CHƯƠNG 3
LÀM SẠCH VÀ TIỀN XỬ LÝ DỮ LIỆU

Giảng viên: Nguyễn Anh Thư

Khoa: Khoa học ứng dụng

NỘI DUNG BÀI HỌC

3.1 THĂM DÒ DỮ LIỆU

3.1.1. Phân tích đơn biến – Univariate Analysis

3.1.2. Phân tích đa biến – Multivariate Analysis

3.1.3. Distribution Fitting

3.2 LÀM SẠCH DỮ LIỆU

3.2.1. Biến đổi kiểu dữ liệu

3.2.2. Xử lý dữ liệu thiếu và các ngoại lệ

3.2.3. Phát hiện và loại bỏ trùng lặp

3.3 TIỀN XỬ LÝ DỮ LIỆU

MỤC TIÊU BÀI HỌC

Sau khi học xong bài này, cần nắm được các vấn đề sau:

- Các phương pháp phân tích đơn biến, đa biến và kiểm tra phân phối phù hợp với dữ liệu.
- Các kỹ thuật cơ bản trong việc làm sạch dữ liệu.
- Sau khi khám phá được nội hàm bên trong dữ liệu, ta thực hiện tiền xử lý dữ liệu để chuẩn bị đưa vào khai thác sâu hơn (như xây dựng mô hình,…).

3.1 THĂM DÒ DỮ LIỆU

- Khi làm việc với một tập dữ liệu, bước đầu tiên là hiểu ý nghĩa và đặc điểm chính của nó, tức là các sự kiện hoặc đối tượng mà dữ liệu đề cập đến và các thuộc tính mà nó biểu thị, cần phải hiểu được nội hàm. Mục tiêu của việc thăm dò trên một tập dữ liệu bao gồm:
  - Xác định kiểu loại của tập dữ liệu (có cấu trúc, phi cấu trúc, bán cấu trúc).
  - Nếu dữ liệu không phải phi cấu trúc, cần hiểu được schema của nó (các thuộc tính cấu thành và cách tổ chức dữ liệu).
  - Xác định kiểu miền của từng thuộc tính (danh mục, thứ tự, hay số) và đặc điểm của giá trị trong miền đó (giá trị điển hình, phạm vi giá trị, v.v.).

3.1 THĂM DÒ DỮ LIỆU

- Mục tiêu trong phần này là làm việc với dữ liệu có cấu trúc và lược đồ đã biết.
- Các bước:
  1. Phân tích từng thuộc tính riêng lẻ được gọi là phân tích đơn biến (univariate analysis) trong thống kê, tập trung vào một biến tại một thời điểm.
  2. Phân tích các mối quan hệ giữa các thuộc tính được gọi là phân tích đa biến trong thống kê. Thông thường một thuộc tính trong dữ liệu thường có liên quan đến các thuộc tính khác vì mỗi thuộc tính mô tả một khía cạnh/ đặc điểm của đối tượng, sự kiện nên nhiều khi chúng thể hiện một số đặc điểm chung.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

- Cần xác định loại dữ liệu của thuộc tính (domain type) như categorical (danh mục), numerical (số), hoặc ordinal (thứ tự). Ngoài ra, cần xem xét liệu kiểu dữ liệu được gán cho thuộc tính (ví dụ, string, number, date) có phù hợp với loại dữ liệu của nó không.
- Ví dụ:
  - Các categorical có thể được mã hóa bằng số ('1', '2', ...), nhưng không có ý nghĩa số học hoặc thứ tự.
  - Thông tin thời gian (temporal data) thường bị nhập dưới dạng chuỗi thay vì kiểu dữ liệu ngày/thời gian, điều này gây khó khăn cho việc phân tích.
  - Thuộc tính dạng số không nên được lưu dưới dạng chuỗi (string), hoặc ngược lại, ngày tháng không nên lưu dưới dạng số.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

- Thường đánh giá về:
  - Khoảng giá trị (min, max).
  - Xu hướng trung tâm (mean, median, mode).
  - Độ phân tán (độ lệch chuẩn, phân vị,…).

