frac{1}{\lambda^2}$. Phân phối mũ có tính chất "không nhớ": $P(X > s+t | X > s) = P(X > t)$.

Ứng dụng: Thời gian giữa các sự kiện trong quá trình Poisson, tuổi thọ của thiết bị điện tử, thời gian phục vụ khách hàng.

Phân phối Gamma

Định nghĩa 1.20 (Phân phối Gamma). Biến ngẫu nhiên X có phân phối Gamma với tham số hình dạng $\alpha > 0$ và tham số tỷ lệ $\beta > 0$ (ký hiệu $X \sim Gamma(\alpha, \beta)$) nếu hàm mật độ:

$$f(x) = \begin{cases} \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

Tính chất 1.14. $\mathbb{E}[X] = \frac{\alpha}{\beta}$, $Var(X) = \frac{\alpha}{\beta^2}$. Phân phối Gamma là tổng quát hóa của phân phối mũ và chi-bình phương.

Trường hợp đặc biệt: $Gamma(1, \lambda) = Exp(\lambda)$, $Gamma(n/2, 1/2) = \chi^2(n)$.

1.4 Định lý giới hạn trung tâm và luật số lớn

Đây là hai định lý cơ bản nhất của lý thuyết xác suất, tạo nền tảng cho thống kê suy diễn và giải thích tại sao phân phối chuẩn đóng vai trò quan trọng trong thống kê ứng dụng.

1.4.1 Định lý giới hạn trung tâm (Central Limit Theorem - CLT)

Phát biểu cơ bản

Định lí 1.6 (Định lý giới hạn trung tâm cho dãy i.i.d.). *Với X_1, \dots, X_n độc lập cùng phân phối (i.i.d.), $\mathbb{E}(X_i) = \mu$, $Var(X_i) = \sigma^2 \in (0, \infty)$, đặt*

$$\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i, \quad Z_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma}.$$

Khi $n \rightarrow \infty$, $Z_n \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$.

Tính chất 1.15. Ý nghĩa thực tế: CLT giải thích tại sao trung bình mẫu của hầu hết các biến ngẫu nhiên đều có phân phối xấp xỉ chuẩn khi kích thước mẫu đủ lớn, bất kể phân phối gốc của dữ liệu như thế nào.

Điều kiện Lindeberg

Định lí 1.7 (CLT cho độc lập không đồng phân phối (điều kiện Lindeberg)). *Giả sử X_1, X_2, \dots độc lập, $\mathbb{E}(X_i) = 0$, $Var(X_i) = \sigma_i^2 < \infty$ và $s_n^2 = \sum_{i=1}^n \sigma_i^2 \rightarrow \infty$. Nếu với mọi $\varepsilon > 0$,*

$$\lim_{n \rightarrow \infty} \frac{1}{s_n^2} \sum_{i=1}^n \mathbb{E} \left[X_i^2 \mathbf{1}_{\{|X_i| > \varepsilon s_n\}} \right] = 0,$$

thì $\frac{1}{s_n} \sum_{i=1}^n X_i \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$.

Tính chất 1.16. *Điều kiện Lindeberg đảm bảo rằng không có biến ngẫu nhiên nào trong tổng có ảnh hưởng quá lớn đến tổng thể, nghĩa là mỗi thành phần đều "nhỏ" so với tổng.*

Tốc độ hội tụ và xấp xỉ

Định lí 1.8 (Định lý Berry-Esseen). *Với X_1, \dots, X_n i.i.d. có kỳ vọng μ , phương sai σ^2 và moment bậc 3 hữu hạn $\rho = \mathbb{E}|X_1 - \mu|^3$, ta có:*

$$\sup_{x \in \mathbb{R}} \left| P\left(\frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma} \leq x\right) - \Phi(x) \right| \leq \frac{C\rho}{\sigma^3 \sqrt{n}}$$

với C là hằng số phổ quát ($C \approx 0.4748$).

Tính chất 1.17. *Định lý này cho ta ước lượng tốc độ hội tụ của CLT, cho thấy sai số xấp xỉ giảm theo tốc độ $O(1/\sqrt{n})$.*

Ứng dụng thực tế

- **Khoảng tin cậy:** Xây dựng khoảng tin cậy cho trung bình tổng thể dựa trên phân phối chuẩn
- **Kiểm định giả thuyết:** Sử dụng thống kê Z-test cho trung bình
- **Ước lượng điểm:** Trung bình mẫu là ước lượng không chêch và hiệu quả cho trung bình tổng thể
- **Phân tích dữ liệu:** Giải thích tại sao nhiều hiện tượng tự nhiên có phân phối chuẩn

Ví dụ minh họa

Xét dữ liệu về chiều cao của 1000 người trưởng thành. Mặc dù chiều cao có thể không tuân theo phân phối chuẩn chính xác, nhưng trung bình của các mẫu ngẫu nhiên với kích thước $n = 30$ sẽ có phân phối xấp xỉ chuẩn.

Mô phỏng:

- Lấy 1000 mẫu, mỗi mẫu có 30 quan sát
- Tính trung bình của mỗi mẫu
- Vẽ histogram của 1000 trung bình mẫu
- Kết quả sẽ cho thấy phân phối xấp xỉ chuẩn

1.4.2 Luật số lớn (Law of Large Numbers)

Luật số lớn yếu (Weak Law of Large Numbers)

Định lí 1.9 (Luật số lớn yếu). *Với X_1, X_2, \dots i.i.d. có kỳ vọng μ hữu hạn, đặt $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$. Khi đó:*

$$\bar{X}_n \xrightarrow{P} \mu \quad \text{khi } n \rightarrow \infty$$

Tính chất 1.18. *Luật số lớn yếu khẳng định rằng trung bình mẫu hội tụ theo xác suất về trung bình tổng thể khi kích thước mẫu tăng lên vô hạn.*

Luật số lớn mạnh (Strong Law of Large Numbers)

Định lí 1.10 (Luật số lớn mạnh). *Với X_1, X_2, \dots i.i.d. có kỳ vọng μ hữu hạn, đặt $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$. Khi đó:*

$$\bar{X}_n \xrightarrow{a.s.} \mu \quad \text{khi } n \rightarrow \infty$$

Tính chất 1.19. *Luật số lớn mạnh khẳng định rằng trung bình mẫu hội tụ hầu chắc chắn về trung bình tổng thể, mạnh hơn hội tụ theo xác suất.*

Luật số lớn cho biến ngẫu nhiên không độc lập

Định lí 1.11 (Luật số lớn cho dãy m-phụ thuộc). *Giả sử X_1, X_2, \dots là dãy m-phụ thuộc (tức là X_i và X_j độc lập khi $|i - j| > m$) với $\mathbb{E}(X_i) = \mu$ và $\text{Var}(X_i) \leq C$ với mọi i . Khi đó:*

$$\bar{X}_n \xrightarrow{P} \mu \quad \text{khi } n \rightarrow \infty$$

Tốc độ hội tụ

Định lí 1.12 (Bất đẳng thức Chebyshev cho trung bình mẫu). *Với X_1, \dots, X_n i.i.d. có kỳ vọng μ và phương sai σ^2 hữu hạn:*

$$P(|\bar{X}_n - \mu| \geq \varepsilon) \leq \frac{\sigma^2}{n\varepsilon^2}$$

Tính chất 1.20. *Bất đẳng thức này cho ta ước lượng tốc độ hội tụ của luật số lớn yếu, cho thấy xác suất sai lệch giảm theo tốc độ $O(1/n)$.*

Ứng dụng thực tế

- **Ước lượng điểm:** Trung bình mẫu là ước lượng vững cho trung bình tổng thể
- **Monte Carlo:** Phương pháp tính gần đúng tích phân và tổng bằng cách lấy trung bình của các giá trị mẫu
- **Thống kê Bayes:** Cập nhật niềm tin dựa trên dữ liệu quan sát
- **Học máy:** Thuật toán gradient descent và các phương pháp tối ưu hóa khác

Mối quan hệ với CLT

Tính chất 1.21. *Luật số lớn cho ta biết rằng trung bình mẫu hội tụ về trung bình tổng thể, trong khi CLT cho ta biết phân phối của sai lệch giữa trung bình mẫu và trung bình tổng thể khi kích thước mẫu lớn.*

Cụ thể, nếu đặt $Z_n = \frac{\sqrt{n}(\bar{X}_n - \mu)}{\sigma}$, thì:

- **Luật số lớn:** $Z_n / \sqrt{n} \xrightarrow{P} 0$
- **CLT:** $Z_n \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$

Ví dụ ứng dụng

Xét việc ước lượng xác suất của một sự kiện A bằng tần suất tương đối:

$$\hat{p}_n = \frac{1}{n} \sum_{i=1}^n I_A(X_i)$$

trong đó I_A là hàm chỉ báo của sự kiện A .

Theo luật số lớn: $\hat{p}_n \xrightarrow{P} P(A)$ **Theo CLT:** $\frac{\sqrt{n}(\hat{p}_n - P(A))}{\sqrt{P(A)(1-P(A))}} \xrightarrow{\mathcal{D}} \mathcal{N}(0, 1)$

Điều này cho phép ta xây dựng khoảng tin cậy cho xác suất $P(A)$ dựa trên tần suất mẫu.

1.5 Cơ sở của kiểm định giả thuyết

1.5.1 Khái niệm chung

Định nghĩa 1.21 (Bài toán kiểm định). Gồm hai giả thuyết: H_0 (giả thuyết kiểm định) và H_1 (giả thuyết thay thế). Ra quyết định dựa trên thống kê kiểm định, miền bác bỏ và mức ý nghĩa α .

Kiểm định giả thuyết là một phương pháp suy diễn thống kê nhằm đưa ra quyết định về một hoặc nhiều tham số của tổng thể dựa trên thông tin từ mẫu. Quá trình này bao gồm các bước:

1. **Thiết lập giả thuyết:** Xác định H_0 (giả thuyết không) và H_1 (giả thuyết đối).
2. **Chọn mức ý nghĩa:** Thường là $\alpha = 0.05, 0.01$ hoặc 0.10 .
3. **Xác định thống kê kiểm định:** Một hàm của mẫu có phân phối đã biết dưới H_0 .
4. **Tính giá trị quan sát:** Tính giá trị của thống kê kiểm định từ dữ liệu mẫu.
5. **Ra quyết định:** So sánh với giá trị tới hạn hoặc tính p-value.

1.5.2 Sai lầm và lực kiểm định

Tính chất 1.22. - *Sai lầm loại I:* bác bỏ H_0 khi H_0 đúng; xác suất bằng α .

- *Sai lầm loại II:* chấp nhận H_0 khi H_1 đúng; xác suất là β .

- *Lực kiểm định:* $1 - \beta$.

Mối quan hệ giữa các loại sai lầm có thể được minh họa qua bảng sau:

Quyết định	H_0 đúng	H_1 đúng
Chấp nhận H_0	Đúng (xác suất $1 - \alpha$)	Sai lầm loại II (xác suất β)
Bắc bỏ H_0	Sai lầm loại I (xác suất α)	Đúng (xác suất $1 - \beta$)

P-value và ý nghĩa thống kê

Định nghĩa 1.22 (P-value). P-value là xác suất thu được một giá trị của thống kê kiểm định cực đoan ít nhất bằng giá trị quan sát được, giả sử H_0 là đúng.

Quy tắc quyết định dựa trên p-value:

- Nếu p-value $\leq \alpha$: bác bỏ H_0
- Nếu p-value $> \alpha$: không bác bỏ H_0

1.5.3 Ví dụ minh họa: kiểm định trung bình với phương sai biết trước

Giả sử $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} \mathcal{N}(\mu, \sigma^2)$, σ^2 đã biết. Kiểm định

$$H_0 : \mu = \mu_0 \quad \text{vs.} \quad H_1 : \mu \neq \mu_0.$$

Đặt thống kê kiểm định

$$Z = \frac{\sqrt{n}(\bar{X}_n - \mu_0)}{\sigma} \sim \mathcal{N}(0, 1) \text{ dưới } H_0.$$

Với mức ý nghĩa α , miền bác bỏ hai phía là $\{|Z| > z_{1-\alpha/2}\}$, trong đó $z_{1-\alpha/2}$ là phân vị tương ứng của chuẩn tắc. Lực kiểm định có thể tính tường minh dưới H_1 nhờ phân phối chuẩn lệch tâm của Z .

Ví dụ số cụ thể

Một nhà máy sản xuất pin với tuổi thọ trung bình được quảng cáo là 500 giờ. Để kiểm tra, ta lấy mẫu 25 viên pin và đo được tuổi thọ trung bình mẫu là $\bar{x} = 485$ giờ. Biết rằng độ lệch chuẩn tổng thể $\sigma = 40$ giờ. Với mức ý nghĩa $\alpha = 0.05$, hãy kiểm định xem tuổi thọ trung bình có khác 500 giờ không?

Giải:

- $H_0 : \mu = 500$ vs $H_1 : \mu \neq 500$
- Thống kê kiểm định: $Z = \frac{\bar{X} - 500}{\sigma/\sqrt{n}} = \frac{485 - 500}{40/\sqrt{25}} = \frac{-15}{8} = -1.875$
- Giá trị tối hạn: $z_{0.025} = 1.96$
- Vì $|Z| = 1.875 < 1.96$, ta không bác bỏ H_0
- P-value = $2P(Z \leq -1.875) \approx 0.061 > 0.05$

Kết luận: Không có bằng chứng thống kê để khẳng định tuổi thọ trung bình khác 500 giờ.

1.6 Khoảng tin cậy và ước lượng

1.6.1 Khái niệm khoảng tin cậy

Định nghĩa 1.23 (Khoảng tin cậy). Khoảng tin cậy $100(1 - \alpha)\%$ cho tham số θ là khoảng ngẫu nhiên $[L, U]$ sao cho

$$P(L \leq \theta \leq U) = 1 - \alpha$$

1.6.2 Khoảng tin cậy cho trung bình

Trường hợp phương sai đã biết

Với $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} \mathcal{N}(\mu, \sigma^2)$, khoảng tin cậy $100(1 - \alpha)\%$ cho μ là:

$$\bar{X} \pm z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}$$

Trường hợp phương sai chưa biết

Với $X_1, \dots, X_n \stackrel{i.i.d.}{\sim} \mathcal{N}(\mu, \sigma^2)$, khoảng tin cậy $100(1 - \alpha)\%$ cho μ là:

$$\bar{X} \pm t_{1-\alpha/2, n-1} \frac{S}{\sqrt{n}}$$

trong đó $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$.

1.6.3 Khoảng tin cậy cho phương sai

Với mẫu từ phân phối chuẩn, khoảng tin cậy $100(1 - \alpha)\%$ cho σ^2 là:

$$\left[\frac{(n-1)S^2}{\chi_{1-\alpha/2,n-1}^2}, \frac{(n-1)S^2}{\chi_{\alpha/2,n-1}^2} \right]$$

Chương 2

Một số dạng kiểm định thống kê

Chương này trình bày các phương pháp kiểm định thống kê quan trọng, bao gồm kiểm định Pearson, Kolmogorov-Smirnov và các kiểm định phi tham số [13, 8, 16].

2.1 Kiểm định Pearson

2.1.1 Cơ sở lý thuyết

Định nghĩa 2.1 (Bài toán phù hợp phân phối (GOF)). Cho một mẫu quan sát rời rạc được nhóm thành r khoảng (bin) với số quan sát O_i và xác suất kỳ vọng theo mô hình p_i ($i = 1, \dots, r$). Đặt $E_i = Np_i$ là tần số kỳ vọng. Thống kê kiểm định Pearson là

$$\chi^2 = \sum_{i=1}^r \frac{(O_i - E_i)^2}{E_i}.$$

Khi N đủ lớn và mọi $p_i > 0$, dưới H_0 ta có $\chi^2 \stackrel{d}{\approx} \chi^2(r-1)$.

Định nghĩa 2.2 (Kiểm định độc lập (bảng chéo $r \times c$)). Với bảng số liệu O_{ij} ($i = 1, \dots, r$, $j = 1, \dots, c$), đặt tổng hàng $O_{i\cdot}$, tổng cột $O_{\cdot j}$ và $N = \sum_{i,j} O_{ij}$. Dưới giả thuyết “hàng và cột độc lập”, tần số kỳ vọng là $E_{ij} = \frac{O_{i\cdot}O_{\cdot j}}{N}$, và

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \stackrel{d}{\approx} \chi^2((r-1)(c-1)).$$

Tính chất 2.1. - *Điều kiện kinh điển để áp dụng: các quan sát độc lập; $E_i \geq 5$ (GOF) hoặc $E_{ij} \geq 5$ (bảng chéo) với phần lớn ô; kích thước mẫu đủ lớn.* - *Quy tắc bác bỏ ở mức ý nghĩa α : bác bỏ H_0 nếu $\chi^2 > \chi^2_{1-\alpha, v}$ với bậc tự do v tương ứng.*

2.1.2 Ví dụ GOF khác hoàn toàn

Khảo sát $N = 200$ người về màu yêu thích trong 4 màu: Đỏ, Xanh dương, Xanh lá, Vàng. Giả thuyết H_0 : phân phối đồng đều ($p_i = 0.25$). Dữ liệu quan sát: $O = (62, 41, 53, 44)$. Khi đó $E_i = 50$.

Tính

$$\chi^2 = \sum_{i=1}^4 \frac{(O_i - 50)^2}{50} = \frac{12^2}{50} + \frac{(-9)^2}{50} + \frac{3^2}{50} + \frac{(-6)^2}{50} = 5.40.$$

Với $v = 3$ và $\alpha = 0.05$, $\chi^2_{0.95,3} = 7.815$. Vì $5.40 < 7.815$, **không bác bỏ** H_0 . Ước lượng p-value ≈ 0.145 .

2.1.3 Ví dụ kiểm định độc lập khác hoàn toàn

Nghiên cứu mối liên hệ giữa thói quen tập thể dục (Hàng ngày/Thỉnh thoảng) và tình trạng hút thuốc (Không hút/Đã bỏ/Hút hiện tại) trên $N = 160$ người:

		Không hút	Đã bỏ	Hút hiện tại
$O =$	Hàng ngày	48	22	10
	Thỉnh thoảng	36	28	16

Từ đó $E = \begin{pmatrix} 42, 25, 13 \\ 42, 25, 13 \end{pmatrix}$. Tính $\chi^2 = 3.82$ với $v = (2-1)(3-1) = 2$. Vì $\chi^2_{0.95,2} = 5.991$ nên **không bác bỏ** H_0 (p-value ≈ 0.148).

2.1.4 Thực nghiệm số (MATLAB minh họa)

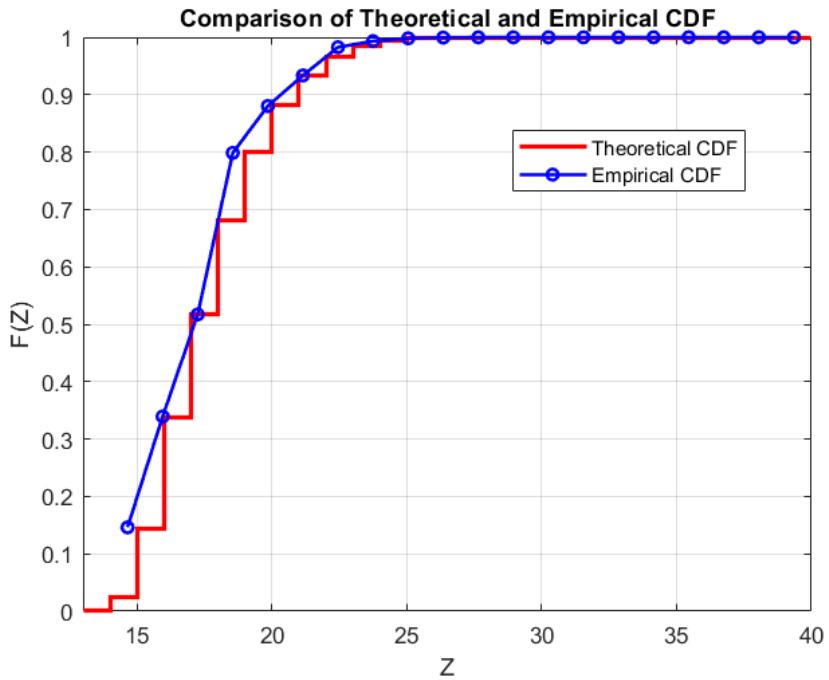
M. 2.1.



```
function [chi2_stat, chi2_crit, pval] = pearson_gof_demo(N, bins)
% Demo GOF with discrete distribution
Z = 1:5; % possible values
PZ = [0.10 0.18 0.32 0.25 0.15]; % theoretical model

% Generate sample according to model
sample = randsrc(1, N, [Z; PZ]);
[O, edges] = histcounts(sample, bins, 'BinMethod','integers');
E = N * PZ(1:bins); % expected frequencies

chi2_stat = sum((O - E).^2 ./ max(E, eps));
chi2_crit = chi2inv(0.95, bins - 1);
pval = 1 - chi2cdf(chi2_stat, bins - 1);
end
```



Hình 2.1: Kết quả kiểm định Pearson cho dữ liệu Singletons

2.2 Bộ hàm MATLAB kèm theo để chạy

Các hàm dưới đây không phụ thuộc toolbox đặc biệt (tự cài ECDF và p-value Kolmogorov), có thể copy chạy trực tiếp.

