Câu 1:

1. Tính các giá trị

Trung bình:

A math equation with numbers and symbols

Description automatically generated

Trung vị:

* Sắp xếp trước khi tính
* N chẵn:

Lấy tổng 2 số giữa + vào chia đôi

* N lẻ:

Lấy luôn số chính giữa

Mode:

Số xuất hiện nhiều nhất nếu có nhiều hơn 1 cái

# b) Vẽ biểu đồ Boxplot

tính:

min

max

Vị trí các phân vị:

*Nếu ra số thập phân thì lấy trung bình giá trị của 2 vị trí trước và sau; nếu ra số nguyên thì là giá trị tại vị trí đó*

* Q1: .
* Q2: median
* Q3:

Tính IQR để tìm min max và giá trị ngoại lệ

IQR= Q3 - Q1

Khoảng giá trị hợp lệ: [ Q1 – 1,5.IQR ; Q3 + 1,5.IQR ]

# c, Chuẩn hoá

**Chuẩn hoá Min – Max**:



**Chuẩn hoá Z-score**:



SA = 

**Chuẩn hoá thập phân**:



Trong đó j là giá trị nhỏ nhất để 

Cho dãy sô 1, 5, 20, 70, 80

j=2 vì 80 < 100. Do giá trị lớn nhất là 80 nên lấy 102 là 100 giá trị gần nhất lớn hơn 80 vậy j=2.

# d, Bin

**Bin theo chiều rộng**:

Xét phần tử trong bin theo độ rộng mỗi bin, lấy (max-min) chia số bin để tìm số giá trị trong mỗi bin hoặc lấy (max-min) chia cho số giá trị trong mỗi bin để tìm số bin.



Ví dụ: 5, 10, 15, 20, 50, 60, 80. 4 bin

(80-5)/4= 18,75 xấp xỉ 19. Mỗi bin có độ rộng là 19

Bin 1: [5-24) 5,15,20

Bin 2: [24-43)

Bin 3: [43-62) 50,60

Bin 4: [62-81) 80

**Bin theo chiều sâu**:

Lấy số giá trị chia số số bin để có số giá trị mỗi bin hoặc số giá trị chia số giá trị mỗi bin để tìm bin



Ví dụ: 5, 10, 15, 20, 50, 60, 80. 4 bin

Số giá trị mỗi bin 7/4 = 1.75 **Làm tròn lên** giá trị trong mỗi bin là 2 2 2 1

Bin 1: 5,10

Bin 2: 15,20

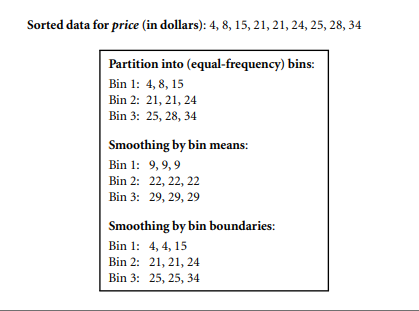
Bin 3: 50,60

Bin 4: 80

**Làm mịn:**

Làm mịn theo giá trị trung bình

Làm mịn theo theo giá trị biên



# e, Hệ số tương quan giữ X và Y





Câu 2:

**Apriori:**

A paper with writing on it

Description automatically generated

Tính |D| = 5 (Số bản ghi)

min\_sup = 60%

=> min\_sup\_count =  => Làm tròn lên

min\_sup\_count = 3

**Thuật toán Apriori:**

* Vòng lặp 1:

C1:

|  |  |
| --- | --- |
| Item | Sup\_count |
| I1 | 3 |
| I2 | 3 |
| I3 | 4 |
| I4 | 2 |
| I5 | 3 |
| I6 | 2 |
| I7 | 1 |

So sánh ≥ min\_sup\_count = 3

L1:

|  |  |
| --- | --- |
| Item | Sup\_count |
| I1 | 3 |
| I2 | 3 |
| I3 | 4 |
| I5 | 3 |

* Vòng lặp 2:

C2:

|  |  |
| --- | --- |
| Item | Sup\_count |
| {I1, I2} | 2 |
| {I1, I3} | 3 |
| {I1, I5} | 2 |
| {I2, I3} | 3 |
| {I2, I5} | 3 |
| {I3, I5} | 3 |

So sánh ≥ min\_sup\_count = 3

L2:

|  |  |
| --- | --- |
| Item | Sup\_count |
| {I1, I3} | 3 |
| {I2, I3} | 3 |
| {I2, I5} | 3 |
| {I3, I5} | 3 |

* Vòng lặp 3:

C3: Ghép các cặp mà cùng phần tử đầu

|  |  |
| --- | --- |
| Item | Sup\_count |
| {I2, I3, I5} | 3 |

So sánh ≥ min\_sup\_count = 3

L3:

|  |  |
| --- | --- |
| Item | Sup\_count |
| {I2, I3, I5} | 3 |

Vậy các tập phổ biến là L = L1 v L2 v L3

**FP-Growth:**

**Thuật toán FP-Growth:**

|  |  |
| --- | --- |
| Item | Sup\_count |
| I1 | 3 |
| I2 | 3 |
| I3 | 4 |
| I4 | 2 |
| I5 | 3 |
| I6 | 2 |
| I7 | 1 |

* Sắp xếp giảm dần theo sup\_count

|  |  |
| --- | --- |
| Item | Sup\_count |
| I3 | 4 |
| I1 | 3 |
| I2 | 3 |
| I5 | 3 |
| I4 | 2 |
| I6 | 2 |
| I7 | 1 |

* Loại bỏ những item < min\_sup\_count

|  |  |
| --- | --- |
| Item | Sup\_count |
| I3 | 4 |
| I1 | 3 |
| I2 | 3 |
| I5 | 3 |

* Viết lại bảng dữ liệu sau khi sắp xếp theo sup\_count

|  |  |
| --- | --- |
| ID | Items |
| T1 | I3, I1, I6 |
| T2 | I3, I1, I2, I5, I4 |
| T3 | I3, I2, I5 |
| T4 | I6, I7 |
| T5 | I3, I1, I2, I5, I4 |

* Vẽ FP-Tree:

A graph on a piece of paper

Description automatically generated

* Conditional pattern base (cơ sở mẫu điều kiện)

(Không xét node nối trực tiếp với NULL)

|  |  |
| --- | --- |
| Item | Conditional pattern base |
| I4 | { {I3, I1, I2, I5:2} } |
| I5 | { {I3, I1, I2:2}, {I3, I2:1} } |
| I6 | { {I3, I1:1} } |
| I7 | { {I6:1} } |
| I1 | { {I3:3} } |
| I2 | { {I3, I1:2}, {I3:1} } |

So sánh ≥ min\_sup\_count = 3

* Conditional FP-tree

(Lấy những dòng có tổng sup\_count ≥ 3)

|  |  |
| --- | --- |
| Item | Conditional FP-tree |
| I5 | { {I3:3}, {I2:3} } |
| I1 | { {I3:3} } |
| I2 | { {I3:3} } |

* Từ Conditional FP-tree, ta có các mẫu phổ biến:

L = { {I3, I5:3}, {I2, I5:3}, {I2, I3, I5:3}, {I1, I3:3}, {I2, I3:3}, {I1:3}, {I2:3}, {I3:3}, {I5:3}}