➔ Với các thước đo này ta sử dụng các hàm AGGREGATE FUNCTION.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình - MEAN**

```
SELECT Avg(Attr) as mean
FROM Data;
```

≡

```
SELECT Sum(Attr) /
(Count(Attr) * 1.0) as mean
FROM Data;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình - MEAN**

VD1: Từ bảng StudentScores(student_id, student_name, score, exam_date, subjects_enrolled) như dưới đây, hãy tính điểm trung bình của các sinh viên.

student_id	student_name	score	exam_date	subjects_enrolled
1	Nguyen Van A	8.5	2024-01-15	3
2	Tran Thi B	9	2024-01-15	1
3	Le Van C	7	2024-01-15	2
4	Pham Thi D	8	2024-01-16	5
5	Hoang Van E	7.5	2024-01-16	4

student_id	student_name	score	exam_date	subjects_enrolled
6	Dang Thi F	6.5	2024-01-16	2
7	Vo Van G	9.5	2024-01-17	1
8	Bui Thi H	8	2024-01-17	3
9	Do Van I	7.5	2024-01-17	3
10	Hoang Bach K	7.2	2024-01-16	5

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình - MEAN**

VD1: Từ bảng StudentScores(student_id, student_name, score, exam_date, subjects_enrolled) như dưới đây, hãy tính điểm trung bình của các sinh viên.

Cách 1:

```
SELECT Avg(score) as mean
FROM StudentScores;
```

Cách 2:

```
SELECT Sum(score) / (Count(score) * 1.0) as mean
FROM StudentScores;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình - MEAN**

- Trong xác suất thống kê, nếu một giá trị x_i xuất hiện f_i lần trong tổng số N quan sát.
- Khi đó, xác suất thực nghiệm là:

$$P(x_i) \approx \frac{f_i}{N} = \frac{f_i}{\sum f_i} \qquad (1)$$

$$\sum x_i \cdot P(x_i) = \sum \left(x_i \cdot \frac{f_i}{\sum f_i} \right) \qquad (2)$$

➔ (2) là công thức TB KỲ VỌNG $\mathrm{E(X)} = \sum x_i \cdot P(x_i)$

3.1 THĂM DÒ DỮ LIỆU

❖ **Giá Trị Trung Bình –** Sử dụng xác suất

```
WITH NHistogram(Value, Prob) AS
(       SELECT Attr, sum(1.0/total)
        FROM  Data,  (SELECT  COUNT(*)  AS
                total FROM Data) AS Temp
        GROUP BY Attr
)
SELECT Sum(Value * Prob) AS mean
FROM NHistogram;
```

❖ **Giá Trị Trung Bình –** Sử dụng tần số

```
WITH Histogram(Value, Frequency) AS
(       SELECT Attr, count(*)
        FROM Data
        GROUP BY Attr
)
SELECT (1.0 * Sum(Value * Frequency)) /
Sum(Frequency) AS mean
FROM Histogram;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình - MEAN**

VD2: Từ bảng StudentScores(student_id, student_name, score, exam_date, subjects_enrolled) như dưới đây, tính số môn học trung bình mà các sinh viên tham gia (**sử dụng 2 cách xác suất và tần số**, biết số môn học tối thiểu mà sinh viên tham gia nằm trong khoảng từ 1 đến 4).

student_id	student_name	score	exam_date	subjects_enrolled
1	Nguyen Van A	8.5	2024-01-15	3
2	Tran Thi B	9	2024-01-15	1
3	Le Van C	7	2024-01-15	2
4	Pham Thi D	8	2024-01-16	5
5	Hoang Van E	7.5	2024-01-16	4

student_id	student_name	score	exam_date	subjects_enrolled
6	Dang Thi F	6.5	2024-01-16	2
7	Vo Van G	9.5	2024-01-17	1
8	Bui Thi H	8	2024-01-17	3
9	Do Van I	7.5	2024-01-17	3
10	Hoang Bach K	7.2	2024-01-16	5

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình - MEAN**

VD2: *Cách 1:*