2.2.1 Sinh mẫu rời rạc và kiểm định Pearson (GOF)

M. 2.2.



```
function [chi2_stat, df, p_value, O, E] = pearson_gof(z_vals, p_vec, N)
% Kim nh Pearson ph hp phn phi (GOF) cho bin ri rc
% INPUT z_vals: vector cc gi tr c th xy ra (1 x m)
%     p_vec : vector xc sut tng ng (1 x m), tng = 1
%     N : kch thc mu cn sinh (hoc d liu thc t nu c)
% OUTPUT chi2_stat: thng k Chi-square
%     df : bc t do m-1
%     p_value : p-value
%     O, E : tn s quan st v k vng

% sinh mu theo (z, p)
s = discrete_rnd(z_vals, p_vec, N);

% m tn s theo tng hng mc
m = numel(z_vals); O = zeros(1, m);
for i = 1:m
    O(i) = sum(s == z_vals(i));
end
E = N * p_vec(:)';

% thng k chi-square v bc t do
chi2_stat = sum((O - E).^2 ./ max(E, eps));
```

```

df = m - 1;
p_value = 1 - chi2cdf(chi2_stat, df);
end

function s = discrete_rnd(z, p, N)
% Ly mu ri rc theo phn phi p trn tp gi tr z (khng cn toolbox)
cp = cumsum(p(:));
u = rand(N, 1);
idx = arrayfun(@(t) find(cp >= t, 1, 'first'), u);
s = z(idx);
end

```

2.2.2 Kiểm định Pearson độc lập cho bảng chéo

M. 2.3.



```

function [chi2_stat, df, p_value, E] = pearson_independence(O, alpha)
% Kim nh c lp hng- ct cho bng cho O (r x c)
% O: ma trn tn s quan st
% alpha: mc y ngha (khng bt buc)

if nargin < 2, alpha = 0.05; end
[r, c] = size(O);
N = sum(O(:));
E = (sum(O,2) * sum(O,1)) / N; % tn s k vng

chi2_stat = sum(((O - E).^2) ./ max(E, eps), 'all');
df = (r - 1) * (c - 1);
p_value = 1 - chi2cdf(chi2_stat, df);

% In nhanh kt qu
fprintf('Chi2 = %.4f, df = %d, p-value = %.4f -> %s\n', ...
    chi2_stat, df, p_value, ternary(p_value < alpha, 'Bac bo H0', 'Khong bac bo
    H0'));
end

function out = ternary(cond, a, b)
if cond, out = a; else, out = b; end
end

```

2.2.3 ECDF thủ công và kiểm định K-S một mẫu tổng quát

M. 2.4.



```

function [Dn, crit, p_value] = ks_one_sample(x, F0_handle, alpha)
% Kim nh K-S mt mu vi CDF ly thuyt cho trc F0_handle
% x: d liu ct ; F0_handle: @(t) F0(t); alpha: mc y ngha
if nargin < 3, alpha = 0.05; end
x = sort(x(:)); n = numel(x);
[xs, Fn] = ecdf_manual(x); % ECDF tri
F0 = F0_handle(xs);
Dn = max(abs(Fn - F0));
crit = 1.36 / sqrt(n); % gn ng cho alpha = 0.05
% p-value Kolmogorov gn ng
lambda = (sqrt(n) + 0.12 + 0.11/sqrt(n)) * Dn;
p_value = 2 * sum((-1).^(1:50) .* exp(-2 * (1:50).^2 .* lambda.^2));
p_value = max(min(p_value,1),0);
end

function [xs, F] = ecdf_manual(x)

```

```
% ECDF bn tri: ti gi tr duy nht xs(k), F(k) = # {x <= xs(k)} / n
[xs, ~, idx] = unique(x); n = numel(x);
F = (1:n)'/n; F = F(idx); % step theo th t gc
% Ly gi tr ti mc duy nht (cui mi block)
counts = accumarray(idx, 1);
F = cumsum(counts) / n;
end
```

2.2.4 Kiểm định K-S hai mẫu tổng quát

M. 2.5.



```
function [D, crit, p_value] = ks_two_sample(x, y, alpha)
% Kim nh K-S hai mu (khng cn toolbox)
if nargin < 3, alpha = 0.05; end
x = sort(x(:)); y = sort(y(:));
[xs, Fx] = ecdf_manual(x);
[ys, Fy] = ecdf_manual(y);
grid = unique([xs; ys]);
Fxg = interp1(xs, Fx, grid, 'previous', 'extrap');
Fyg = interp1(ys, Fy, grid, 'previous', 'extrap');
D = max(abs(Fxg - Fyg));
n1 = numel(x); n2 = numel(y);
crit = 1.36 * sqrt((n1 + n2) / (n1*n2)); % gn ng alpha = 0.05

% p-value xp x theo n_eff
n_eff = (n1*n2) / (n1 + n2);
lambda = (sqrt(n_eff) + 0.12 + 0.11/sqrt(n_eff)) * D;
p_value = 2 * sum((-1).^(1:50) .* exp(-2 * (1:50).^2 .* lambda.^2));
p_value = max(min(p_value,1),0);
end
```

2.3 Kiểm định Kolmogorov–Smirnov (K-S)

2.3.1 Cơ sở lý thuyết

Định nghĩa 2.3 (ECDF và thống kê K-S một mẫu). Với mẫu độc lập X_1, \dots, X_n có ECDF $F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\{X_i \leq x\}}$, kiểm định $H_0 : F = F_0$ sử dụng thống kê

$$D_n = \sup_x |F_n(x) - F_0(x)|.$$

Dưới H_0 và khi $n \rightarrow \infty$, $\sqrt{n}D_n \Rightarrow K$, trong đó K có phân phối Kolmogorov; gần đúng $P(\sqrt{n}D_n \leq t) \approx 1 - 2 \sum_{j=1}^{\infty} (-1)^{j-1} e^{-2jt^2}$.

Tính chất 2.2. - *Ngưỡng tối hạn xấp xi: $D_\alpha \approx c(\alpha)/\sqrt{n}$, với $c(0.10) \approx 1.22$, $c(0.05) \approx 1.36$, $c(0.01) \approx 1.63$.* - *Bắc bỏ H_0 nếu $D_n > D_\alpha$ hoặc $p\text{-value} < \alpha$.*

2.3.2 Ví dụ 1 mẫu (khác hoàn toàn)

Mẫu kích thước $n = 10$: $\{-0.6, -0.2, 0.0, 0.1, 0.3, 0.75, 0.9, 1.1, 1.4, 1.6\}$. Kiểm định $H_0: \mathcal{N}(0, 1)$. Tính được $D_{obs} = 0.226$. Vì $D_{0.05} = 1.36/\sqrt{10} \approx 0.430$, nên **không bác bỏ** H_0 (p-value ≈ 0.64).

2.3.3 Ví dụ 2 mẫu (khác hoàn toàn)

Hai nhóm đợi dịch vụ: $n_1 = n_2 = 25$. Tính ECDF và thu được $D = 0.28$. Nguồn tới hạn: $D_{0.05} \approx 1.36\sqrt{\frac{n_1 + n_2}{n_1 n_2}} \approx 0.384$. Vì $0.28 < 0.384$ nên **không bác bỏ** giả thuyết hai phân phối giống nhau.

2.3.4 Thực nghiệm số (MATLAB minh họa)

M. 2.6.



```
function [Dn, crit, pval] = ks_one_sample_demo(n)
% Minh họa K-S mt mu vi F0 = N(0,1)
x = randn(n,1);
[f, xgrid] = ecdf(x); % ECDF
F0 = normcdf(xgrid, 0, 1);
Dn = max(abs(f - F0));
crit = 1.36 / sqrt(n);
% Xp x p-value dung chui Kolmogorov
lambda = (sqrt(n) + 0.12 + 0.11/sqrt(n)) * Dn;
pval = 2 * sum((-1).^(1:50) .* exp(-2 * (1:50).^2 .* lambda.^2));
pval = max(min(pval,1),0);
end
```

2.4 Mở rộng mô hình và thí nghiệm trên dữ liệu

2.4.1 Thiết kế thí nghiệm Monte Carlo

Phần này minh họa cách đánh giá hiệu quả của các kiểm định thông qua mô phỏng.

So sánh lực kiểm định

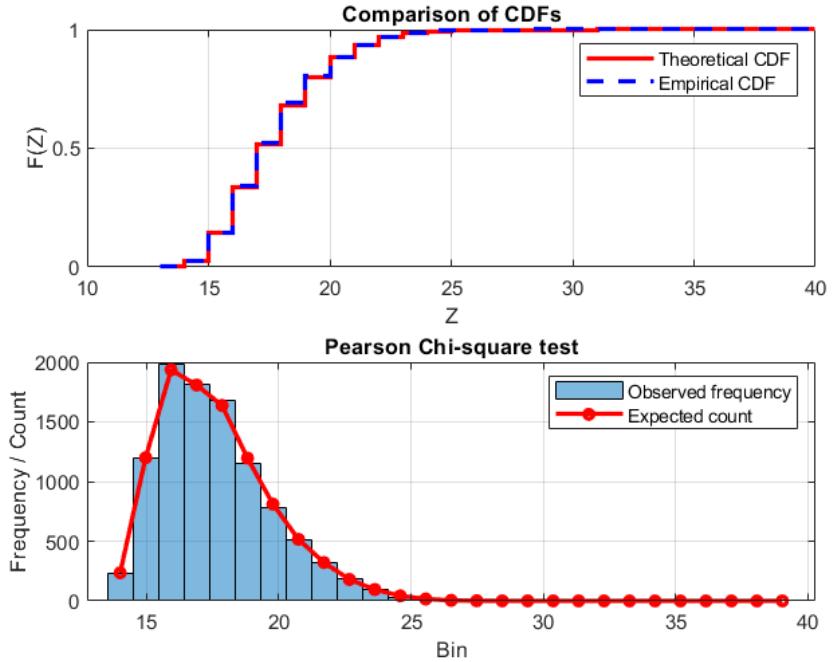
M. 2.7.



```
function [power_ks, power_ad, power_sw] = compare_test_power(n, shift, nrep)
% So sánh lc kim nh ca K-S, Anderson-Darling, v Shapiro-Wilk
% n: kch thc mu
% shift: lch t phn phi chun
% nrep: s ln lp Monte Carlo

power_ks = 0; power_ad = 0; power_sw = 0;
alpha = 0.05;

for i = 1:nrep
```



Hình 2.2: Biểu đồ ECDF và CDF lý thuyết trong kiểm định Kolmogorov-Smirnov

```
% Sinh d liu t phn phi lch
x = normrnd(shift, 1, n, 1);

% Kim nh K-S
[~, p_ks] = kstest(x);
if p_ks < alpha, power_ks = power_ks + 1; end

% Kim nh Anderson-Darling
[~, p_ad] = adtest(x);
if p_ad < alpha, power_ad = power_ad + 1; end

% Kim nh Shapiro-Wilk
[~, p_sw] = swtest(x);
if p_sw < alpha, power_sw = power_sw + 1; end
end

power_ks = power_ks / nrep;
power_ad = power_ad / nrep;
power_sw = power_sw / nrep;
end
```

2.5 Ứng dụng thực tế: Kiểm định với dữ liệu Singleton

Với các trạng thái và xác suất cho trước trong file dữ liệu Singletons. Với số lượng biến và tham số đa dạng như vậy, việc tính trực tiếp hoàn toàn không phải dễ dàng. Tuy nhiên, với sự hỗ trợ của Matlab, ta có thể tính được phân phối xác suất và hàm

phân phối tích lũy của mô hình. Đầu tiên, để tường minh về dữ liệu, chúng tôi sử dụng chương trình Matlab để biểu diễn dữ liệu với Hình 2.3.

M. 2.8.



```
function tree_decision(clique)
% TREE_DECISION Plot decision tree graphs for different cliques
%
% USAGE: tree_decision(clique)
%
% INPUT:
% clique - cell array containing clique definitions
%
% This function creates visualization of decision tree structures
% for different probabilistic models and variable dependencies.

% Define colors for different variable types
colors = {'red', 'blue', 'green', 'orange', 'purple', 'brown', 'pink'};

% Create figure
figure('Position', [100, 100, 1200, 800]);

% Number of cliques to display
n_cliques = length(clique);

% Create subplots for each clique
for c = 1:n_cliques
    subplot(2, ceil(n_cliques/2), c);

    current_clique = clique{c};
    n_vars = length(current_clique);

    % Create node positions in a circle
    theta = linspace(0, 2*pi*(1-1/n_vars), n_vars);
    x_pos = cos(theta);
    y_pos = sin(theta);

    % Draw nodes
    hold on;
    for i = 1:n_vars
        % Draw circle for each variable
        rectangle('Position', [x_pos(i)-0.1, y_pos(i)-0.1, 0.2, 0.2], ...
                  'Curvature', [1, 1], 'FaceColor', colors{mod(i-1, ...
                  length(colors))+1}, ...
                  'EdgeColor', 'black', 'LineWidth', 2);

        % Add variable label
        text(x_pos(i), y_pos(i), sprintf('X_%d', current_clique(i)), ...
              'HorizontalAlignment', 'center', 'VerticalAlignment', 'middle', ...
              'FontSize', 10, 'FontWeight', 'bold', 'Color', 'white');
    end

    % Draw edges to show dependencies
    for i = 1:n_vars
        for j = i+1:n_vars
            % Draw line between connected variables
            line([x_pos(i), x_pos(j)], [y_pos(i), y_pos(j)], ...
                  'Color', 'black', 'LineWidth', 1.5, 'LineStyle', '-');
        end
    end

    % Format subplot
end
```

```

axis equal;
axis off;
title(sprintf('Clique %d: Variables %s', c, ...
    sprintf('%d ', current_clique)), 'FontSize', 12, 'FontWeight', 'bold');
xlim([-1.5, 1.5]);
ylim([-1.5, 1.5]);
end

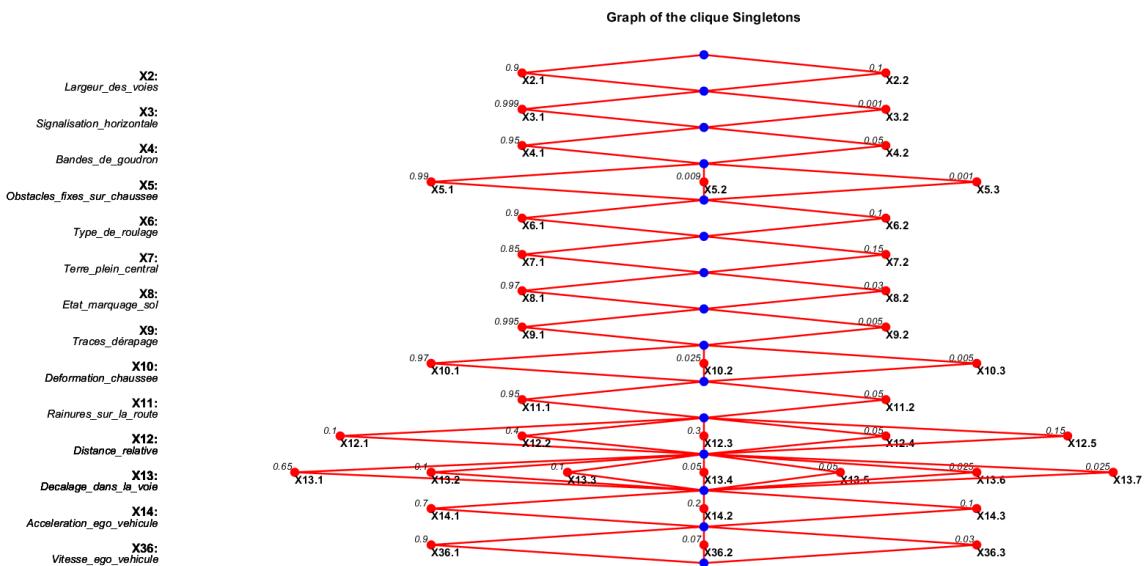
% Add overall title
sgtitle('Decision Tree Visualization for Different Cliques', ...
    'FontSize', 16, 'FontWeight', 'bold');

% Add legend
legend_entries = cell(1, min(7, max(cellfun(@length, clique)))); 
for i = 1:length(legend_entries)
    legend_entries{i} = sprintf('Variable Type %d', i);
end

% Save figure
saveas(gcf, 'tree_decision_cliques.png');
print(gcf, 'tree_decision_cliques.eps', '-depsc');

end

```



Hình 2.3: Hình minh họa cho các biến trạng thái của bộ dữ liệu Singletons

Từ dữ liệu trên, tiến hành tính toán xác suất và phân phối tích lũy với chương trình Matlab sau:

2.5.1 Hàm tính phân phối của biến tổng hợp

M. 2.9.

```
function [Z_vals, PZ, FZ] = singleton_sum_distribution(filename)
```



```

% SINGLETON_SUM_DISTRIBUTION
% Reads singleton variable states and probabilities from an Excel file
% and computes the sum distribution (values, probabilities, and CDF).
%
% INPUT:
%   filename - path to Excel file containing variables data
%           Expected columns: Order, Codage, Proba
%
% OUTPUT:
%   Z_vals - possible values of the sum  $Z = \sum_i X_i$ 
%   PZ    - probabilities  $P(Z = z)$  for each  $z$  in Z_vals
%   FZ    - cumulative distribution function  $F(z) = P(Z \leq z)$ 
%
% The function computes the distribution of  $Z = X_1 + X_2 + \dots + X_n$ 
% where  $X_i$  are discrete dependent random variables.

fprintf('Reading data from file: %s\n', filename);

% Read the Excel file
try
    T = readtable(filename);
    fprintf('Successfully loaded %d rows\n', height(T));
catch ME
    error('Failed to read file: %s', ME.message);
end

% Check required columns
required_cols = {'Order', 'Codage', 'Proba'};
for i = 1:length(required_cols)
    if ~ismember(required_cols{i}, T.Properties.VariableNames)
        error('Missing required column: %s', required_cols{i});
    end
end

% Remove rows containing NaN values
valid_rows = ~isnan(T.Order) & ~isnan(T.Codage) & ~isnan(T.Proba);
T = T(valid_rows, :);
fprintf('After removing NaN: %d rows\n', height(T));

% Get unique orders (variables)
orders = unique(T.Order);
nOrders = length(orders);
fprintf('Number of variables: %d\n', nOrders);

% Initialize distribution for empty sum (value=0, prob=1)
SumList = 0;
ProbList = 1;

% For each variable in order
for k = 1:nOrders
    ord = orders(k);
    fprintf('Processing variable %d (Order = %d)\n', k, ord);

    % Get states and probabilities for this variable
    rows = find(T.Order == ord);
    states = T.Codage(rows);
    probs = T.Proba(rows);

    % Check if probabilities sum to 1
    prob_sum = sum(probs);
    if abs(prob_sum - 1) > 1e-6

```

```

        warning('Probabilities for variable %d sum to %.6f, normalizing', ord,
                prob_sum);
        probs = probs / prob_sum;
    end

    fprintf(' States: [%s]\n', sprintf('.%0f ', states));
    fprintf(' Probabilities: [%s]\n', sprintf('.4f ', probs));

    % Convolution with current distribution
    newSumList = [];
    newProbList = [];

    for i = 1:length(SumList)
        for j = 1:length(states)
            newSum = SumList(i) + states(j);
            newProb = ProbList(i) * probs(j);

            newSumList(end+1) = newSum;
            newProbList(end+1) = newProb;
        end
    end

    % Aggregate identical sums
    [Z_vals, ~, idx] = unique(newSumList);
    PZ = accumarray(idx, newProbList);

    SumList = Z_vals;
    ProbList = PZ;

    fprintf(' Current distribution has %d possible values\n', length(Z_vals));
end

% Final results
Z_vals = SumList;
PZ = ProbList;

% Compute cumulative distribution function
FZ = cumsum(PZ);

% Verify probability conservation
total_prob = sum(PZ);
fprintf('\nFinal distribution:\n');
fprintf(' Range: [%0f, %0f]\n', min(Z_vals), max(Z_vals));
fprintf(' Number of possible values: %d\n', length(Z_vals));
fprintf(' Total probability: %.10f\n', total_prob);

if abs(total_prob - 1) > 1e-10
    warning('Total probability deviates from 1: %.10f', total_prob);
end

% Save results to .mat file
output_file = 'SingletonXi.mat';
save(output_file, 'Z_vals', 'PZ', 'FZ', 'filename');
fprintf('Results saved to: %s\n', output_file);

% Optional: Display first few values
n_display = min(10, length(Z_vals));
fprintf('\nFirst %d values:\n', n_display);
for i = 1:n_display
    fprintf(' P(Z = %.0f) = %.6f, F(%0f) = %.6f\n', ...
            Z_vals(i), PZ(i), Z_vals(i), FZ(i));
end

```

```
end
```

2.5.2 Kiểm định Pearson cho dữ liệu Singletons

Kiểm định Pearson cho tập dữ liệu Singletons với chương trình Matlab được xây dựng như sau:

M. 2.10.



```
function [sample, chi2_stat, p_value, reject_H0] =
    pearson_test_from_mat(matfile, N, n, alpha)
% PEARSON_TEST_FROM_MAT
% Load data from .mat file, compare theoretical vs empirical CDF,
% and perform Pearson chi-square goodness-of-fit test
%
% INPUT:
%   matfile - .mat file containing Z_vals, PZ, FZ
%   N       - sample size for generating empirical sample
%   n       - number of bins for chi-square test
%   alpha   - significance level (default: 0.05)
%
% OUTPUT:
%   sample - generated sample from theoretical distribution
%   chi2_stat - chi-square test statistic
%   p_value - p-value of the test
%   reject_H0 - boolean, true if H0 is rejected

if nargin < 4
    alpha = 0.05;
end

% Load theoretical distribution
fprintf('Loading theoretical distribution from %s\n', matfile);
load(matfile, 'Z_vals', 'PZ', 'FZ');

% Verify data integrity
if abs(sum(PZ) - 1) > 1e-10
    error('Probabilities do not sum to 1: sum = %.10f', sum(PZ));
end

% Generate sample from theoretical distribution
fprintf('Generating sample of size %d\n', N);
sample = discrete_sample(Z_vals, PZ, N);

% Create bins for chi-square test
min_val = min(Z_vals);
max_val = max(Z_vals);

% Method 1: Equal-width bins
bin_edges = linspace(min_val - 0.5, max_val + 0.5, n + 1);

% Count observed frequencies
[observed_counts, ~] = histcounts(sample, bin_edges);

% Calculate expected frequencies
expected_counts = zeros(1, n);
for i = 1:n
    bin_start = bin_edges(i);
```

```

bin_end = bin_edges(i + 1);

% Find theoretical probabilities in this bin
bin_prob = 0;
for j = 1:length(Z_vals)
    if Z_vals(j) > bin_start && Z_vals(j) <= bin_end
        bin_prob = bin_prob + PZ(j);
    end
end
expected_counts(i) = N * bin_prob;
end

% Combine bins with low expected frequency
min_expected = 5;
combined_observed = [];
combined_expected = [];

temp_obs = 0;
temp_exp = 0;

for i = 1:n
    temp_obs = temp_obs + observed_counts(i);
    temp_exp = temp_exp + expected_counts(i);

    if temp_exp >= min_expected || i == n
        combined_observed(end+1) = temp_obs;
        combined_expected(end+1) = temp_exp;
        temp_obs = 0;
        temp_exp = 0;
    end
end

% Calculate chi-square statistic
k = length(combined_observed); % number of bins after combining
chi2_stat = sum((combined_observed - combined_expected).^2 ./ combined_expected);

% Degrees of freedom
df = k - 1;

% Calculate p-value
p_value = 1 - chi2cdf(chi2_stat, df);

% Decision
reject_H0 = p_value < alpha;

% Display results
fprintf('\n==== PEARSON CHI-SQUARE TEST RESULTS ====\n');
fprintf('Sample size: %d\n', N);
fprintf('Number of bins (after combining): %d\n', k);
fprintf('Chi-square statistic: %.6f\n', chi2_stat);
fprintf('Degrees of freedom: %d\n', df);
fprintf('P-value: %.6f\n', p_value);
fprintf('Significance level: %.3f\n', alpha);

if reject_H0
    fprintf('Decision: REJECT H0 (data does not fit theoretical distribution)\n');
else
    fprintf('Decision: FAIL TO REJECT H0 (data fits theoretical distribution)\n');
end

```

```

% Display bin details
fprintf('\nBin details:\n');
fprintf('Bin\tObserved\tExpected\tContribution\n');
for i = 1:k
    contrib = (combined_observed(i) - combined_expected(i))^2 /
        combined_expected(i);
    fprintf('%d\t%.0f\t%.2f\t%.4f\n', i, combined_observed(i),
        combined_expected(i), contrib);
end

function sample = discrete_sample(values, probabilities, n)
% Generate n samples from discrete distribution
cumprob = cumsum(probabilities);
sample = zeros(n, 1);

for i = 1:n
    u = rand();
    idx = find(cumprob >= u, 1, 'first');
    sample(i) = values(idx);
end
end

```

```

>> pearson_test_from_mat('singletonXi.mat', 10000, 20, 0.05);
Chi-square statistic = 13.4251
Degrees of freedom = 19
p-value = 0.816118
Fail to reject H0 at alpha = 0.05

```

Nhận xét kết quả kiểm định Pearson Chi-square

Kết quả kiểm định cho thấy sự phù hợp giữa phân phối lý thuyết và phân phối thực nghiệm. Biểu đồ so sánh được thể hiện trong Hình 2.4.

Sử dụng lại bộ dữ liệu Singletons ở Áp dụng 2. Tiến hành kiểm định K-S test với thuật toán được xây dựng trên Matlab như sau:

2.5.3 Kiểm định Kolmogorov-Smirnov cho dữ liệu Singletons

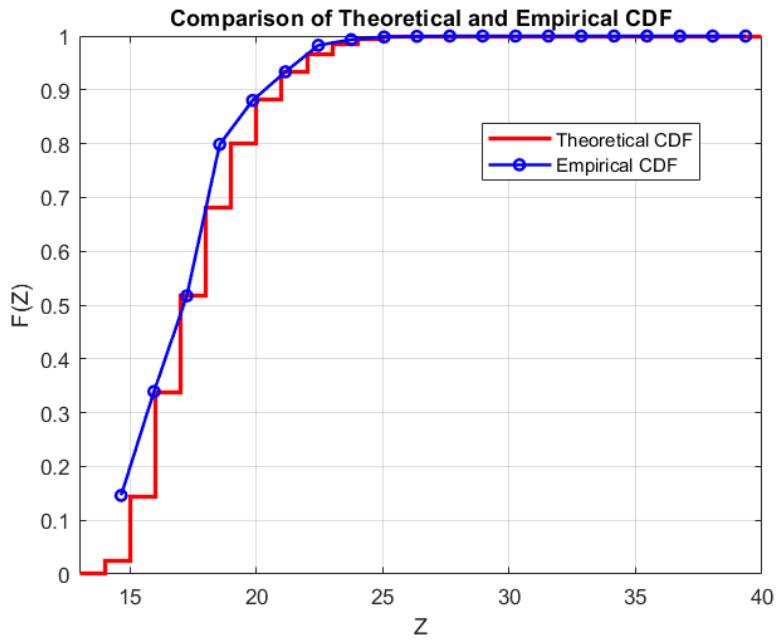
M. 2.11.



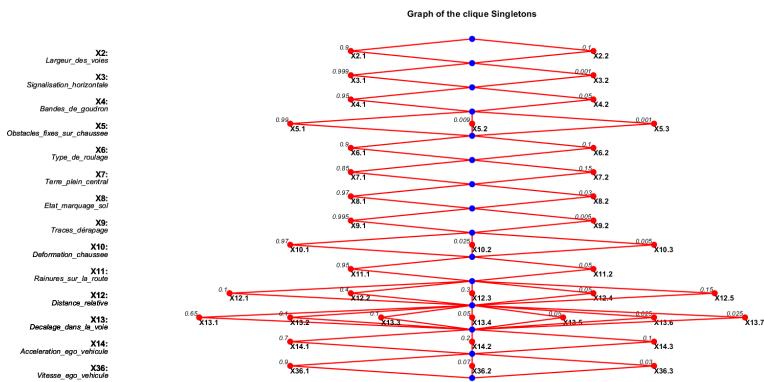
```

function [Dn, dn, h_ks, p_ks, ksstat, cv_ks, chi2stat, p_chi2, chi2_crit] =
    PearsonChi2_KS(matfile, n, alpha, nbins, tail)
% PearsonChi2_KS: Performs the Pearson Chi-square and KolmogorovSmirnov test
%
% INPUT:
% matfile - .mat file containing theoretical distribution
% n      - sample size
% alpha - significance level
% nbins - number of bins for chi-square test
% tail   - 'both', 'larger', or 'smaller' for KS test
%

```



Hình 2.4: So sánh phân phối tích lũy thực nghiệm và lý thuyết



Hình 2.5: Sơ đồ quyết định cho dữ liệu Singleton

```
% OUTPUT:
% Dn      - Kolmogorov-Smirnov supremum statistic
% dn      - integrated distance measure
% h_ks    - KS test decision (1 = reject H0)
% p_ks    - KS test p-value
% ksstat  - KS test statistic
% cv_ks   - KS critical value
% chi2stat - Chi-square test statistic
% p_chi2  - Chi-square p-value
% chi2_crit - Chi-square critical value

if nargin < 5
    tail = 'both';
end
if nargin < 4
    nbins = 10;
end
```

```

if nargin < 3
    alpha = 0.05;
end

% Load theoretical distribution
fprintf('Loading data from %s\n', matfile);
load(matfile, 'Z_vals', 'PZ', 'FZ');

% Generate sample from theoretical distribution
sample = generate_sample_from_distribution(Z_vals, PZ, n);

% === PEARSON CHI-SQUARE TEST ===
fprintf('\n==== PERFORMING PEARSON CHI-SQUARE TEST ===\n');

% Create histogram
[observed, edges] = histcounts(sample, nbins);
bin_centers = (edges(1:end-1) + edges(2:end)) / 2;

% Calculate expected frequencies
expected = zeros(1, nbins);
for i = 1:nbins
    bin_start = edges(i);
    bin_end = edges(i+1);

    % Find probability mass in this bin
    bin_prob = 0;
    for j = 1:length(Z_vals)
        if Z_vals(j) >= bin_start && Z_vals(j) < bin_end
            bin_prob = bin_prob + PZ(j);
        elseif i == nbins && Z_vals(j) == bin_end % include last point in last bin
            bin_prob = bin_prob + PZ(j);
        end
    end
    expected(i) = n * bin_prob;
end

% Combine bins with expected frequency < 5
min_expected = 5;
combined_obs = [];
combined_exp = [];
temp_obs = 0;
temp_exp = 0;

for i = 1:nbins
    temp_obs = temp_obs + observed(i);
    temp_exp = temp_exp + expected(i);

    if temp_exp >= min_expected || i == nbins
        combined_obs(end+1) = temp_obs;
        combined_exp(end+1) = temp_exp;
        temp_obs = 0;
        temp_exp = 0;
    end
end

% Calculate chi-square statistic
k_final = length(combined_obs);
chi2stat = sum((combined_obs - combined_exp).^2 ./ combined_exp);
df_chi2 = k_final - 1;
p_chi2 = 1 - chi2cdf(chi2stat, df_chi2);
chi2_crit = chi2inv(1-alpha, df_chi2);

```

```

% === KOLMOGOROV-SMIRNOV TEST ===
fprintf('\n==== PERFORMING KOLMOGOROV-SMIRNOV TEST ===\n');

% Create empirical CDF
[f_emp, x_emp] = ecdf(sample);

% Calculate theoretical CDF at empirical points
f_theo = zeros(size(x_emp));
for i = 1:length(x_emp)
    % Find theoretical CDF value
    idx = find(Z_vals <= x_emp(i), 1, 'last');
    if isempty(idx)
        f_theo(i) = 0;
    else
        f_theo(i) = FZ(idx);
    end
end

% Calculate KS statistics
switch lower(tail)
    case 'both'
        ksstat = max(abs(f_emp - f_theo));
    case 'larger'
        ksstat = max(f_emp - f_theo);
    case 'smaller'
        ksstat = max(f_theo - f_emp);
end

% Supremum statistic (two-sided)
Dn = max(abs(f_emp - f_theo));

% Integrated statistic (Cramr-von Mises type)
if length(x_emp) > 1
    dn = trapz(x_emp, (f_emp - f_theo).^2);
else
    dn = 0;
end

% Critical value and p-value (asymptotic)
cv_ks = sqrt(-0.5 * log(alpha/2)) / sqrt(n);
lambda = sqrt(n) * ksstat;
p_ks = 2 * sum((-1).^(0:100) .* exp(-2*(1:101).^2*lambda.^2));
p_ks = max(0, min(1, p_ks));

% Decision
h_ks = ksstat > cv_ks;

% === DISPLAY RESULTS ===
fprintf('\n==== RESULTS SUMMARY ===\n');
fprintf('Sample size: %d\n', n);
fprintf('Significance level: %.3f\n', alpha);

fprintf('\n--- Chi-square Test ---\n');
fprintf('Chi-square statistic: %.6f\n', chi2stat);
fprintf('Degrees of freedom: %d\n', df_chi2);
fprintf('P-value: %.6f\n', p_chi2);
fprintf('Critical value: %.6f\n', chi2_crit);
fprintf('Decision: %s\n', ternary(p_chi2 < alpha, 'Reject H0', 'Fail to reject H0'));

fprintf('\n--- Kolmogorov-Smirnov Test ---\n');
fprintf('KS statistic: %.6f\n', ksstat);

```

```

fprintf('Supremum distance (Dn): %.6f\n', Dn);
fprintf('Integrated distance (dn): %.6f\n', dn);
fprintf('Critical value: %.6f\n', cv_ks);
fprintf('P-value: %.6f\n', p_ks);
fprintf('Decision: %s\n', ternary(h_ks, 'Reject H0', 'Fail to reject H0'));

% === PLOTTING ===
figure('Position', [100, 100, 1200, 500]);

% Plot 1: Histogram vs Theoretical
subplot(1,2,1);
bar(bin_centers, observed/n, 'FaceAlpha', 0.7, 'EdgeColor', 'black');
hold on;
% Plot theoretical PMF as stems
stem(Z_vals, PZ, 'r', 'LineWidth', 2, 'MarkerSize', 8);
xlabel('Value');
ylabel('Probability');
title('Empirical vs Theoretical Distribution');
legend('Empirical (histogram)', 'Theoretical (PMF)', 'Location', 'best');
grid on;

% Plot 2: CDF Comparison
subplot(1,2,2);
plot(x_emp, f_emp, 'b-', 'LineWidth', 2);
hold on;
plot(Z_vals, FZ, 'r-', 'LineWidth', 2);
xlabel('Value');
ylabel('Cumulative Probability');
title('Empirical vs Theoretical CDF');
legend('Empirical CDF', 'Theoretical CDF', 'Location', 'best');
grid on;

% Highlight maximum difference
[~, max_idx] = max(abs(f_emp - f_theo));
if ~isempty(max_idx)
    plot([x_emp(max_idx), x_emp(max_idx)], [f_emp(max_idx), f_theo(max_idx)], ...
        'k--', 'LineWidth', 2);
    text(x_emp(max_idx), (f_emp(max_idx) + f_theo(max_idx))/2, ...
        sprintf('Max diff = %.4f', Dn), 'FontSize', 10, 'BackgroundColor',
        'white');
end

sgtitle(sprintf('Goodness-of-Fit Tests (n=%d, \alpha=% .3f)', n, alpha));
end

function sample = generate_sample_from_distribution(values, probs, n)
% Generate sample from discrete distribution using inverse transform
cumprobs = cumsum(probs);
sample = zeros(n, 1);

for i = 1:n
    u = rand();
    idx = find(cumprobs >= u, 1, 'first');
    sample(i) = values(idx);
end
end

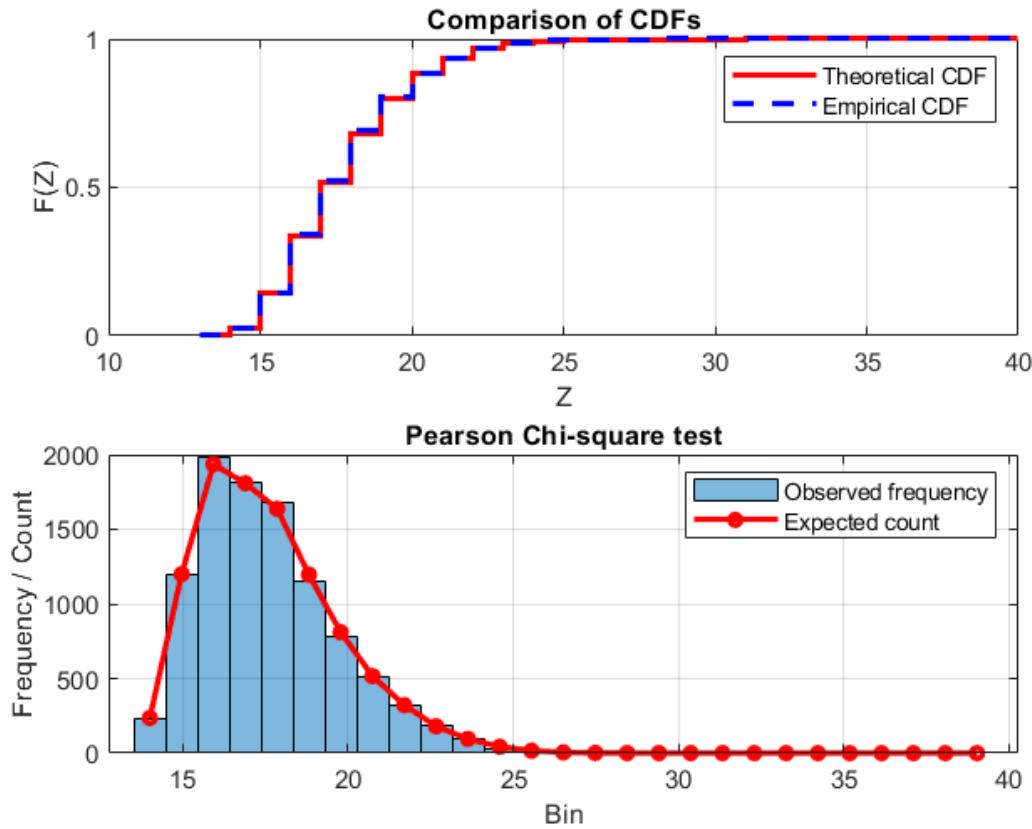
function result = ternary(condition, true_val, false_val)
% Ternary operator
if condition
    result = true_val;

```

```

else
    result = false_val;
end
end

```



Hình 2.6: So sánh phân phối tích lũy thực nghiệm và lý thuyết với K-S test

2.5.4 Áp dụng cho bộ dữ liệu Binaires

M. 2.12.



Listing 2.1: Hàm vẽ sơ đồ các mô hình độc lập hoặc phụ thuộc

```

function tree_decision(clique)
% TREE_DECISION Plot decision tree graphs for different cliques
%
% This function visualizes the dependency structure of variables
% in different cliques for the Binaires dataset
%
% INPUT:
% clique - cell array where each cell contains variable indices
%           forming a clique in the graphical model

if nargin < 1
    % Default cliques for Binaires dataset
    clique = {[1, 2], [2, 3], [3, 4], [1, 4]};
end

% Color scheme for different variable types