```
WITH NHistogram(Value, Prob)
AS ( SELECT Subjects_Enrolled, sum(1.0/total)
     FROM StudentScores, (SELECT COUNT(*) AS total FROM StudentScores) AS Temp
     GROUP BY Subjects_Enrolled )
SELECT Sum(Value * Prob) AS mean FROM NHistogram;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình - MEAN**

VD2: *Cách 2:*

```
WITH Histogram(Value, Frequency)
AS ( SELECT Subjects_Enrolled, count(*)
     FROM StudentScores GROUP BY Subjects_Enrolled )
SELECT (1.0 * Sum(Value * Frequency)) / Sum(Frequency) AS mean
FROM StudentScores;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – TRIMMED MEAN**

- GTTB rất nhạy bởi các giá trị ngoại lai (outlier) và các giá trị cực trị.
- Ví dụ nếu có một giá trị rất cao hoặc rất thấp thì nó sẽ ảnh hưởng rất lớn đến giá trị trung bình.

➔ Sử dụng TRIMMED MEAN – giá trị trung bình được cắt tỉa: GTTB sẽ được tính toán sau khi bỏ qua các giá trị cực trị, thường là GTNN và GTLN, hoặc khái quát hơn là loại bỏ k% giá trị cao nhất/ thấp nhất.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – TRIMMED MEAN**

Loại bỏ các giá trị cực trị là GTLN và GTNN

```
SELECT avg(A)
FROM Data,
    (SELECT max(A) AS Amax FROM Data) AS T1,
    (SELECT min(A) AS Amin FROM Data) AS T2
WHERE A < Amax and A > Amin;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – TRIMMED MEAN**

Loại bỏ các giá trị cực trị là GTLN và GTNN

```
SELECT AVG(score)
FROM   studentscores,
       (SELECT min(score) As min_score FROM
              studentscores) AS T1,
       (SELECT max(score) AS max_score FROM
              studentscores) as T2
WHERE min_score < score AND score < max_score;
```

avg(score)
7.837499976158142

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – TRIMMED MEAN**

Loại bỏ k% GTLN và GTNN:

Bước 1. Sắp xếp dữ liệu theo chiều tăng/giảm dần giá trị:

```
WITH RankedData
AS (SELECT A,
    ROW_NUMBER() OVER (ORDER BY A ASC) AS rank_asc,
    ROW_NUMBER() OVER (ORDER BY A DESC) AS rank_desc
    FROM Data)
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – TRIMMED MEAN**

Loại bỏ k% GTLN và GTNN:

Bước 2. Loại bỏ k% phần tử là GTLN và GTNN:

```
WITH TrimmedData
AS (SELECT A
    FROM RankedData
    WHERE rank_asc > (SELECT COUNT(*) * 0.01 * k FROM Data)
    AND rank_desc > (SELECT COUNT(*) * 0.01 * k FROM Data)
)
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – TRIMMED MEAN**

Loại bỏ k% GTLN và GTNN:

Bước 3. Tính TRIMMED MEAN:

```
SELECT AVG(A) AS TrimmedMean
FROM TrimmedData;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ Giá Trị Trung Bình – GEOMETRIC MEAN

- Geometric mean (trung bình nhân) có công thức tổng quát như sau: $\sqrt[n]{x_1 * x_2 * \cdots * x_n}$.
- Ưu điểm: ít nhạy cảm với các giá trị ngoại lại so với trung bình cộng, đặc biệt với các giá trị lớn.
- Nhược điểm: vẫn bị ảnh hưởng bởi các giá trị nhỏ.
- SQL không có hàm AGGREGATE để tính trực tiếp.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – GEOMETRIC MEAN** $\sqrt[n]{x_1 * x_2 * \cdots * x_n}$.

- **Cách 1**: giữ nguyên công thức

Bước 1: Sử dụng hàm logarit để tính tích:

$$\boldsymbol{log(a * b) = log\, a + log\, b}$$

$$\Rightarrow \boldsymbol{a * b = \exp(\log a + \log b)}$$

```
SELECT exp(sum(log(Attr))
FROM Data;
```

Bước 2: Sử dụng hàm POW trong SQL với số mũ là $\frac{1}{n}$ để tính căn.