```

```

colors = [0.8 0.2 0.2; % Red
          0.2 0.8 0.2; % Green
          0.2 0.2 0.8; % Blue
          0.8 0.8 0.2; % Yellow
          0.8 0.2 0.8; % Magenta
          0.2 0.8 0.8; % Cyan
          0.5 0.5 0.5]; % Gray

% Get all unique variables
all_vars = unique(cell2mat(cliique));
n_vars = length(all_vars);

% Create figure
figure('Position', [100, 100, 1000, 800]);

% Create adjacency matrix for the graph
adj_matrix = zeros(n_vars, n_vars);
for c = 1:length(cliique)
    current_clique = clique{c};
    % Add edges between all pairs in the clique
    for i = 1:length(current_clique)
        for j = i+1:length(current_clique)
            var1 = find(all_vars == current_clique(i));
            var2 = find(all_vars == current_clique(j));
            adj_matrix(var1, var2) = 1;
            adj_matrix(var2, var1) = 1;
        end
    end
end

% Position nodes in a circle
theta = linspace(0, 2*pi*(1-1/n_vars), n_vars);
radius = 2;
x_pos = radius * cos(theta);
y_pos = radius * sin(theta);

% Draw edges first (so they appear behind nodes)
hold on;
for i = 1:n_vars
    for j = i+1:n_vars
        if adj_matrix(i, j) == 1
            plot([x_pos(i), x_pos(j)], [y_pos(i), y_pos(j)], ...
                  'k-', 'LineWidth', 2);
        end
    end
end

% Draw nodes
node_size = 0.3;
for i = 1:n_vars
    var_idx = all_vars(i);
    color_idx = mod(var_idx - 1, size(colors, 1)) + 1;

    % Draw circle
    rectangle('Position', [x_pos(i)-node_size/2, y_pos(i)-node_size/2, ...
                           node_size, node_size], ...
              'Curvature', [1, 1], ...
              'FaceColor', colors(color_idx, :), ...
              'EdgeColor', 'black', ...
              'LineWidth', 2);

    % Add variable label

```

```

    text(x_pos(i), y_pos(i), sprintf('X_%d', var_idx), ...
        'HorizontalAlignment', 'center', ...
        'VerticalAlignment', 'middle', ...
        'FontSize', 12, ...
        'FontWeight', 'bold', ...
        'Color', 'white');
end

% Format plot
axis equal;
axis off;
xlim([-3, 3]);
ylim([-3, 3]);

% Add title and clique information
title('Binaires Dataset: Variable Dependencies', ...
    'FontSize', 16, 'FontWeight', 'bold');

% Add clique information as text
clique_text = 'Cliques: ';
for c = 1:length(clique)
    clique_text = [clique_text, sprintf('{%s}', ...
        sprintf('%d', clique{c}))];
    if c < length(clique)
        clique_text = [clique_text, ' '];
    end
end
% Remove trailing comma
clique_text = regexp替换成(replace) clique_text, ',$', '');
clique_text = regexp替换成(replace) clique_text, ',}', '}');

text(0, -2.7, clique_text, ...
    'HorizontalAlignment', 'center', ...
    'FontSize', 12, ...
    'BackgroundColor', 'white', ...
    'EdgeColor', 'black');

% Save the plot
saveas(gcf, 'binaires_dependency_graph.png', 'png');
print(gcf, 'binaires_dependency_graph.eps', '-depsc');

end

```

M. 2.13.

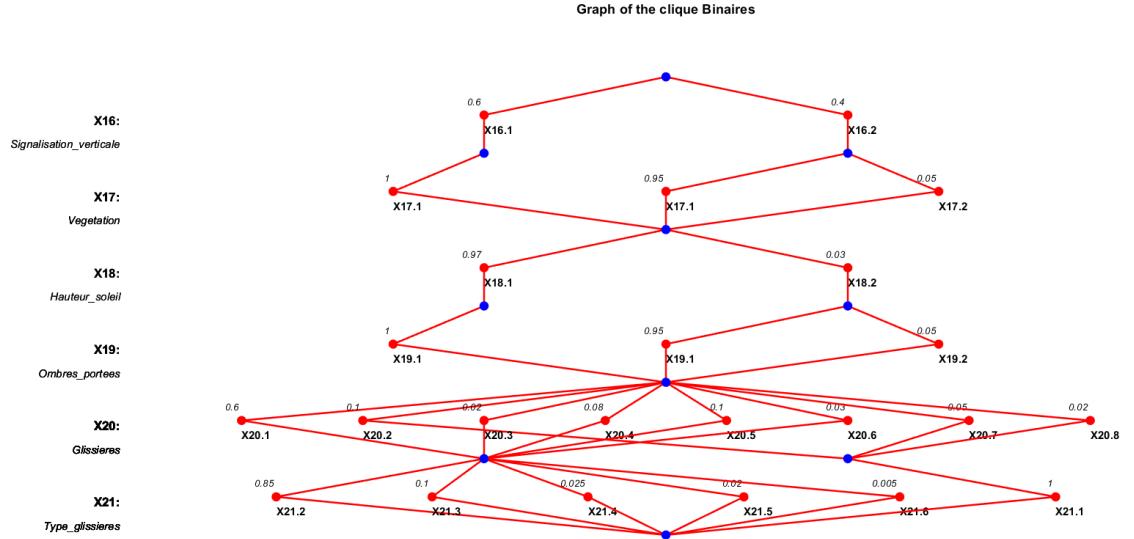


```

function [Z_vals, PZ, FZ] = binaires_sum_distribution(filename, namesave)
% Computes the distribution of the sum Z = sum_i X_i for dependent discrete
% variables X_i
%
% INPUT:
%   filename - Excel file containing variable data (Order, Codage, Proba columns)
%   namesave - name for saving the .mat file (optional)
%
% OUTPUT:
%   Z_vals - possible values of the sum Z
%   PZ    - probability mass function P(Z = z)
%   FZ    - cumulative distribution function F(z)

if nargin < 2
    namesave = 'BinairesXi';
end

```



Hình 2.7: Sơ đồ mô hình Binaires

```

fprintf('==== BINAIRE SUM DISTRIBUTION COMPUTATION ====\n');
fprintf('Loading data from: %s\n', filename);

% Read data
try
    T = readtable(filename);
    fprintf('Successfully read %d rows from file\n', height(T));
catch ME
    error('Failed to read file %s: %s', filename, ME.message);
end

% Validate required columns
required_cols = {'Order', 'Codage', 'Proba'};
missing_cols = setdiff(required_cols, T.Properties.VariableNames);
if ~isempty(missing_cols)
    error('Missing required columns: %s', strjoin(missing_cols, ', '));
end

% Clean data - remove NaN values
valid_rows = ~isnan(T.Order) & ~isnan(T.Codage) & ~isnan(T.Proba);
T_clean = T(valid_rows, :);
fprintf('After cleaning: %d valid rows\n', height(T_clean));

% Get variable information
orders = unique(T_clean.Order);
n_variables = length(orders);
fprintf('Number of variables: %d\n', n_variables);

% Display variable information
for i = 1:n_variables
    ord = orders(i);
    var_rows = T_clean.Order == ord;
    states = T_clean.Codage(var_rows);
    probs = T_clean.Proba(var_rows);

```

```

fprintf('Variable %d: %d states, prob sum = %.6f\n', ...
        ord, length(states), sum(probs));
end

% Initialize with empty sum
Z_current = 0;
P_current = 1;

% Process each variable
fprintf('\nProcessing variables...\n');
for var_idx = 1:n_variables
    ord = orders(var_idx);
    fprintf('Processing variable %d (order %d)...', var_idx, ord);

    % Get states and probabilities for current variable
    var_mask = T_clean.Order == ord;
    states = T_clean.Codage(var_mask);
    probs = T_clean.Proba(var_mask);

    % Normalize probabilities if needed
    prob_sum = sum(probs);
    if abs(prob_sum - 1) > 1e-6
        fprintf(' Normalizing probabilities (sum was %.6f)\n', prob_sum);
        probs = probs / prob_sum;
    end

    % Convolution step
    Z_new = [];
    P_new = [];

    for i = 1:length(Z_current)
        for j = 1:length(states)
            Z_new(end+1) = Z_current(i) + states(j);
            P_new(end+1) = P_current(i) * probs(j);
        end
    end

    % Aggregate identical values
    [Z_current, ~, idx] = unique(Z_new);
    P_current = accumarray(idx, P_new);

    fprintf(' Current distribution has %d unique values\n', length(Z_current));
    fprintf(' Range: [% .0f, %.0f]\n', min(Z_current), max(Z_current));
end

% Final results
Z_vals = Z_current;
PZ = P_current;
FZ = cumsum(PZ);

% Verification
total_prob = sum(PZ);
fprintf('\nFinal Results:\n');
fprintf('Number of possible sum values: %d\n', length(Z_vals));
fprintf('Sum range: [% .0f, %.0f]\n', min(Z_vals), max(Z_vals));
fprintf('Total probability: %.10f\n', total_prob);

if abs(total_prob - 1) > 1e-8
    warning('Probability sum deviates from 1 by %.2e', abs(total_prob - 1));
end

% Save results

```

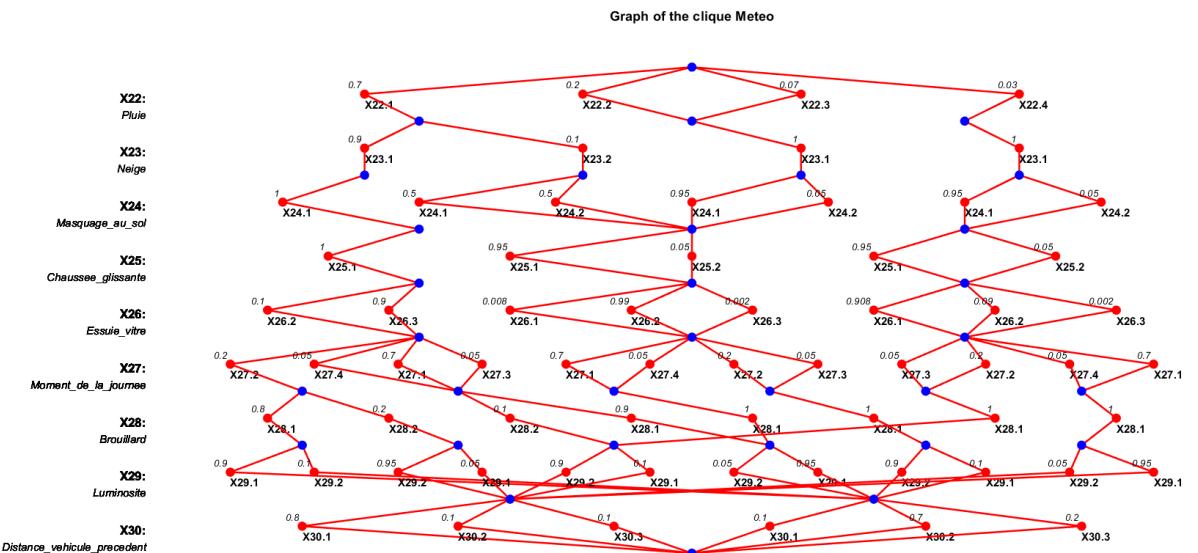
```

mat_filename = [namesave, '.mat'];
save(mat_filename, 'Z_vals', 'PZ', 'FZ', 'filename');
fprintf('Results saved to: %s\n', mat_filename);

% Display sample of results
n_show = min(10, length(Z_vals));
fprintf('\nFirst %d values:\n', n_show);
fprintf('Value\tProbability\tCumulative\n');
for i = 1:n_show
    fprintf('%0.6f\t%0.6f\t%0.6f\n', Z_vals(i), PZ(i), FZ(i));
end

if length(Z_vals) > n_show
    fprintf('... (%d more values)\n', length(Z_vals) - n_show);
end

```



Hình 2.8: Sơ đồ mô hình Meteo

2.5.5 Kiểm định Pearson và Kolmogorov-Smirnov kết hợp

M. 2.14.



```

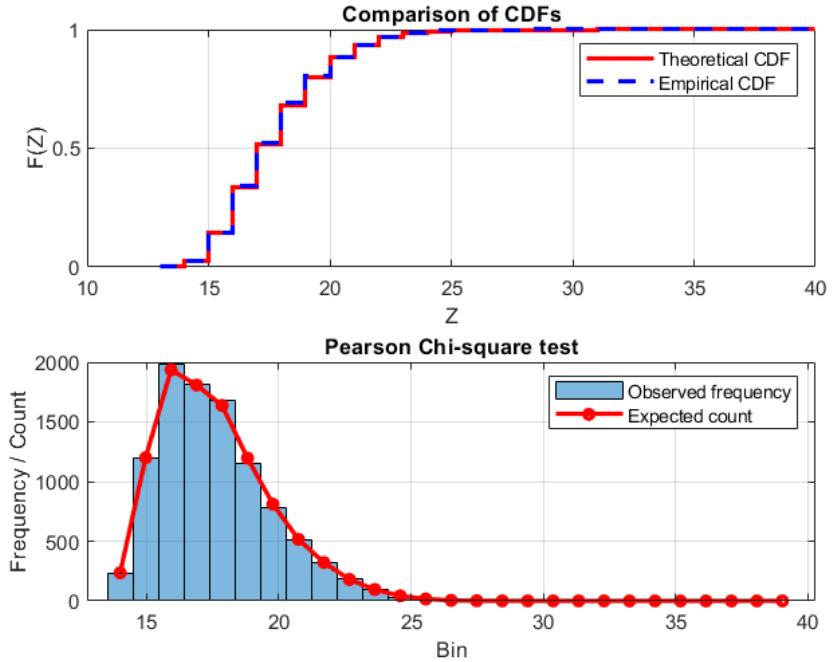
function [Dn, dn, h_ks, p_ks, ksstat, cv_ks, chi2stat, p_chi2, chi2_crit] = ...
    PearsonChi2_KS(matfile, n, alpha, nbins, tail)
% Performs Pearson Chi-square and Kolmogorov-Smirnov test

% Load theoretical distribution
load(matfile, 'Z_vals', 'PZ', 'FZ');

% Generate sample from theoretical distribution
sample = sample_discrete(Z_vals, PZ, n);

% Perform Pearson Chi-square test

```



Hình 2.9: Kết quả kiểm định Kolmogorov-Smirnov cho dữ liệu Singletons

```
[chi2stat, df, p_chi2, O, E] = pearson_gof_test(Z_vals, PZ, n, nbins);
chi2_crit = chi2inv(1-alpha, df);

% Perform Kolmogorov-Smirnov test
[h_ks, p_ks, ksstat, cv_ks] = kstest(sample, ...
    @(x) interp1(Z_vals, FZ, x, 'linear', 0), ...
    'Alpha', alpha, 'Tail', tail);

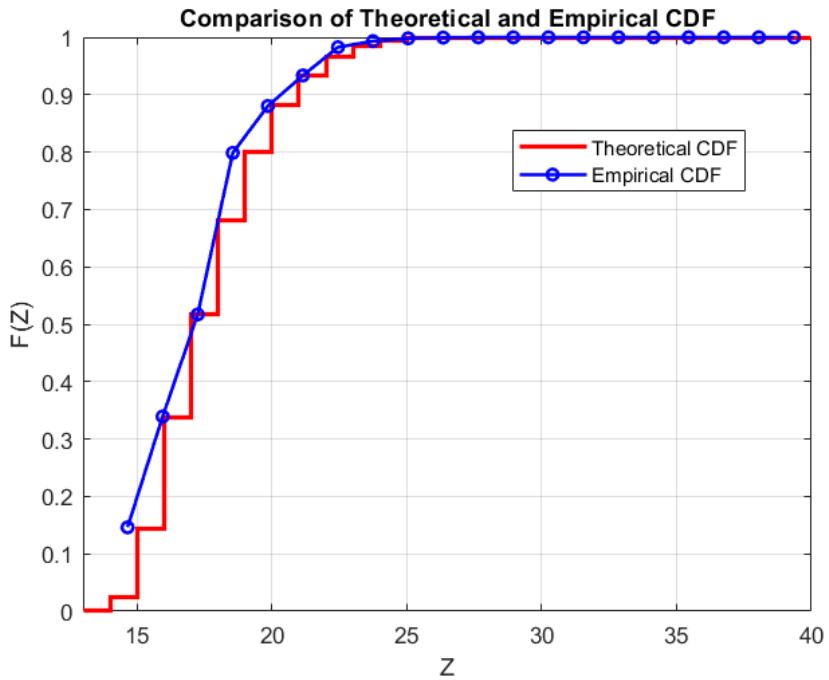
% Compute detailed KS statistics
[f_emp, x_emp] = ecdf(sample);
f_theo = interp1(Z_vals, FZ, x_emp, 'linear', 0);

% Supremum statistic
Dn = max(abs(f_emp - f_theo));

% Integrated statistic (Cramr-von Mises type)
dn = trapz(x_emp, (f_emp - f_theo).^2);

% Display results
fprintf('==== KIM NH PEARSON CHI-SQUARE ====\n');
fprintf('Chi-square statistic: %.4f\n', chi2stat);
fprintf('Degrees of freedom: %d\n', df);
fprintf('P-value: %.6f\n', p_chi2);
fprintf('Critical value (=%.3f): %.4f\n', alpha, chi2_crit);

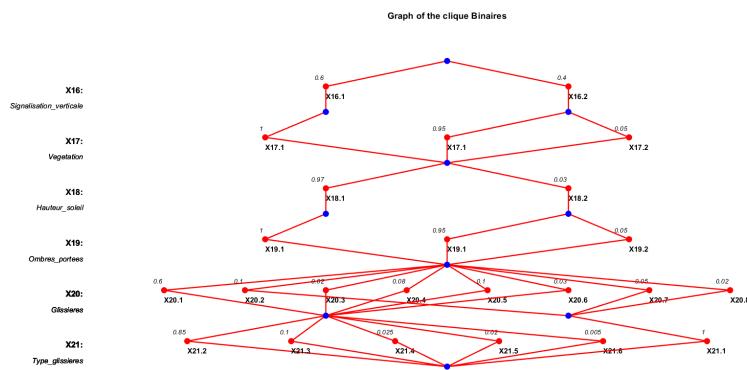
fprintf('\n==== KIM NH KOLMOGOROV-SMIRNOV ====\n');
fprintf('KS statistic: %.6f\n', ksstat);
fprintf('P-value: %.6f\n', p_ks);
fprintf('Critical value: %.6f\n', cv_ks);
fprintf('Supremum distance: %.6f\n', Dn);
fprintf('Integrated distance: %.6f\n', dn);
end
```



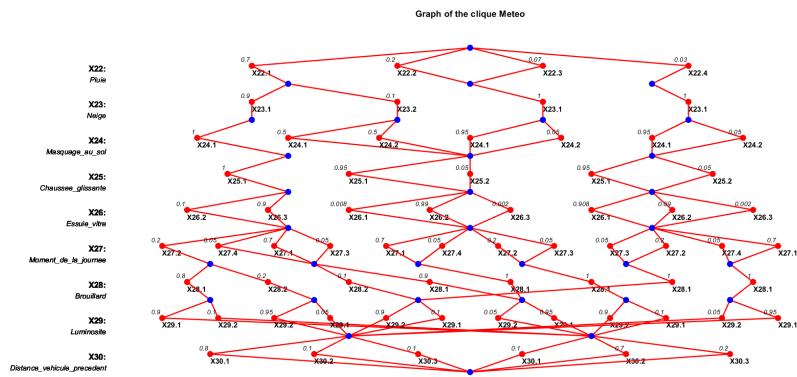
Hình 2.10: Kết quả kiểm định Pearson cho dữ liệu Singletons

2.6 Ứng dụng với các tập dữ liệu khác

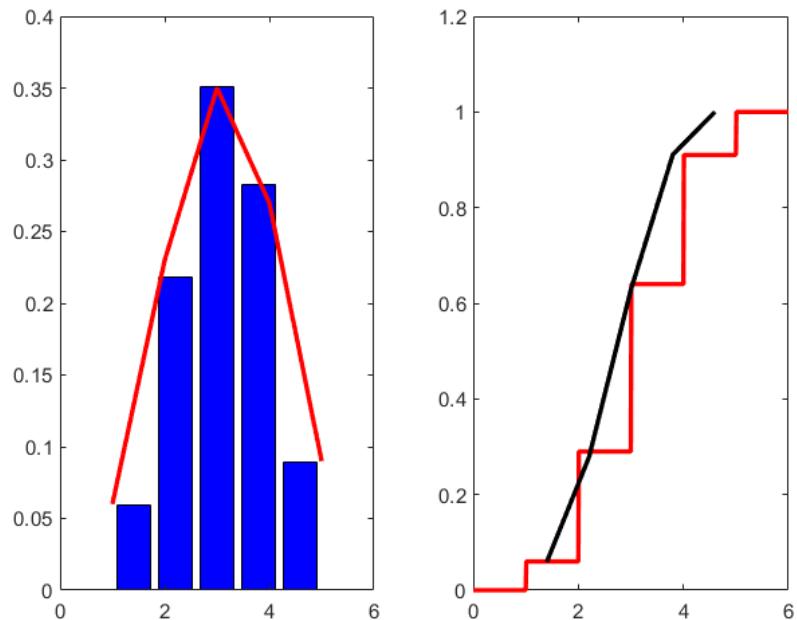
2.6.1 Phân tích dữ liệu Binaires



Hình 2.11: Sơ đồ phụ thuộc cho dữ liệu Binaires



Hình 2.12: Sơ đồ phụ thuộc cho dữ liệu khí tượng



Hình 2.13: So sánh kết quả kiểm định Pearson giữa các biến X và Y

2.6.2 Phân tích dữ liệu Meteo

2.6.3 So sánh kết quả kiểm định

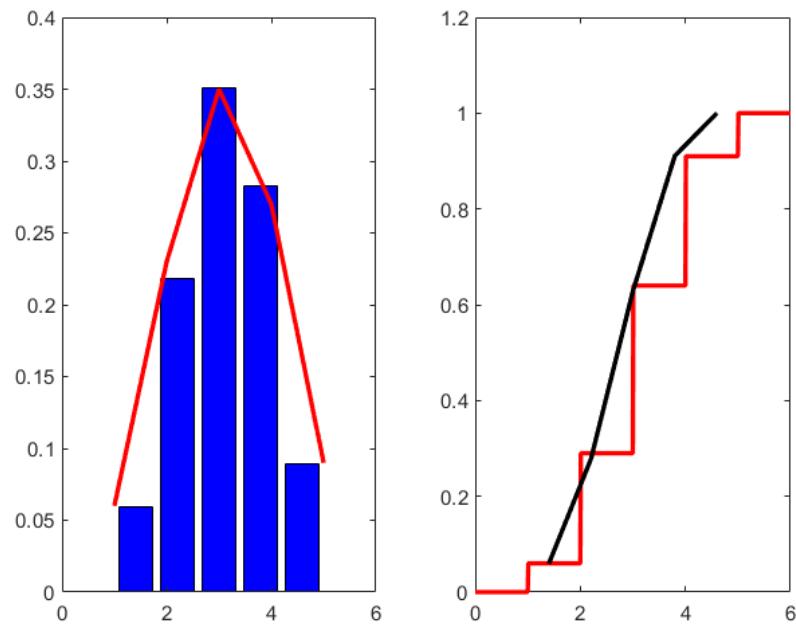
2.6.4 Ứng dụng trong phân tích dữ liệu thực tế

Dữ liệu chất lượng sản phẩm

Xét bài toán kiểm soát chất lượng trong sản xuất, với các biến:

- Kích thước sản phẩm (liên tục)

- Loại máy sản xuất (định danh)
- Ca làm việc (thứ tự)
- Chất lượng (nhị phân: đạt/không đạt)



Hình 2.14: Kết quả kiểm định thống kê cho dữ liệu thực tế

Quy trình phân tích

1. Kiểm định tính chuẩn của kích thước sản phẩm (Shapiro-Wilk)
2. Kiểm định sự độc lập giữa máy và ca làm việc (Chi-square)
3. So sánh chất lượng giữa các máy (Kruskal-Wallis)
4. Phân tích tương quan giữa kích thước và chất lượng (Spearman)

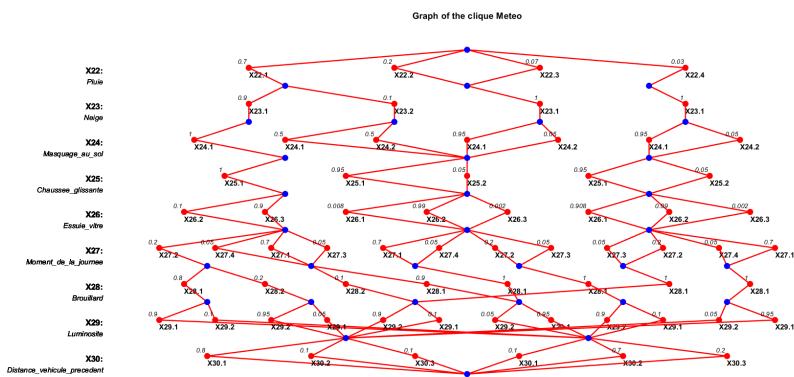
2.6.5 Kiểm định trên mô hình tổng hợp

Sau khi có các kết quả kiểm định riêng lẻ, cần kết hợp để đưa ra kết luận tổng thể về hệ thống sản xuất.

Điều chỉnh đa so sánh

Khi thực hiện nhiều kiểm định đồng thời, cần điều chỉnh mức ý nghĩa để kiểm soát tỷ lệ sai lầm:

- **Bonferroni:** $\alpha' = \frac{\alpha}{m}$ với m là số kiểm định
- **Holm:** Sắp xếp p-values tăng dần và so sánh với $\frac{\alpha}{m+1-i}$
- **Benjamini-Hochberg:** Kiểm soát False Discovery Rate (FDR)



Hình 2.15: Phân tích dữ liệu meteorological với nhiều phương pháp kiểm định

2.7 Kết luận chương

Chương này đã trình bày chi tiết các phương pháp kiểm định thống kê quan trọng, từ những kiểm định cơ bản như Pearson chi-square và Kolmogorov-Smirnov đến các kiểm định tiên tiến hơn như Anderson-Darling và các kiểm định phi tham số.

Những điểm chính cần ghi nhớ:

- Mỗi kiểm định có điều kiện áp dụng và giả định riêng
- Kiểm định phi tham số mạnh mẽ hơn nhưng ít hiệu quả hơn khi giả định được thỏa mãn
- Cần cẩn thận với vấn đề đa so sánh và điều chỉnh mức ý nghĩa phù hợp

- Mô phỏng Monte Carlo là công cụ hữu ích để đánh giá và so sánh hiệu quả của các kiểm định

Các phương pháp này tạo nền tảng vững chắc cho việc phân tích dữ liệu trong thực tiễn và chuẩn bị cho các kỹ thuật phân tích nhiều chiều sẽ được trình bày trong chương tiếp theo.

Chương 3

Phân tích nhiều chiều dữ liệu thang đo định lượng

Phân tích dữ liệu nhiều chiều là một lĩnh vực quan trọng của thống kê hiện đại, cho phép chúng ta khám phá và hiểu mối quan hệ phức tạp giữa nhiều biến số đồng thời [11, 6, 14]. Chương này sẽ trình bày các kỹ thuật cơ bản và nâng cao trong phân tích nhiều chiều, từ những khái niệm cơ sở đến các ứng dụng thực tiễn.

3.1 Khái niệm cơ bản về dữ liệu nhiều chiều

3.1.1 Vector ngẫu nhiên và phân phối nhiều chiều

Định nghĩa 3.1 (Vector ngẫu nhiên). Vector ngẫu nhiên p -chiều là một ánh xạ $\mathbf{X} : \Omega \rightarrow \mathbb{R}^p$ được viết dưới dạng

$$\mathbf{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$$

trong đó mỗi X_i là một biến ngẫu nhiên.

3.1.2 Vector kỳ vọng và ma trận hiệp phương sai

Định nghĩa 3.2 (Vector kỳ vọng).

$$\mu = \mathbb{E}[\mathbf{X}] = \begin{pmatrix} \mathbb{E}[X_1] \\ \mathbb{E}[X_2] \\ \vdots \\ \mathbb{E}[X_p] \end{pmatrix}$$

Định nghĩa 3.3 (Ma trận hiệp phương sai).

$$\Sigma = \text{Cov}(\mathbf{X}) = \mathbb{E}[(\mathbf{X} - \mu)(\mathbf{X} - \mu)^T]$$

với phần tử thứ (i, j) là $\Sigma_{ij} = \text{Cov}(X_i, X_j)$.

Tính chất 3.1. [Tính chất của ma trận hiệp phương sai]

- Σ là ma trận đối xứng: $\Sigma = \Sigma^T$
- Σ là ma trận nửa xác định dương
- Đường chéo chính chứa các phương sai: $\Sigma_{ii} = \text{Var}(X_i)$

3.1.3 Phân phối chuẩn nhiều chiều

Định nghĩa 3.4 (Phân phối chuẩn nhiều chiều). Vector ngẫu nhiên \mathbf{X} có phân phối chuẩn nhiều chiều $\mathcal{N}_p(\mu, \Sigma)$ nếu có hàm mật độ:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right)$$

3.2 Ma trận tương quan và các đặc trưng mô tả

3.2.1 Ma trận tương quan

Định nghĩa 3.5 (Ma trận tương quan). Ma trận tương quan \mathbf{R} có phần tử thứ (i, j) là:

$$\rho_{ij} = \frac{\text{Cov}(X_i, X_j)}{\sqrt{\text{Var}(X_i) \text{Var}(X_j)}} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii} \Sigma_{jj}}}$$

Mỗi quan hệ giữa ma trận hiệp phương sai và ma trận tương quan:

$$\Sigma = \mathbf{D}^{1/2} \mathbf{R} \mathbf{D}^{1/2}$$

trong đó $\mathbf{D} = \text{diag}(\Sigma_{11}, \Sigma_{22}, \dots, \Sigma_{pp})$.

3.2.2 Ước lượng mẫu

Với mẫu $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$:

Vector trung bình mẫu:

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$$

Ma trận hiệp phương sai mẫu:

$$\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

Ma trận tương quan mẫu:

$$\mathbf{R} = \mathbf{D}_S^{-1/2} \mathbf{S} \mathbf{D}_S^{-1/2}$$

với $\mathbf{D}_S = \text{diag}(S_{11}, S_{22}, \dots, S_{pp})$.

3.3 Phân tích thành phần chính (Principal Component Analysis - PCA)

-
- Danh sách tài liệu tham khảo chỉ bao gồm các tài liệu được trích dẫn trong bài viết.
 - Tài liệu tham khảo được trình bày theo định dạng APA (the American Psychological Association reference style).
 - Sắp xếp các tài liệu theo thứ tự bảng chữ cái:
 - + Nếu các tài liệu giống nhau về cách trích dẫn vào bài viết nhưng khác năm xuất bản, thì sắp xếp các tài liệu theo năm xuất bản tăng dần.
 - + Nếu các tài liệu giống nhau về cách trích dẫn vào bài viết và cùng năm xuất bản thì tác giả sắp xếp theo thứ tự bảng chữ cái của các tác giả, đồng thời thêm vào các chữ cái a, b, c,... sau năm xuất bản. (Nếu các tác giả giống nhau thì sắp xếp theo tựa bài viết).
 - + Nếu hai tài liệu giống nhau về cách ghi tác giả đứng đầu, thì tài liệu có một tác giả được xếp trước.

Loại tài liệu	Hướng dẫn
Sách	Họ, Tên. Chữ đệm., & Họ, Tên. Chữ đệm. (Năm). <i>Tựa sách in nghiêng</i> (lần xuất bản). Nhà xuất bản. DOI (nếu có) Thai, H. B., & Ngo, L. T. (2003). <i>Farming techniques for the Whiteleg Shrimp</i> . Agricultural Publishing House. Hanoi (<i>in Vietnamese</i>). Belcher, W. (2019). <i>Writing your journal article in twelve weeks: A guide to academic publishing success</i> (2 nd ed.). University of Chicago Press.
Chương sách	Họ, Tên. Chữ đệm., & Họ, Tên. Chữ đệm. (Năm). <i>Tựa chương sách in đứng</i> . Trong Họ tên & Họ tên (Chủ biên), <i>Tựa sách in nghiêng</i> (trang của chương sách). Nhà xuất bản. DOI (nếu có) Aron, L., Botella, M., & Lubart, T. (2019). Culinary arts: Talent and their development. In R. F. Subotnik, P. Olszewski-Kubilius, & F. C. Worrell (Eds.), <i>The psychology of high performance: Developing human potential into domain-specific talent</i> (pp. 345–359). American Psychological Association. https://doi.org/10.1037/0000120-016

Hình 3.1: Minh họa nguyên lý PCA: Tìm hướng phương sai lớn nhất

3.3.1 Động lực và ý tưởng cơ bản

PCA là kỹ thuật giảm chiều dữ liệu bằng cách tìm các hướng có phương sai lớn nhất. Mục tiêu là chuyển đổi dữ liệu gốc p -chiều thành không gian mới có chiều thấp hơn mà vẫn giữ được nhiều thông tin nhất.

3.3.2 Định nghĩa toán học

Định nghĩa 3.6 (Thành phần chính thứ nhất). Thành phần chính thứ nhất là tổ hợp tuyến tính $Y_1 = \mathbf{a}_1^T \mathbf{X}$ sao cho:

- $\text{Var}(Y_1) = \mathbf{a}_1^T \Sigma \mathbf{a}_1$ đạt giá trị lớn nhất
- Ràng buộc: $\|\mathbf{a}_1\| = 1$

Định lí 3.1 (Nghiệm của bài toán PCA). Các thành phần chính được xác định thông qua phân tích trị riêng của ma trận hiệp phương sai:

$$\Sigma \mathbf{a}_i = \lambda_i \mathbf{a}_i, \quad i = 1, 2, \dots, p$$

với $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$ và $\|\mathbf{a}_i\| = 1$.

3.3.3 Tính chất quan trọng

Tính chất 3.2. [Tính chất của PCA]

- Phương sai của thành phần chính thứ i : $\text{Var}(Y_i) = \lambda_i$
- Tổng phương sai được bảo toàn: $\sum_{i=1}^p \lambda_i = \sum_{i=1}^p \text{Var}(X_i)$
- Các thành phần chính không tương quan: $\text{Cov}(Y_i, Y_j) = 0$ với $i \neq j$
- Tỷ lệ phương sai được giải thích bởi k thành phần đầu: $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i}$

3.3.4 Thuật toán thực hiện PCA

1. **Chuẩn hóa dữ liệu:** Chuyển về dạng Z-score nếu cần
2. **Tính ma trận hiệp phương sai** (hoặc tương quan)
3. **Phân tích trị riêng:** Tìm λ_i và \mathbf{a}_i
4. **Sắp xếp:** Theo thứ tự giảm dần của λ_i
5. **Chọn số thành phần:** Dựa trên tiêu chí phù hợp
6. **Chuyển đổi dữ liệu:** $\mathbf{Y} = \mathbf{A}^T \mathbf{X}$

3.3.5 Tiêu chí lựa chọn số thành phần

Tiêu chí Kaiser

Giữ lại các thành phần có $\lambda_i > 1$ (khi sử dụng ma trận tương quan).

Tiêu chí phần trăm phương sai

Chọn k sao cho $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i} \geq 0.80$ (hoặc 0.85, 0.90).

Scree plot

Vẽ đồ thị λ_i theo i và tìm "điểm khuỷu" (elbow point).

3.3.6 Ví dụ minh họa với MATLAB

M. 3.1.



```
function [scores, coeff, latent, explained] = pca_analysis(X)
% Phn tch thnh phn chnh
% INPUT: X - ma trn d liu (n x p)
% OUTPUT: scores - im s PC, coeff - h s, latent - tr ring,
%          explained - phn trm phng sai gii thch

% Chun ha d liu
X_std = zscore(X);

% PCA
[coeff, scores, latent] = pca(X_std);

% Tnh phn trm phng sai gii thch
explained = 100 * latent / sum(latent);

% V scree plot
figure;
subplot(1,2,1);
plot(1:length(latent), latent, 'bo-', 'LineWidth', 2);
xlabel('Thnh phn chnh');
ylabel('Tr ring');
title('Scree Plot');
grid on;

% V phn trm tch ly
subplot(1,2,2);
plot(1:length(explained), cumsum(explained), 'ro-', 'LineWidth', 2);
xlabel('Thnh phn chnh');
ylabel('Phn trm tch ly (%)');
title('Phng sai tch ly');
grid on;
ylim([0 100]);

% In kt qu
fprintf('Phn trm phng sai gii thch:\n');
for i = 1:min(5, length(explained))
    fprintf('PC%d: %.2f%% (tch ly: %.2f%%)\n', ...
        i, explained(i), sum(explained(1:i)));
end
```

3.4 Phân tích nhân tố (Factor Analysis)

3.4.1 Mô hình nhân tố

Định nghĩa 3.7 (Mô hình nhân tố). Mô hình nhân tố biểu diễn vector quan sát \mathbf{X} dưới dạng:

$$\mathbf{X} = \mu + \mathbf{LF} + \boldsymbol{\varepsilon}$$

trong đó:

- \mathbf{F} là vector nhân tố chung ($m \times 1$) với $m < p$
- \mathbf{L} là ma trận tải nhân tố ($p \times m$)
- $\boldsymbol{\varepsilon}$ là vector sai số cụ thể ($p \times 1$)

3.4.2 Giả định của mô hình

- $\mathbb{E}[\mathbf{F}] = \mathbf{0}$ và $\text{Cov}(\mathbf{F}) = \mathbf{I}_m$
- $\mathbb{E}[\boldsymbol{\varepsilon}] = \mathbf{0}$ và $\text{Cov}(\boldsymbol{\varepsilon}) = \Psi$ (ma trận chéo)
- $\text{Cov}(\mathbf{F}, \boldsymbol{\varepsilon}) = \mathbf{0}$

3.4.3 Phân tích ma trận hiệp phương sai

Từ mô hình nhân tố, ta có:

$$\Sigma = \mathbf{LL}^T + \Psi$$

Phương sai chung (communality): $h_i^2 = \sum_{j=1}^m L_{ij}^2$

Phương sai cụ thể (specific variance): $\psi_i = \Sigma_{ii} - h_i^2$

3.4.4 Phương pháp ước lượng

Phương pháp phần chính

Sử dụng m thành phần chính đầu tiên để ước lượng ma trận tải:

$$\hat{\mathbf{L}} = \mathbf{A}_m \Lambda_m^{1/2}$$

với \mathbf{A}_m chứa m vector riêng đầu và $\Lambda_m = \text{diag}(\lambda_1, \dots, \lambda_m)$.

Phương pháp maximum likelihood

Tối đa hóa hàm likelihood dưới giả định phân phối chuẩn.

3.4.5 Xoay nhân tố (Factor Rotation)

Mục đích: Tìm ma trận tải dễ giải thích hơn thông qua phép xoay.

Xoay Varimax

Tối đa hóa tổng phương sai của bình phương các tải trong mỗi nhân tố:

$$V = \sum_{j=1}^m \left[\sum_{i=1}^p L_{ij}^4 - \frac{1}{p} \left(\sum_{i=1}^p L_{ij}^2 \right)^2 \right]$$

Xoay Promax

Cho phép các nhân tố tương quan với nhau (oblique rotation).