```
SELECT pow(exp(sum(log(Attr)), 1.0/total)
FROM Data, (SELECT count(Attr) AS total FROM Data);
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Giá Trị Trung Bình – GEOMETRIC MEAN** $\sqrt[n]{x_1 * x_2 * \cdots * x_n}$.

- **Cách 2**: biến đổi thành công thức tương đương $\sqrt[n]{\boldsymbol{x_1 * x_2 * \cdots * x_n}} = \boldsymbol{exp}\,\frac{\boldsymbol{ln\, x_1 + ln\, x_2 + \cdots + ln\, x_n}}{\boldsymbol{n}}$.

```
SELECT exp(sum(log(Attr)) / count(Attr))
FROM Data;
```

Vì sum/count = average nên câu lệnh trên có thể thay thế thành:

```
SELECT exp(avg(log(Attr))
FROM Data;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ Giá Trị Trung Bình – GEOMETRIC MEAN

- Khi nào thì nên sử dụng trung bình nhân?
  - Hữu ích khi tính toán tỷ lệ lãi suất trong ngân hàng.
  - Đo lường tốc độ tăng trưởng/ suy giảm trung bình trong tài chính, kinh tế và sinh học.
  - Phù hợp với các tường hợp có tỷ lệ thay đổi mạnh hoặc dữ liệu có biến động lớn.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **MODE**

- Là giá trị xuất hiện nhiều nhất trong dữ liệu (có thể có nhiều mode).
- Trong SQL không có hàm tính mode mà phải tính toán theo các bước như sau:

```
WITH Histogram AS
 (SELECT Value AS val, count(*) AS freq FROM Data GROUP BY Attr)
SELECT val
FROM Histogram, (SELECT max(freq) AS top FROM Histogram) AS T
WHERE freq = top;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ MODE

- **VD3**: Trong các chuyến bay đến NY, hãy tính điểm đến (destination) được ghé thăm nhiều nhất (điểm đến với nhiều chuyến bay đến đó).

FlightID	Origin	Destination	FlightDate
1	NY	Los Angeles	2024-02-01
2	NY	Chicago	2024-02-02
3	NY	Los Angeles	2024-02-03
4	NY	San Francisco	2024-02-04
5	NY	Chicago	2024-02-05

FlightID	Origin	Destination	FlightDate
6	NY	Los Angeles	2024-02-06
7	NY	Miami	2024-02-07
8	NY	San Francisco	2024-02-08
9	NY	Los Angeles	2024-02-09
10	NY	Chicago	2024-02-10

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ MODE

- **VD3**: Trong các chuyến bay đến NY, hãy tính điểm đến (destination) được ghé thăm nhiều nhất (điểm đến với nhiều chuyến bay đến đó).

  Cách 1:

```
SELECT Destination, COUNT(*) AS FlightCount
FROM Flights
GROUP BY Destination
ORDER BY FlightCount DESC
LIMIT 1;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ MODE

- **VD3**: Trong các chuyến bay đến NY, hãy tính điểm đến (destination) được ghé thăm nhiều nhất (điểm đến với nhiều chuyến bay đến đó).

Cách 2:

```
WITH Histogram AS  (SELECT Destination AS val, count(*) AS freq
                    FROM Flights GROUP BY Destination)
SELECT val, COUNT(*) AS FlightCount
FROM Histogram, (SELECT max(freq) AS top FROM Histogram) AS T
WHERE freq = top;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ MEDIAN