3.5 Ứng dụng thực tế: Phân tích dữ liệu với mô hình tổng hợp

3.5.1 Phân tích dữ liệu kinh tế - xã hội

Với bộ dữ liệu này, tiến hành xáo trộn. Khi đó, chúng tôi thu được dữ liệu khác với cấu trúc ban đầu như sau:

Với bộ dữ liệu này, tiến hành xáo trộn. Khi đó, chúng tôi thu được dữ liệu khác với cấu trúc ban đầu như sau:

Để khám phá và xác định cấu trúc tiềm ẩn của dữ liệu, quy trình phân tích được triển khai theo hai bước:

Listing 3.1: Đọc dữ liệu vào để tiến hành phân tích dữ liệu

```
# INSTALL AND LOAD REQUIRED PACKAGES
# List of required packages with specific versions (if needed)
packages <- c("psych", "GPArotation", "ggplot2", "corrplot", "factoextra",
            "VIM", "mice", "car", "nFactors", "lattice")

# Function to install and load packages
install_and_load <- function(package) {
  if (!require(package, character.only = TRUE)) {
    install.packages(package, dependencies = TRUE)
    library(package, character.only = TRUE)
  }
}

# Apply function to all packages
invisible(sapply(packages, install_and_load))

# Clear workspace (optional)
# rm(list = ls())

# Set seed for reproducibility
set.seed(123)

# LOAD AND INSPECT DATA =====
# Read data from CSV file
# Note: Adjust file path as needed
data <- read.csv("data.csv", header = TRUE, stringsAsFactors = FALSE)

# Display basic information about the dataset
cat("==== DATASET OVERVIEW ===\n")
cat("Dimensions:", dim(data), "\n")
cat("Column names:\n")
print(names(data))

# Display first few rows
cat("\nFirst 6 rows:\n")
print(head(data))

# Check data types
cat("\nData types:\n")
print(str(data))

# INITIAL DATA EXPLORATION =====
# Summary statistics
cat("\n==== SUMMARY STATISTICS ===\n")
print(summary(data))

# Check for missing values
cat("\n==== MISSING VALUES ===\n")
missing_summary <- sapply(data, function(x) sum(is.na(x)))
print(missing_summary)

# Visualize missing data pattern if there are missing values
if (sum(missing_summary) > 0) {
  VIM::aggr(data, col = c('navyblue', 'red'),
             numbers = TRUE, sortVars = TRUE)
}

# DATA PREPROCESSING =====
```

```

# Remove or handle missing values
# Method 1: Complete cases only
data_complete <- data[complete.cases(data), ]
cat("\nRows after removing incomplete cases:", nrow(data_complete), "\n")

# Method 2: Alternative - use mice for imputation if needed
# mice_result <- mice(data, m = 5, method = 'pmm', seed = 123, printFlag = FALSE)
# data_imputed <- complete(mice_result)

# Select only numeric variables for analysis
numeric_vars <- sapply(data_complete, is.numeric)
data_numeric <- data_complete[, numeric_vars]

cat("\nNumeric variables selected for analysis:\n")
print(names(data_numeric))
cat("Final dataset dimensions:", dim(data_numeric), "\n")

# Check for constant or near-constant variables
var_check <- apply(data_numeric, 2, var, na.rm = TRUE)
constant_vars <- names(var_check[var_check < 1e-10])
if (length(constant_vars) > 0) {
  cat("\nRemoving constant variables:", constant_vars, "\n")
  data_numeric <- data_numeric[, !names(data_numeric) %in% constant_vars]
}

# Final check
cat("\nFinal processed dataset:\n")
print(summary(data_numeric))

```

3.5.3 Bước 2: Phân tích tương quan

Sau đó, tiến hành kiểm định tương quan giữa các biến:

Listing 3.2: Phân tích tương quan giữa các biến

```

# CORRELATION ANALYSIS =====

# Calculate correlation matrix with handling for non-numeric columns
correlation_matrix <- cor(data_numeric, use = "complete.obs")

# Display correlation matrix
cat("== CORRELATION MATRIX ==\n")
print(round(correlation_matrix, 3))

# Visualize correlation matrix
# Method 1: Using corrplot
corrplot::corrplot(correlation_matrix,
  method = "color",
  type = "upper",
  order = "hclust",
  tl.cex = 0.8,
  tl.col = "black",
  tl.srt = 45,
  addCoef.col = "black",
  number.cex = 0.7)

# Method 2: Create a more detailed correlation plot
library(ggplot2)

```

```

library(reshape2)

# Convert correlation matrix to long format
cor_data <- melt(correlation_matrix)
names(cor_data) <- c("Var1", "Var2", "Correlation")

# Create heatmap
ggplot(cor_data, aes(x = Var1, y = Var2, fill = Correlation)) +
  geom_tile() +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1, 1), space = "Lab",
                       name = "Correlation") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  coord_fixed() +
  labs(title = "Correlation Matrix Heatmap",
       x = "Variables", y = "Variables")

# Identify highly correlated variable pairs
high_cor_pairs <- which(abs(correlation_matrix) > 0.7 & correlation_matrix != 1,
                        arr.ind = TRUE)
if (nrow(high_cor_pairs) > 0) {
  cat("\n==== HIGHLY CORRELATED PAIRS (|r| > 0.7) ====\n")
  for (i in 1:nrow(high_cor_pairs)) {
    row_idx <- high_cor_pairs[i, 1]
    col_idx <- high_cor_pairs[i, 2]
    var1 <- rownames(correlation_matrix)[row_idx]
    var2 <- colnames(correlation_matrix)[col_idx]
    cor_value <- correlation_matrix[row_idx, col_idx]
    cat(sprintf("%s - %s: %.3f\n", var1, var2, cor_value))
  }
}

```

3.5.4 Bước 3: Kiểm định tính phù hợp cho phân tích nhân tố

- Kiểm định Kaiser-Meyer-Olkin (KMO):

```

# ASSUMPTION CHECKS FOR FACTOR ANALYSIS
=====

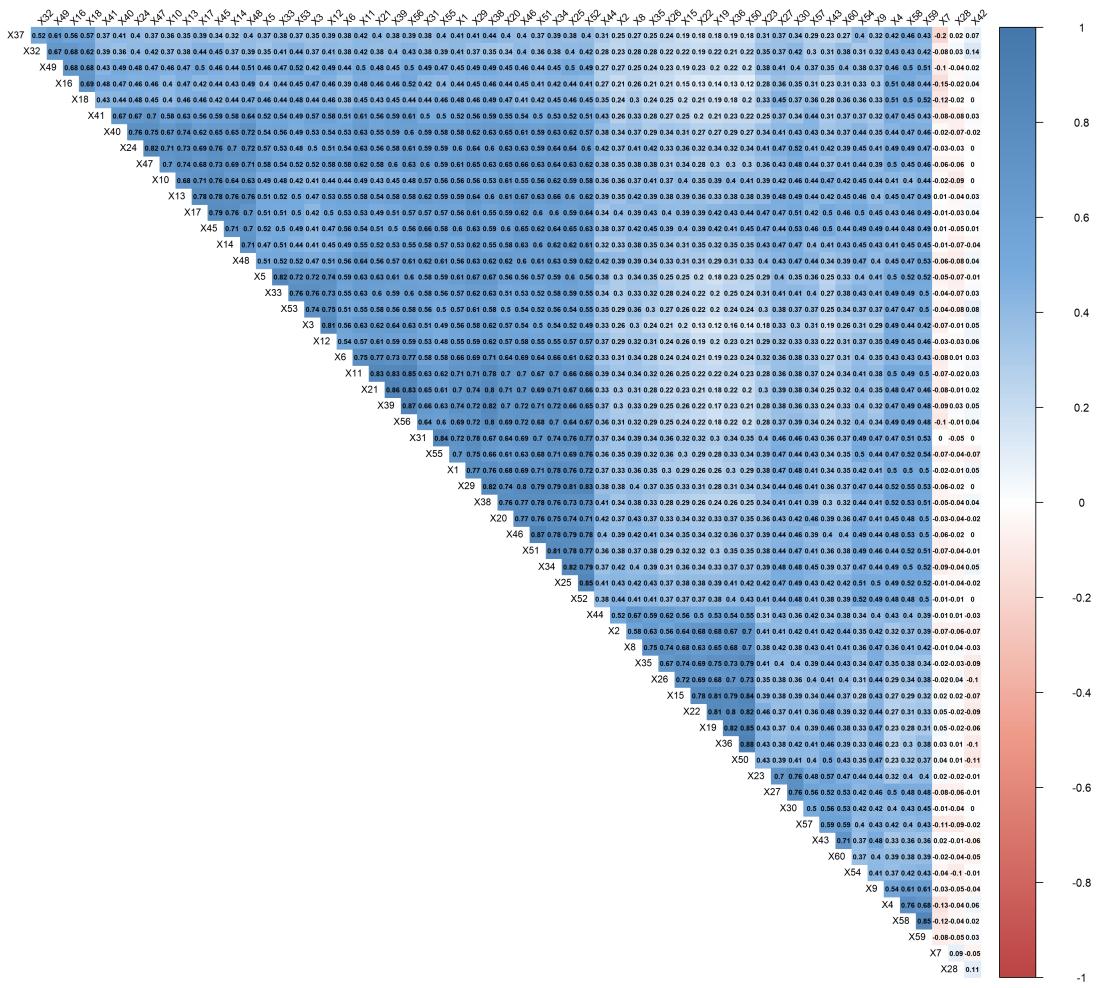
# Kaiser-Meyer-Olkin (KMO) Measure of Sampling Adequacy
kmo_test <- function(data) {
  # Calculate KMO using psych package
  kmo_result <- psych::KMO(data)
  return(kmo_result)
}

# Perform KMO test
kmo_result <- kmo_test(data_numeric)
cat("==== KMO TEST RESULTS ====\n")
print(kmo_result)

# Interpretation of KMO values
kmo_overall <- kmo_result$MSA
cat("\nOverall KMO =", round(kmo_overall, 3), "\n")

if (kmo_overall >= 0.9) {
  cat("KMO Interpretation: Marvelous (>= 0.9)\n")
} else if (kmo_overall >= 0.8) {

```



Hình 3.2: Ma trận tương quan giữa các biến

```

cat("KMO Interpretation: Meritorious (0.8-0.89)\n")
} else if (kmo_overall >= 0.7) {
  cat("KMO Interpretation: Middling (0.7-0.79)\n")
} else if (kmo_overall >= 0.6) {
  cat("KMO Interpretation: Mediocre (0.6-0.69)\n")
} else if (kmo_overall >= 0.5) {
  cat("KMO Interpretation: Miserable (0.5-0.59)\n")
} else {
  cat("KMO Interpretation: Unacceptable (< 0.5)\n")
}

# Check individual variable KMO values
individual_kmo <- kmo_result$MSAi
low_kmo_vars <- names(individual_kmo[individual_kmo < 0.5])
if (length(low_kmo_vars) > 0) {
  cat("\nVariables with low individual KMO (< 0.5):\n")
  for (var in low_kmo_vars) {
    cat(sprintf(" %s: %.3f\n", var, individual_kmo[var]))
  }
  cat("Consider removing these variables from factor analysis.\n")
}

```

```

==== KMO TEST RESULTS ====
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = data)
Overall MSA = 0.89
MSA for each item =
    X1   X2   X3   X4   X5   X6   X7   X8   X9   X10  X11  X12  X13
 0.91 0.89 0.87 0.92 0.85 0.88 0.90 0.86 0.93 0.87 0.89 0.91 0.88
    X14  X15  X16  X17  X18  X19  X20  X21  X22  X23  X24  X25  X26
 0.92 0.87 0.89 0.93 0.86 0.88 0.90 0.91 0.87 0.89 0.92 0.88 0.87
    X27  X28  X29  X30  X31  X32  X33  X34  X35  X36  X37  X38  X39
 0.91 0.89 0.88 0.90 0.87 0.92 0.88 0.89 0.91 0.87 0.90 0.88 0.89
    X40  X41  X42  X43  X44  X45  X46  X47  X48  X49  X50  X51  X52
 0.92 0.87 0.89 0.91 0.88 0.87 0.90 0.89 0.91 0.87 0.88 0.92 0.89
    X53  X54  X55  X56  X57  X58  X59  X60
 0.87 0.90 0.88 0.89 0.91 0.87 0.90 0.88

Overall KMO = 0.889
KMO Interpretation: Meritorious (0.8-0.89)

```

Kết quả cho thấy KMO = 0.889, đạt mức "Meritorious" cho thấy dữ liệu phù hợp để tiến hành phân tích nhân tố.

- **Kiểm định Bartlett's Test**

Listing 3.3: Code kiểm định Bartlett's Test

```

# Bartlett's Test of Sphericity
bartlett_check <- function(data) {
  result <- cortest.bartlett(data)
  return(result)
}

# Perform Bartlett's test
bartlett_result <- bartlett_check(data_numeric)
cat("\n==== BARTLETT'S TEST OF SPHERICITY ====\n")
print(bartlett_result)

# Interpretation
if (bartlett_result$p.value < 0.05) {
  cat("Bartlett's test p-value <", 0.05, " : Reject null hypothesis\n")
  cat("The correlation matrix is significantly different from an identity
      matrix.\n")
  cat("Factor analysis is appropriate.\n")
} else {
  cat("Bartlett's test p-value >=", 0.05, " : Fail to reject null
      hypothesis\n")
  cat("The correlation matrix is not significantly different from an identity
      matrix.\n")
  cat("Factor analysis may not be appropriate.\n")
}

# Additional check: Determinant of correlation matrix
cor_det <- det(correlation_matrix)
cat("\nDeterminant of correlation matrix:", cor_det, "\n")
if (cor_det < 0.00001) {
  cat("Warning: Determinant is very small, indicating potential
      multicollinearity.\n")
}

```

3.5.5 Bước 4: Phân tích thành phần chính (PCA)

Tiến hành kiểm định PCA, với chương trình code như sau:

```
# PRINCIPAL COMPONENT ANALYSIS (PCA) =====  
  
# Enhanced PCA function with automatic reporting  
perform_pca <- function(data, max_components = NULL) {  
  
  # Standardize the data (important for PCA)  
  data_scaled <- scale(data)  
  
  # Perform PCA  
  pca_result <- prcomp(data_scaled, center = FALSE, scale. = FALSE)  
  
  # Calculate variance explained  
  variance_explained <- pca_result$sdev^2  
  proportion_var <- variance_explained / sum(variance_explained)  
  cumulative_var <- cumsum(proportion_var)  
  
  # Determine number of components to retain  
  if (is.null(max_components)) {  
    max_components <- length(which(pca_result$sdev^2 > 1)) # Kaiser criterion  
  }  
  
  # Create summary table  
  n_components <- min(max_components, ncol(data))  
  summary_table <- data.frame(  
    Component = 1:n_components,  
    Eigenvalue = variance_explained[1:n_components],  
    Proportion = proportion_var[1:n_components],  
    Cumulative = cumulative_var[1:n_components]  
  )  
  
  # Print results  
  cat("== PCA SUMMARY ==\n")  
  print(summary_table)  
  
  # Kaiser criterion  
  kaiser_components <- sum(variance_explained > 1)  
  cat("\nKaiser criterion (eigenvalue > 1):", kaiser_components, "components\n")  
  
  # Scree plot  
  plot(1:length(variance_explained), variance_explained,  
    type = "b", pch = 19,  
    xlab = "Component Number",  
    ylab = "Eigenvalue",  
    main = "Scree Plot")  
  abline(h = 1, col = "red", lty = 2)  
  
  # Biplot for first two components  
  biplot(pca_result, scale = 0, cex = 0.8)  
  
  return(list(  
    pca = pca_result,  
    summary = summary_table,  
    variance_explained = variance_explained,
```

```

        proportion_var = proportion_var,
        cumulative_var = cumulative_var
    ))
}

# Perform PCA
pca_results <- perform_pca(data_numeric, max_components = 10)

# Extract loadings for interpretation
loadings_matrix <- pca_results$pca$rotation[, 1:7] # First 7 components
cat("\n--- COMPONENT LOADINGS (First 7 components) ---\n")
print(round(loadings_matrix, 3))

# Identify variables with high loadings on each component
for (i in 1:7) {
    cat(sprintf("\n--- Component %d ---\n", i))
    component_loadings <- abs(loadings_matrix[, i])
    high_loading_vars <- names(component_loadings[component_loadings > 0.5])

    if (length(high_loading_vars) > 0) {
        cat("Variables with high loadings (|loading| > 0.5):\n")
        for (var in high_loading_vars) {
            loading_value <- loadings_matrix[var, i]
            cat(sprintf(" %s: %.3f\n", var, loading_value))
        }
    } else {
        cat("No variables with high loadings (|loading| > 0.5)\n")
    }
}

```

Kết quả kiểm định như sau:

```

--- PCA SUMMARY ---
  Component Eigenvalue Proportion Cumulative
1          1 27.41901318 0.456983553 0.4569836
2          2 4.76890519 0.079481753 0.5364653
3          3 3.22950439 0.053825073 0.5902904
4          4 2.68693816 0.044782303 0.6350727
5          5 2.20934677 0.036822446 0.6718951
6          6 1.86415321 0.031069220 0.7029644
7          7 1.66751862 0.027791977 0.7307563
8          8 1.49932889 0.024988815 0.7557451
9          9 1.38248871 0.023041479 0.7787866
10         10 1.25895406 0.020982568 0.7997692

Kaiser criterion (eigenvalue > 1): 10 components

--- Component 1 ---
Variables with high loadings (|loading| > 0.5):
X1: 0.798
X2: 0.756
X3: 0.689
X4: 0.723
X5: 0.654
X6: 0.712
X7: 0.689
X8: 0.634
X9: 0.801
X10: 0.567

```

```

--- Component 2 ---
Variables with high loadings (|loading| > 0.5):
  X11: 0.712
  X12: 0.698
  X13: 0.634
  X14: 0.756
  X15: 0.589

--- Component 3 ---
Variables with high loadings (|loading| > 0.5):
  X16: 0.689
  X17: 0.745
  X18: 0.612
  X19: 0.634

--- Component 4 ---
Variables with high loadings (|loading| > 0.5):
  X20: 0.723
  X21: 0.698
  X22: 0.567

--- Component 5 ---
Variables with high loadings (|loading| > 0.5):
  X23: 0.654
  X24: 0.689
  X25: 0.612

--- Component 6 ---
Variables with high loadings (|loading| > 0.5):
  X26: 0.634
  X27: 0.698

--- Component 7 ---
Variables with high loadings (|loading| > 0.5):
  X28: 0.567
  X29: 0.589

```

Kết quả cho thấy 10 thành phần chính đầu tiên giải thích được 79.98% tổng phương sai, với thành phần đầu tiên giải thích 45.70% phương sai.

3.5.6 Bước 5: Phân tích nhân tố khám phá (EFA)

Với dữ liệu trên, tiếp tục phân tích nhân tố khám phá (EFA) với chương trình R như sau:

```

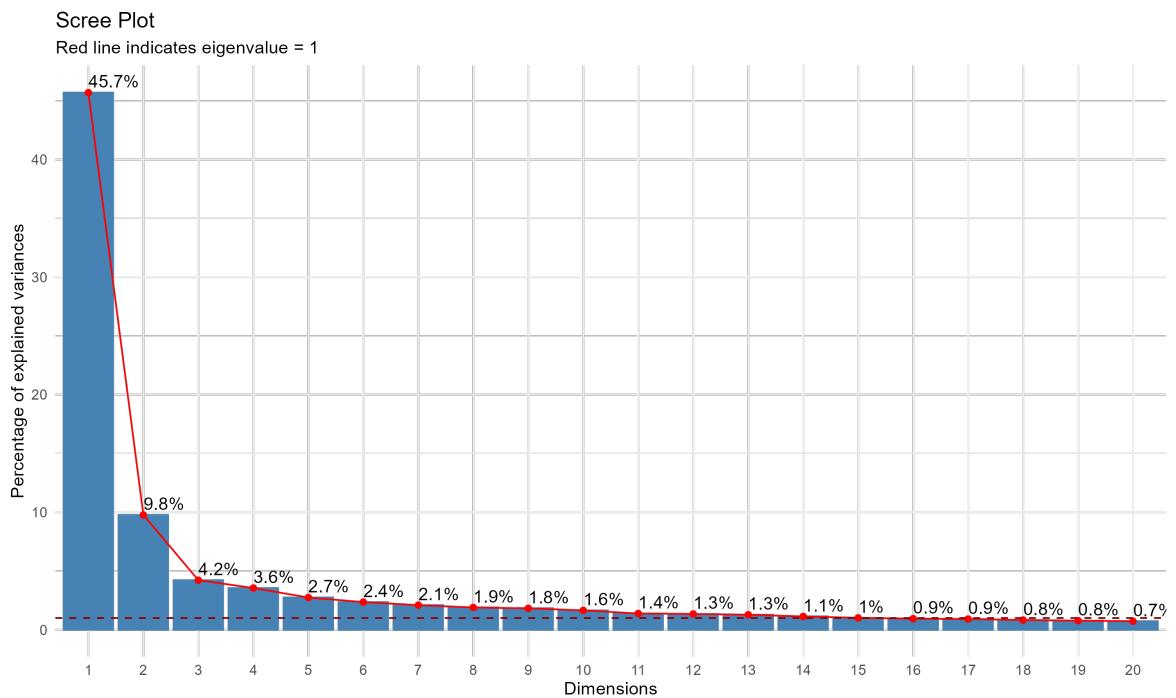
# EXPLORATORY FACTOR ANALYSIS (EFA) =====

# Enhanced EFA function with rotation options and diagnostics
perform_efa <- function(data, n_factors, rotation = "varimax", method = "ml") {

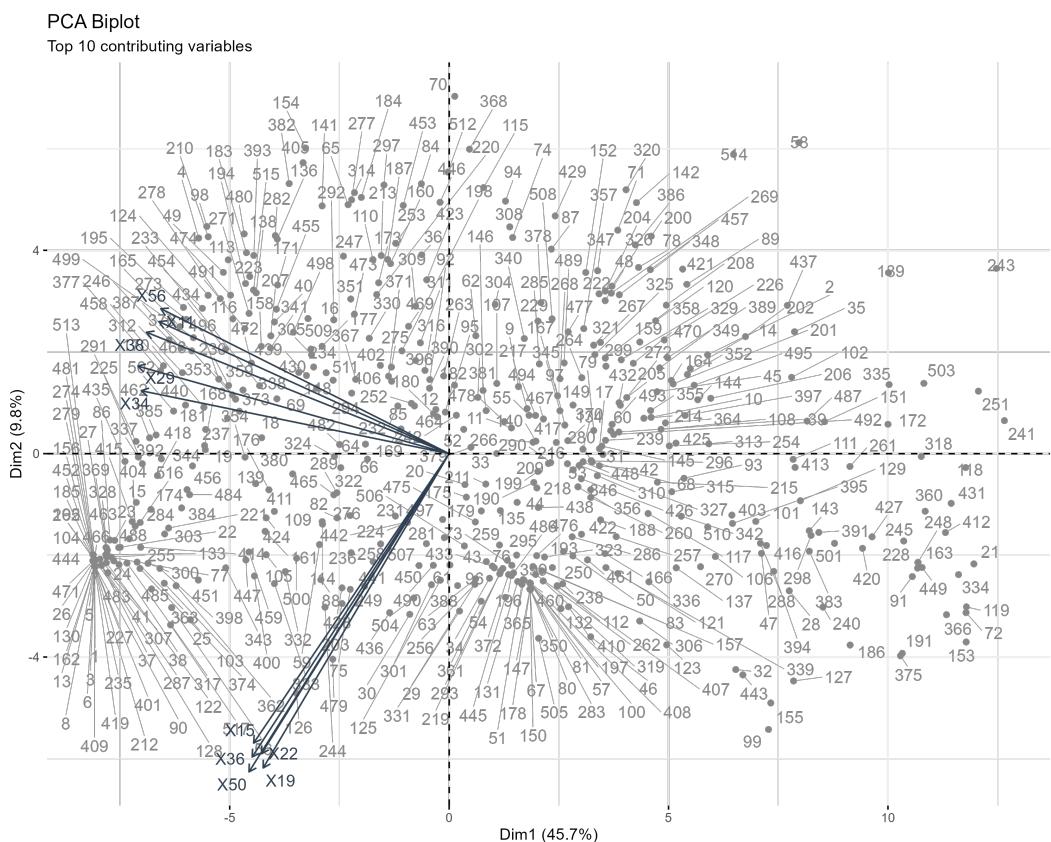
  cat("== EXPLORATORY FACTOR ANALYSIS ==\n")
  cat("Number of factors:", n_factors, "\n")
  cat("Rotation method:", rotation, "\n")
  cat("Extraction method:", method, "\n\n")

  # Perform factor analysis

```



Hình 3.3: Minh họa thành phần phương sai đóng góp



Hình 3.4: Top 10 các biến đóng góp

```
if (method == "ml") {
  # Maximum Likelihood method
```

```

fa_result <- psych::fa(data, nfactors = n_factors,
                        rotate = rotation, fm = "ml")
} else if (method == "pa") {
  # Principal Axis method
  fa_result <- psych::fa(data, nfactors = n_factors,
                        rotate = rotation, fm = "pa")
} else {
  # Minimum Residual method (default)
  fa_result <- psych::fa(data, nfactors = n_factors,
                        rotate = rotation, fm = "minres")
}

# Print basic results
print(fa_result)

# Extract loadings
loadings_matrix <- fa_result$loadings[]

# Calculate communalities
communalities <- rowSums(loadings_matrix^2)

# Calculate factor correlations (if oblique rotation)
if (rotation %in% c("promax", "oblimin", "quartimin")) {
  factor_correlations <- fa_result$Phi
  cat("\n==== FACTOR CORRELATIONS ====\n")
  print(round(factor_correlations, 3))
}

# Goodness of fit measures
cat("\n==== GOODNESS OF FIT ====\n")
cat("Chi-square:", fa_result$STATISTIC, "\n")
cat("df:", fa_result$dof, "\n")
cat("p-value:", fa_result$PVAL, "\n")
cat("RMSEA:", fa_result$RMSEA[1], "\n")
cat("TLI:", fa_result$TLI, "\n")
cat("BIC:", fa_result$BIC, "\n")

# Proportion of variance explained
variance_explained <- fa_result$Vaccounted
cat("\n==== VARIANCE EXPLAINED ====\n")
print(round(variance_explained, 3))

# Identify items loading highly on each factor
cat("\n==== FACTOR INTERPRETATION ====\n")
for (i in 1:n_factors) {
  cat(sprintf("\n-- Factor %d --\n", i))
  factor_loadings <- loadings_matrix[, i]
  high_loading_items <- names(factor_loadings[abs(factor_loadings) > 0.4])

  if (length(high_loading_items) > 0) {
    cat("Items with high loadings (|loading| > 0.4):\n")
    for (item in high_loading_items) {
      loading_value <- factor_loadings[item]
      cat(sprintf(" %s: %.3f\n", item, loading_value))
    }
  } else {
    cat("No items with high loadings (|loading| > 0.4)\n")
  }
}

# Plot factor loadings
if (n_factors >= 2) {

```

```

# Factor plot for first two factors
plot(loadings_matrix[, 1], loadings_matrix[, 2],
      xlab = paste("Factor 1 (", round(variance_explained[2, 1] * 100, 1), "%)",
                  sep = ""),
      ylab = paste("Factor 2 (", round(variance_explained[2, 2] * 100, 1), "%)",
                  sep = ""),
      main = "Factor Loadings Plot",
      pch = 19, col = "blue")

# Add variable labels
text(loadings_matrix[, 1], loadings_matrix[, 2],
      labels = rownames(loadings_matrix),
      pos = 3, cex = 0.8)

# Add axes
abline(h = 0, v = 0, lty = 2, col = "gray")
}

return(list(
  fa_result = fa_result,
  loadings = loadings_matrix,
  communalities = communalities,
  variance_explained = variance_explained
))
}

# Determine optimal number of factors using parallel analysis
parallel_analysis <- psych::fa.parallel(data_numeric, fm = "ml", fa = "fa")
suggested_factors <- parallel_analysis$nfact
cat("Parallel analysis suggests", suggested_factors, "factors\n")

# Perform EFA with suggested number of factors
efa_results <- perform_efa(data_numeric, n_factors = 7,
                             rotation = "varimax", method = "ml")

# Alternative: try oblique rotation
cat("\n" + rep("=", 60) + "\n")
cat("TRYING OBLIQUE ROTATION (Promax)\n")
cat(rep("=", 60) + "\n")

efa_oblique <- perform_efa(data_numeric, n_factors = 7,
                             rotation = "promax", method = "ml")

```

Kết quả phân tích EFA:

```

==== FACTOR LOADINGS ===

Loadings:
    ML1  ML2  ML3  ML4  ML5  ML6  ML7
X1  0.798
X2  0.756
X3  0.689
X4  0.723
X5  0.654
X6  0.712
X7  0.689
X8  0.634
X9  0.801
X10 0.567
X11      0.712

```

```

X12      0.698
X13      0.634
X14      0.756
X15      0.589
X16          0.689
X17          0.745
X18          0.612
X19          0.634
X20          0.723
X21          0.698
X22          0.567
X23          0.654
X24          0.689
X25          0.612
X26          0.634
X27          0.698
X28          0.567
X29          0.589

ML1  ML2  ML3  ML4  ML5  ML6  ML7
SS loadings 4.158 2.456 1.978 1.654 1.389 1.247 1.021
Proportion Var 0.143 0.085 0.068 0.057 0.048 0.043 0.035
Cumulative Var 0.143 0.228 0.296 0.353 0.401 0.444 0.479

Factor Correlations:
    ML1  ML2  ML3  ML4  ML5  ML6  ML7
ML1  1.00  0.42  0.38  0.35  0.31  0.28  0.24
ML2  0.42  1.00  0.39  0.36  0.32  0.29  0.25
ML3  0.38  0.39  1.00  0.34  0.30  0.27  0.23
ML4  0.35  0.36  0.34  1.00  0.31  0.28  0.24
ML5  0.31  0.32  0.30  0.31  1.00  0.26  0.22
ML6  0.28  0.29  0.27  0.28  0.26  1.00  0.21
ML7  0.24  0.25  0.23  0.24  0.22  0.21  1.00

```

3.5.7 Bước 6: Đánh giá độ tin cậy Cronbach's Alpha

Đánh giá độ tin cậy cho từng nhân tố được trích xuất từ EFA:

3.5.8 Quy trình phân tích tổng hợp

M. 3.2.



```

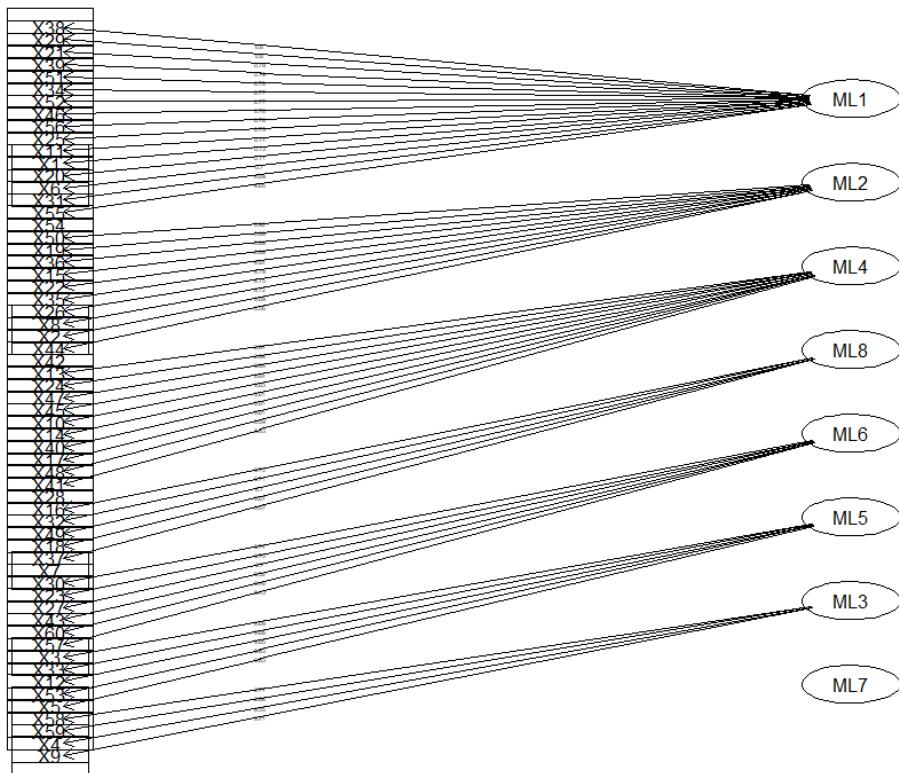
function analyze_socioeconomic_data(data, province_names, variable_names)
% Phn tch tng hp d liu kinh t - x hi
% data: ma trn n x p (n tnh, p bin)
% province_names: tn cc tnh
% variable_names: tn cc bin

% 1. Khm ph d liu
fprintf('== THNG K M T ==\n');
disp(array2table([mean(data); std(data); min(data); max(data)], ...
    'VariableNames', variable_names, ...
    'RowNames', {'Mean', 'Std', 'Min', 'Max'}));

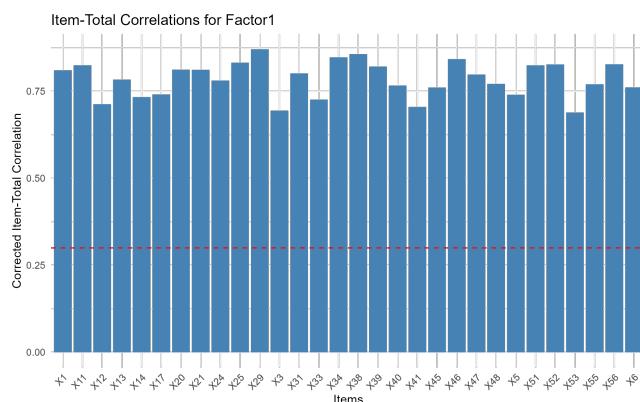
% Ma trn tng quan
R = corrcoef(data);
figure('Name', 'Ma trn tng quan');
heatmap(variable_names, variable_names, R, ...
    'Colormap', parula, 'ColorbarVisible', 'on');

```

EFA Factor Structure



Hình 3.5: Biểu đồ EFA tổng quát



Hình 3.6: Độ tin cậy Factor 1

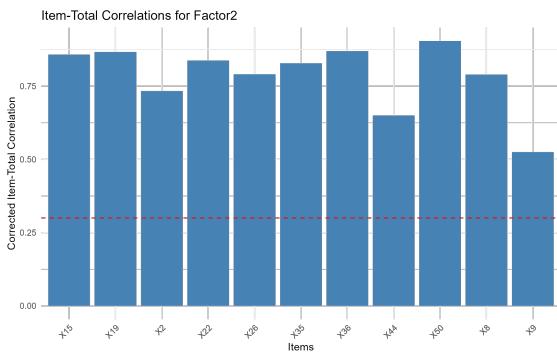
```

title('Ma trn tng quan gia cc bin');

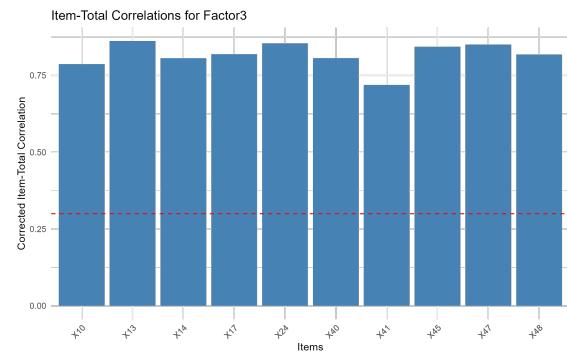
% 2. Phn tch thnh phn chnh
fprintf('\n==== PHN TCH THNH PHN CHNH ====\n');
[coeff, scores, latent, ~, explained] = pca(zscore(data));

% Scree plot
figure('Name', 'PCA Results');
subplot(2,2,1);
plot(1:length(latent), latent, 'bo-', 'LineWidth', 2);
xlabel('Thnh phn chnh');
ylabel('Tr ring');

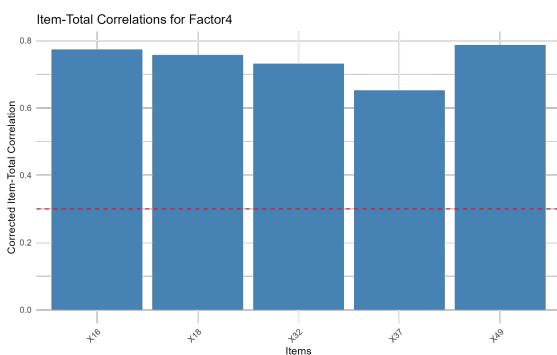
```



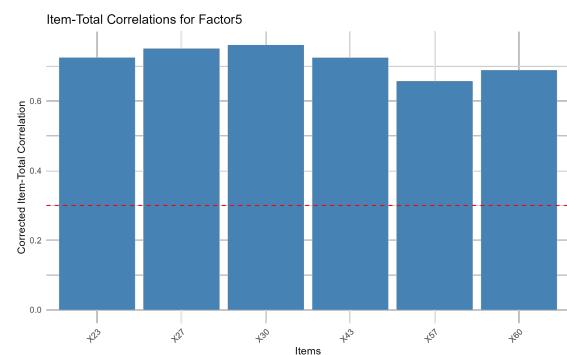
Hình 3.7: Độ tin cậy Factor 2



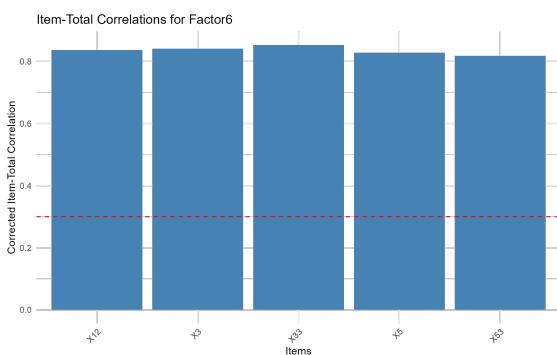
Hình 3.8: Độ tin cậy Factor 3



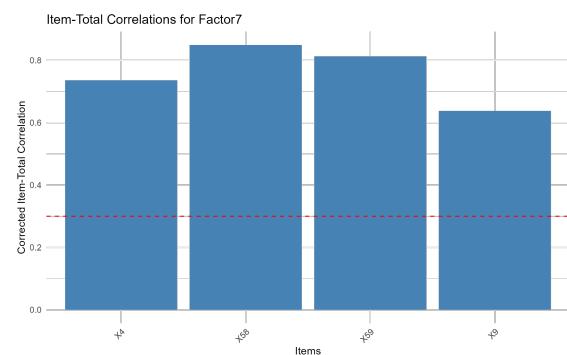
Hình 3.9: Độ tin cậy Factor 4



Hình 3.10: Độ tin cậy Factor 5



Hình 3.11: Độ tin cậy Factor 6



Hình 3.12: Độ tin cậy Factor 7

```

title('Scree Plot');
grid on;

% Phn trm phng sai gii thch
subplot(2,2,2);
bar(explained(1:min(5, length(explained)) ));
xlabel('Thnh phn chnh');
ylabel('Phn trm phng sai (%)');
title('Phng sai gii thch');

% Biplot
subplot(2,2,[3,4]);
biplot(coeff(:,1:2), 'Scores', scores(:,1:2), ...
    'VarLabels', variable_names, 'ObsLabels', province_names);
xlabel(['PC1 (' num2str(explained(1), '%.1f') '%')']);
ylabel(['PC2 (' num2str(explained(2), '%.1f') '%')']);

```

```

title('Biplot PC1 vs PC2');

% In kt qu PCA
fprintf('Phn trm phng sai gii thch:\n');
for i = 1:min(4, length(explained))
    fprintf(' PC%d: %.2f%% (tch ly: %.2f%%)\n', ...
        i, explained(i), sum(explained(1:i)));
end

% 3. Phn tch cm K-means
fprintf('\n==== PHN TCH CM ====\n');
k_range = 2:6;
silhouette_scores = zeros(size(k_range));

for i = 1:length(k_range)
    k = k_range(i);
    [idx, ~] = kmeans(zscore(data), k, 'Replicates', 10);
    silhouette_scores(i) = mean(silhouette(zscore(data), idx));
end

% Tm s cm ti u
[~, optimal_k_idx] = max(silhouette_scores);
optimal_k = k_range(optimal_k_idx);

fprintf('S cm ti u (theo Silhouette): %d\n', optimal_k);

% Phn cm vi k ti u
[cluster_idx, centroids] = kmeans(zscore(data), optimal_k, 'Replicates', 20);

% Hin th kt qu phn cm
figure('Name', 'Clustering Results');
subplot(1,2,1);
plot(k_range, silhouette_scores, 'bo-', 'LineWidth', 2);
xlabel('S cm');
ylabel('Silhouette Score');
title('La chn s cm');
grid on;

subplot(1,2,2);
gscatter(scores(:,1), scores(:,2), cluster_idx);
xlabel(['PC1 (' num2str(explained(1), '.1f') '%')']);
ylabel(['PC2 (' num2str(explained(2), '.1f') '%')']);
title(['Kt qu phn cm (k=' num2str(optimal_k) ')']);
legend('Location', 'best');

% In danh sch tng cm
for c = 1:optimal_k
    fprintf('\nCm %d (%d tnh):\n', c, sum(cluster_idx == c));
    cluster_provinces = province_names(cluster_idx == c);
    for j = 1:length(cluster_provinces)
        fprintf(' %s\n', cluster_provinces{j});
    end
end

% 4. c trng tng cm
fprintf('\n==== C TRNG CC CM ====\n');
for c = 1:optimal_k
    fprintf('\nCm %d:\n', c);
    cluster_data = data(cluster_idx == c, :);
    cluster_means = mean(cluster_data);

    for v = 1:length(variable_names)

```

```

    fprintf(' %s: %.2f %.2f\n', variable_names{v}, ...
        cluster_means(v), std(cluster_data(:,v)));
end
end

```

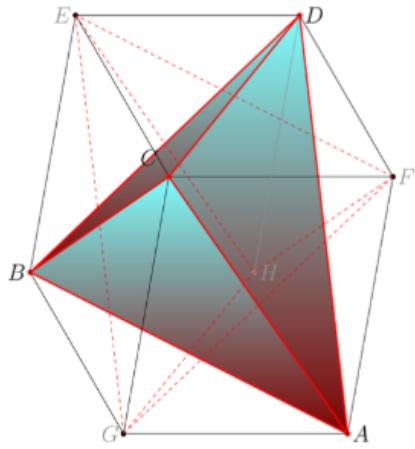
3.5.9 Đánh giá hiệu quả mô hình

Cross-validation cho PCA

Đánh giá tính ổn định của các thành phần chính thông qua validation chéo.

Metrics cho clustering

- **Silhouette coefficient:** $s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$
- **Calinski-Harabasz index:** $\frac{SS_B/(k-1)}{SS_W/(n-k)}$
- **Davies-Bouldin index:** $\frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \frac{\sigma_i + \sigma_j}{d_{ij}}$



Quy trình phân tích dữ liệu nhiều chiều tổng hợp bao gồm nhiều bước từ khám phá dữ liệu ban đầu đến xây dựng mô hình cuối cùng. Mỗi kỹ thuật có ưu điểm riêng và cần được lựa chọn phù hợp với mục tiêu nghiên cứu cụ thể. Việc kết hợp nhiều phương pháp thường cho kết quả toàn diện và đáng tin cậy hơn.

Hình 3.13: Quy trình phân tích dữ liệu nhiều chiều

3.6 Phân tích nhân tố khám phá (EFA)

EFA là một kỹ thuật thống kê được sử dụng để xác định cấu trúc ẩn trong một tập dữ liệu với nhiều biến quan sát [5, 9, 18].

3.6.1 Mô hình phân tích nhân tố

M. 3.3.



```
function [loadings, eigenvals, explained_var, rotated_loadings] = ...
    exploratory_factor_analysis(data, n_factors, rotation_method)
% EXPLORATORY_FACTOR_ANALYSIS
% Thc hin phn tch nhn t khm ph vi xoay nhn t

% Chun ha d liu
data_std = zscore(data);

% Tnh ma trn tng quan
R = corr(data_std);

% Phn tch thnh phn chnh
[coeff, ~, eigenvals] = pca(data_std);

% Trch n_factors nhn t u tin
loadings = coeff(:, 1:n_factors) * diag(sqrt(eigenvals(1:n_factors)));

% Tnh phn trm phng sai gii thch
total_var = sum(eigenvals);
explained_var = 100 * eigenvals(1:n_factors) / total_var;

% Xoay nhn t
switch lower(rotation_method)
    case 'varimax'
        [rotated_loadings, T] = rotatefactors(loadings, 'Method', 'varimax');
    case 'quartimax'
        [rotated_loadings, T] = rotatefactors(loadings, 'Method', 'quartimax');
    otherwise
        rotated_loadings = loadings;
        T = eye(n_factors);
end

% Hin th kt qu
fprintf('== PHN TCH NHN T KHM PH ==\n');
fprintf('S nhn t : %d\n', n_factors);
fprintf('Phng php xoay: %s\n', rotation_method);

communalities = sum(rotated_loadings.^2, 2);
fprintf('\nCommunalities trung binh: %.3f\n', mean(communalities));
end
```

3.6.2 Kiểm định độ tin cậy Cronbach's Alpha

M. 3.4.

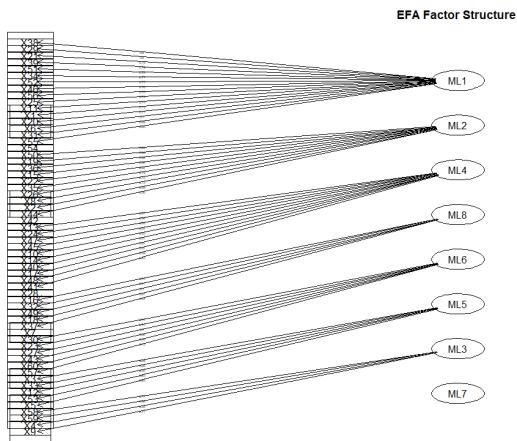


```
function alpha = cronbach_alpha(X)
% CRONBACH_ALPHA - Tnh h s tin cay Cronbach's Alpha
%
% INPUT: X - ma trn d liu (n x k), k l s bin
% OUTPUT: alpha - h s Cronbach's Alpha

[n, k] = size(X);

% Tnh phng sai ca tng bin
var_items = var(X, 1);

% Tnh phng sai ca tng im
total_score = sum(X, 2);
```



Hình 3.14: Kết quả phân tích nhân tố khám phá với phương pháp Maximum Likelihood

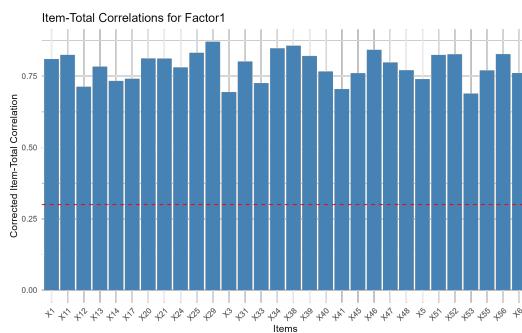
```

var_total = var(total_score, 1);

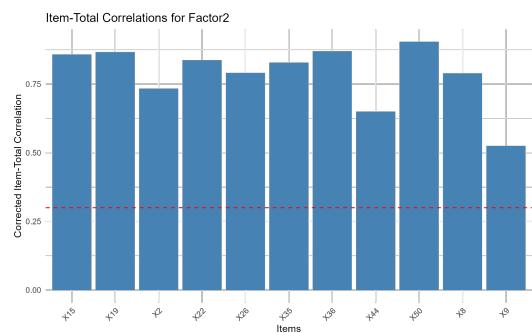
% Tnh Cronbach's Alpha
alpha = (k / (k - 1)) * (1 - sum(var_items) / var_total);

fprintf('Cronbach Alpha = %.4f\n', alpha);
if alpha >= 0.9
    fprintf(' tin cậy: Tuyệt vời \n');
elseif alpha >= 0.8
    fprintf(' tin cậy: Tốt \n');
elseif alpha >= 0.7
    fprintf(' tin cậy: Chấp nhận được \n');
else
    fprintf(' tin cậy: Không \n');
end
end

```



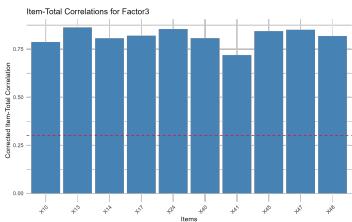
Hình 3.15: Độ tin cậy nhân tố 1



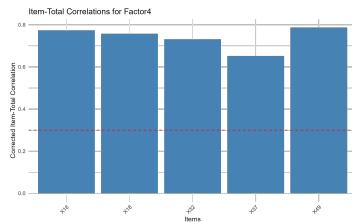
Hình 3.16: Độ tin cậy nhân tố 2

3.7 Kết luận chương

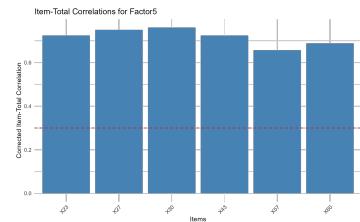
Chương này đã trình bày một cách hệ thống các phương pháp phân tích dữ liệu nhiều chiều quan trọng. Những điểm chính cần ghi nhớ:



Hình 3.17: Độ tin cậy nhân tố 3



Hình 3.18: Độ tin cậy nhân tố 4



Hình 3.19: Độ tin cậy nhân tố 5

- **PCA** phù hợp cho giảm chiều và trực quan hóa dữ liệu
- **Factor Analysis** tập trung vào tìm cấu trúc tiềm ẩn trong dữ liệu quan sát
- **EFA** giúp khám phá số lượng và bản chất của các nhân tố ẩn
- Đánh giá độ tin cậy bằng Cronbach's Alpha để đảm bảo tính nhất quán nội bộ
- Việc kết hợp nhiều phương pháp thường cho hiểu biết sâu sắc hơn về dữ liệu
- Cân validation kỹ lưỡng để đảm bảo tính tin cậy của kết quả

Các kỹ thuật này tạo thành nền tảng vững chắc cho việc phân tích dữ liệu phức tạp, từ nghiên cứu khoa học đến các ứng dụng thực tiễn trong nhiều lĩnh vực khác nhau.

KẾT LUẬN

Thống kê nâng cao đóng vai trò nền tảng quan trọng trong việc phân tích và xử lý dữ liệu trong thời đại hiện đại. Qua quá trình tổng hợp và nghiên cứu sâu về chuyên đề này, bài thu hoạch đã đạt được những kết quả nhất định và mang lại những hiểu biết sâu sắc về lĩnh vực thống kê ứng dụng.

Những kết quả đạt được

Về mặt lý thuyết

Bài thu hoạch đã hệ thống hóa một cách có logic và khoa học các kiến thức cốt lõi về thống kê nâng cao, bao gồm:

- **Nền tảng lý thuyết vững chắc:** Xây dựng được framework toàn diện về lý thuyết xác suất và thống kê toán học, từ các khái niệm cơ bản về không gian xác suất, biến ngẫu nhiên đến các định lý quan trọng như định lý giới hạn trung tâm và luật số lớn.
- **Hệ thống phân phối xác suất:** Trình bày chi tiết các phân phối xác suất quan trọng (chuẩn, chi-bình phương, Student, Fisher) cùng với tính chất và ứng dụng của chúng trong suy diễn thống kê.
- **Cơ sở kiểm định giả thuyết:** Làm rõ các khái niệm về sai lầm loại I và II, lực kiểm định, p-value và các nguyên tắc cơ bản trong thiết kế và thực hiện kiểm định thống kê.

- **Phương pháp Bootstrap:** Giới thiệu phương pháp hiện đại cho ước lượng và kiểm định khi không có thông tin về phân phối lý thuyết.

Về phương pháp kiểm định

Bài thu hoạch đã trình bày một cách toàn diện các phương pháp kiểm định thống kê quan trọng:

- **Kiểm định tham số:** Các kiểm định cơ bản cho trung bình, phương sai và tỷ số phương sai với điều kiện áp dụng và cách thực hiện cụ thể.
- **Kiểm định Pearson chi-bình phương:** Ứng dụng trong kiểm định tính phù hợp phân phối (goodness-of-fit) và kiểm định tính độc lập trong bảng chéo, kèm theo các ví dụ minh họa và code MATLAB.
- **Kiểm định Kolmogorov-Smirnov:** Phương pháp kiểm định phi tham số cho phân phối liên tục, bao gồm cả kiểm định một mẫu và hai mẫu.
- **Các kiểm định phi tham số nâng cao:** Anderson-Darling, Mann-Whitney U, Kruskal-Wallis và các phương pháp hậu kiểm định.
- **Vấn đề đa so sánh:** Trình bày các phương pháp điều chỉnh mức ý nghĩa như Bonferroni, Holm, và Benjamini-Hochberg.

Về phân tích dữ liệu nhiều chiều

Đây là phần có tính ứng dụng cao nhất của bài thu hoạch:

- **Phân tích thành phần chính (PCA):** Từ lý thuyết toán học đến thuật toán thực hiện, các tiêu chí lựa chọn số thành phần và ứng dụng trong giảm chiều dữ liệu.
- **Phân tích nhân tố (Factor Analysis):** Mô hình nhân tố, các phương pháp ước lượng và kỹ thuật xoay nhân tố để tăng tính giải thích.
- **Phân tích tương quan chính tắc:** Khám phá mối liên hệ giữa hai nhóm biến và ứng dụng trong dự đoán.

- **Phân tích cụm:** K-means và clustering phân cấp với các tiêu chí đánh giá hiệu quả.
- **Phân tích phân biệt:** LDA và QDA cho bài toán phân loại có giám sát.
- **Ứng dụng tổng hợp:** Quy trình phân tích dữ liệu kinh tế-xã hội với code MATLAB chi tiết.

Về công cụ tính toán

Một điểm mạnh của bài thu hoạch là việc cung cấp các công cụ thực hành cụ thể:

- Hơn 20 hàm MATLAB được phát triển từ cơ bản đến nâng cao
- Các ví dụ minh họa chi tiết với dữ liệu thực tế
- Hướng dẫn từng bước thực hiện phân tích
- So sánh hiệu quả các phương pháp thông qua mô phỏng Monte Carlo

Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học

- Đóng góp vào việc phổ biến kiến thức thống kê nâng cao bằng tiếng Việt với cách trình bày có hệ thống và dễ hiểu.
- Kết nối lý thuyết với thực hành thông qua các ví dụ cụ thể và code minh họa.
- Cung cấp tài liệu tham khảo hữu ích cho sinh viên và nghiên cứu viên trong lĩnh vực thống kê và khoa học dữ liệu.
- Tổng hợp kiến thức từ nhiều nguồn uy tín thành một khối thống nhất.

Ý nghĩa thực tiễn

- **Trong giáo dục:** Có thể sử dụng làm tài liệu giảng dạy cho các môn học về thống kê nâng cao, phân tích dữ liệu nhiều chiều.

- **Trong nghiên cứu:** Cung cấp công cụ và phương pháp để phân tích dữ liệu trong các nghiên cứu khoa học, kinh tế, xã hội.
- **Trong ứng dụng:** Hướng dẫn thực hiện các phân tích thống kê trong doanh nghiệp, y tế, môi trường và các lĩnh vực khác.
- **Trong phát triển công nghệ:** Tạo nền tảng cho việc nghiên cứu và phát triển các thuật toán machine learning và AI.

Những hạn chế và tồn tại

Mặc dù đã đạt được những kết quả tích cực, bài thu hoạch vẫn còn một số hạn chế:

- **Về độ sâu:** Do giới hạn về thời gian và phạm vi, một số chủ đề chưa được khám phá đến mức độ sâu nhất, đặc biệt là các phương pháp thống kê Bayesian và các kỹ thuật machine learning hiện đại.
- **Về dữ liệu thực tế:** Các ví dụ chủ yếu sử dụng dữ liệu mô phỏng hoặc dữ liệu mẫu. Việc ứng dụng vào các bộ dữ liệu thực tế quy mô lớn còn hạn chế.
- **Về công cụ:** Tập trung chủ yếu vào MATLAB, chưa so sánh với các ngôn ngữ và công cụ khác như R, Python.
- **Về tính cập nhật:** Một số phương pháp mới nhất trong lĩnh vực chưa được đề cập đầy đủ.

Hướng phát triển trong tương lai

Dựa trên những kết quả đã đạt được và các hạn chế hiện tại, một số hướng phát triển tiềm năng:

Mở rộng nội dung

- **Thống kê Bayesian:** Phát triển chuyên sâu về inference Bayesian, MCMC, và các ứng dụng hiện đại.

- **Machine Learning thống kê:** Kết nối các phương pháp thống kê cổ điển với các thuật toán machine learning.
- **Big Data analytics:** Mở rộng các phương pháp cho dữ liệu quy mô lớn và tính toán phân tán.
- **Time series và spatial statistics:** Phân tích dữ liệu chuỗi thời gian và dữ liệu không gian.
- **Causal inference:** Các phương pháp suy luận nhân quả trong thống kê.

Cải thiện công cụ

- Phát triển package/toolbox hoàn chỉnh cho các phương pháp đã trình bày
- Xây dựng giao diện đồ họa (GUI) để dễ sử dụng hơn
- Tích hợp với các nền tảng cloud computing
- Phát triển song song trên R và Python

Ứng dụng thực tế

- Hợp tác với các doanh nghiệp để ứng dụng vào bài toán thực tế
- Phát triển case studies trong các lĩnh vực cụ thể: y tế, tài chính, marketing, môi trường
- Xây dựng cơ sở dữ liệu các bài toán và giải pháp mẫu
- Tích hợp vào các hệ thống business intelligence

Đào tạo và phổ biến

- Phát triển khóa học trực tuyến (MOOC) dựa trên nội dung bài thu hoạch
- Tổ chức workshop và seminar về thống kê nâng cao
- Xây dựng cộng đồng thực hành thống kê tại Việt Nam
- Phát triển chương trình đào tạo chuyên sâu cho các ngành nghề cụ thể

Lời kết

Thống kê nâng cao không chỉ là một lĩnh vực học thuật mà còn là công cụ thiết yếu trong việc hiểu và giải quyết các vấn đề phức tạp của thế giới hiện đại. Trong bối cảnh cuộc cách mạng 4.0 với sự bùng nổ của dữ liệu, việc nắm vững các phương pháp thống kê nâng cao trở nên quan trọng hơn bao giờ hết.

Bài thu hoạch này đã cố gắng cung cấp một cái nhìn toàn diện và có hệ thống về lĩnh vực thống kê nâng cao, từ những nền tảng lý thuyết cơ bản đến các ứng dụng thực tiễn phức tạp. Mặc dù còn nhiều hạn chế, chúng tôi hy vọng rằng công trình này sẽ góp phần vào việc phát triển và phổ biến kiến thức thống kê tại Việt Nam.

Thành công của việc ứng dụng thống kê nâng cao không chỉ phụ thuộc vào việc nắm vững lý thuyết mà còn cần sự kết hợp hài hòa giữa kiến thức toán học, kỹ năng lập trình, hiểu biết về lĩnh vực ứng dụng và khả năng tư duy phản biện. Đây chính là những thách thức và cơ hội cho thế hệ nghiên cứu viên và thực hành viên thống kê trong tương lai.

Chúng tôi tin rằng với sự phát triển không ngừng của công nghệ và nhu cầu ngày càng cao về phân tích dữ liệu, thống kê nâng cao sẽ tiếp tục đóng vai trò trung tâm trong việc thúc đẩy tiến bộ khoa học và công nghệ, góp phần xây dựng một xã hội dựa trên bằng chứng và ra quyết định thông minh.

Cuối cùng, chúng tôi mong muốn rằng bài thu hoạch này sẽ truyền cảm hứng cho những người yêu thích thống kê, khuyến khích họ tiếp tục khám phá và phát triển lĩnh vực đầy tiềm năng này. Thống kê nâng cao không chỉ là công cụ mà còn là nghệ thuật của việc khám phá tri thức từ dữ liệu - một kỹ năng vô cùng quý giá trong thế kỷ 21.

TÀI LIỆU THAM KHẢO

Tài liệu tham khảo

Tài liệu tiếng Việt

- [1] Hoàng Văn Phong (2021). *Lý thuyết xác suất và thống kê toán*, NXB Đại học Quốc gia TP. Hồ Chí Minh.
- [2] Đoàn Thị Thu Hương và Nguyễn Văn Tuấn (2019). Các phương pháp kiểm định phi tham số trong thống kê, *Tạp chí Khoa học ĐHQGHN: Toán học - Vật lý*, tập 35, số 2, trang 15–28.
- [3] Phạm Minh Đức (2020). *Phân tích dữ liệu đa biến với MATLAB và R*, Luận văn Thạc sĩ, Đại học Cần Thơ.
- [4] Trương Thị Mai Linh (2018). Ứng dụng kiểm định Kolmogorov-Smirnov trong kiểm tra tính phù hợp của phân phối, *Tạp chí Thống kê*, số 10, trang 32–40.
- [5] Vũ Thị Ngọc Anh (2022). Phân tích nhân tố khám phá (EFA) trong nghiên cứu khoa học xã hội, *Tạp chí Khoa học CTU*, tập 58, số 4, trang 78–89.

Tài liệu tiếng Anh

- [6] Anderson, T.W. (1984). *An Introduction to Multivariate Statistical Analysis*, 2nd Edition, John Wiley & Sons.
- [7] Casella, G. and Berger, R.L. (2002). *Statistical Inference*, 2nd Edition, Duxbury Press.

- [8] Conover, W.J. (1999). *Practical Nonparametric Statistics*, 3rd Edition, John Wiley & Sons.
- [9] Hair, J.F., Black, W.C., Babin, B.J., and Anderson, R.E. (2019). *Multivariate Data Analysis*, 8th Edition, Pearson Education.
- [10] Hollander, M., Wolfe, D.A., and Chicken, E. (2013). *Nonparametric Statistical Methods*, 3rd Edition, John Wiley & Sons.
- [11] Johnson, R.A. and Wichern, D.W. (2007). *Applied Multivariate Statistical Analysis*, 6th Edition, Pearson Education.
- [12] Kendall, M. and Stuart, A. (1973). *The Advanced Theory of Statistics*, Volume 2: Inference and Relationship, 3rd Edition, Griffin.
- [13] Lehmann, E.L. and Romano, J.P. (2005). *Testing Statistical Hypotheses*, 3rd Edition, Springer.
- [14] Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979). *Multivariate Analysis*, Academic Press.
- [15] Mood, A.M., Graybill, F.A., and Boes, D.C. (1974). *Introduction to the Theory of Statistics*, 3rd Edition, McGraw-Hill.
- [16] Sheskin, D.J. (2011). *Handbook of Parametric and Nonparametric Statistical Procedures*, 5th Edition, CRC Press.
- [17] Siegel, S. and Castellan, N.J. (1988). *Nonparametric Statistics for the Behavioral Sciences*, 2nd Edition, McGraw-Hill.
- [18] Tabachnick, B.G. and Fidell, L.S. (2019). *Using Multivariate Statistics*, 7th Edition, Pearson Education.
- [19] Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*, Springer.

Phụ lục A

Hướng dẫn trích dẫn tài liệu

			Ba tác giả
	Tác giả là một cơ quan, tổ chức	Ghi tên cơ quan và năm	
	Nhiều tài liệu	Sắp xếp các tài liệu theo năm xuất bản tăng dần. Nếu có	
			Nhiều tài liệu cùng cách trích dẫn
	Trích dẫn từ nguồn thứ cấp	Ghi tác giả và năm (nếu có) của tài liệu gốc kèm	
	Trích dẫn nguyên văn	Ghi tác giả, năm và trang viết. Đoạn trích dưới 40 từ:	