- Là giá trị xuất hiện ở vị trí giữa khi các giá trị được sắp xếp theo thứ tự nhỏ đến lớn sao cho số lượng số bên trái và phải nó là bằng nhau, được gọi là trung vị.
  - ➢ Nếu số lượng phần tử là lẻ, trung vị là giá trị ở giữa.
  - ➢ Nếu số lượng phần tử là chẵn, trung vị là trung bình của hai giá trị ở giữa.
- Trung vị ít bị ảnh hưởng bởi các giá trị ngoại lai hơn so với trung bình cộng.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ MEDIAN

- Tính trung vị trong SQL khá phức tạp do cần sắp xếp các giá trị (ORDER BY) và xác định vị trí ở giữa, thường sử dụng kết hợp LIMIT và OFFSET để lấy giá trị mong muốn:
  - Xác định tổng số phần tử `size`.
  - Sắp xếp dữ liệu sử dụng `ORDER BY`.
  - `MOD(size, 2)` kiểm tra tổng số phần tử chẵn hay lẻ .
  - Xác định vị trí bắt đầu của trung vị bằng cách
    - Xác định vị trí chính giữa của DL `CEIL(size / 2.0)`
    - `OFFSET` dịch con trỏ đến vị trí chính giữa của DL.
    - Tính trung bình của các phần tử chính giữa.

```
SELECT avg(Attr)
FROM (SELECT Attr FROM Data,
      (SELECT count(*) as size FROM Data)
      ORDER BY value
      LIMIT 2 - MOD(size, 2)
      OFFSET CEIL(size / 2.0) ) AS T;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ **Mức Độ Phân Tán**

- Đơn giản nhất là khoảng giá trị (range) giữa GTLN và GTNN, đã có sẵn trong SQL thông qua hàm MAX() và MIN().
- Độ lệch chuẩn (standard deviation) thường sử dụng phổ biến để đo độ phân tán, được tích hợp sẵn trong hệ quản trị CSDL thông qua hàm STD(). Công thức :

$$\sqrt{\frac{(\Sigma_{i=1}^{n} x_i^2)}{(n-1)} - \left(\frac{(\Sigma_{i=1}^{n} x_i)}{(n-1)}\right)^2}$$

- Phương sai (variance) cũng cần để phân tích trong một số trường hợp, được xác định là bình phương của độ lệch chuẩn, là một phép toán được tích hợp sẵn trong hầu hết các hệ thống thông qua hàm VARIANCE().

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ SKEWNESS VÀ KURTOSIS

- Skewness (Độ nghiêng/ Độ chệch) được sử dụng để đo lường sự đối xứng của phân phối.
- Phân phối đối xứng khi (mean) bằng giá trị trung vị (median), với độ nghiêng bằng 0, trong đó phân phối lệch phải (skew dương) là phân phối có đuôi kéo dài về bên phải, phân phối lệch trái (skew âm) có đuôi kéo dài về bên trái. Độ chệch này cho biết phân phối có cân bằng xung quanh GTTB hay không.

$$Skewness = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^3}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right)^{\frac{3}{2}}}$$

- n là số lượng giá trị,
- x_i là giá trị từng biến và $\overline{x}$ là GTTB.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ SKEWNESS VÀ KURTOSIS

- Kurtosis (độ nhọn) sử dụng để đo lường mức độ tập trung của các giá trị xa TB (đuôi phân phối), giúp xác định khả năng xuất hiện, xu hướng phân phối tạo ra các giá trị outlier.

$$Kurtosis = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^4}{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2\right)^2}$$

- n là số lượng giá trị,
- x_i là giá trị từng biến và $\overline{x}$ là GTTB.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

a. VỚI CÁC THUỘC TÍNH KIỂU SỐ

❖ SKEWNESS VÀ KURTOSIS

- Kurtosis (độ nhọn) sử dụng để đo lường mức độ tập trung của các giá trị xa TB (đuôi phân phối), giúp xác định khả năng xuất hiện, xu hướng phân phối tạo ra các giá trị outlier.
- Kurtosis thường được so sánh với 3 (kurtosis chuẩn của phân phối chuẩn).
  - ➢ Kurtosis > 3: phân phối có đuôi dày hơn, dễ xuất hiện giá trị ngoại lai (leptokurtic).
  - ➢ Kurtosis = 3: phân phối chuẩn (mesokurtic).
  - ➢ Kurtosis < 3: đuôi mỏng hơn, ít ngoại lai hơn (platykurtic)

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- Khi làm việc với thuộc tính dạng phân loại, công cụ quan trọng nhất trong trường hợp này là histogram – biểu đồ tần suất. Ở dạng đơn giản nhất của histogram, một tập các giá trị trong tập dữ liệu đi kèm với tần suất xuất hiện của chúng, đây là cách biểu diễn đơn giản và trực quan để hiểu sự phân bố của các giá trị thuộc tính phân loại.

- Một cách đơn giản để xây dựng một histogram của một thuộc tính phân loại Attr như sau:

```
SELECT Attr, count(*)

FROM table

GROUP BY Attr;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Binning** là một trong những kỹ thuật tổng quát hơn histogram:
  - Các giá trị của một biến được chia thành các khoảng rời rạc gọi là bin/ bucket, các giá trị rơi vào cùng một khoảng sẽ được thay thế bằng một giá trị đại diện (representative value)
  - Với các biến liên tục: các giá trị được nhóm thành các khoảng và thường là các khoảng giá trị liên tiếp.
  - Với biến phân loại: mỗi giá trị phân loại sẽ ứng với một bin riêng biệt, (có thể nhóm nhiều giá trị phân loại vào cùng một khoảng nếu cần).

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Binning** là một trong những kỹ thuật tổng quát hơn histogram:

Histogram	Binning
Tần suất xuất hiện đại diện cho một bin.	Một giá trị thống kê (GTTB, tổng,…) đại diện cho một bin.

- Binning linh hoạt và hữu ích để xử lý và tổng hợp dữ liệu, đặc biệt là khi làm việc với các biến liên tục hay cần đơn giản hóa dữ liệu.
- Histogram là một dạng cụ thể của binning, trong đó giá trị đại diện là tần suất xuất hiện.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- Để thực hiện binning tổng quát trong SQL, các giá trị cần được chia thành các khoảng và số lượng khoảng cần tạo. Có 2 phương pháp chính:

  - **Equi-depth histogram:** giới hạn các bin sao cho mỗi bin chứa số lượng điểm dữ liệu bằng nhau
  - **Equi-space histogram:** tất cả các bin có cùng độ rộng (cố định độ rộng). Với các thuộc tính phân loại, mỗi khoảng sẽ chứa cùng số lượng giá trị.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Equi-depth histogram:** xác định các phân vị (quantiles) của dữ liệu, là các điểm cắt chia miền giá trị thành các khoảng có cùng số điểm dữ liệu:

  - **Percentiles**: Chia miền giá trị thành 100 khoảng, mỗi khoảng chứa 1% dữ liệu.
  - **Quartiles**: Chia miền giá trị thành 4 khoảng, tương ứng với các mức 25%, 50%, 75%, và 100% dữ liệu.
  - **Deciles**: Chia miền giá trị thành 10 khoảng, tương ứng với các mức 10%, 20%, ..., và 100% dữ liệu.
  - **Median** (trung vị) cũng được xem là một dạng 2-quantile, vì nó chia dữ liệu thành 2 khoảng có số lượng điểm bằng nhau.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Equi-depth histogram:**

VD5: giả sử ta có bảng Heights(age, size) với age là số nguyên, hãy tạo histogram cho thuộc tính age dựa trên quartiles, nghĩa là bin đầu tiên đại diện cho 25% lứa tuổi đầu tiên, bin thứ hai đại diện cho 25-50% lứa tuổi tiếp theo, bin thứ 3 từ 50-75% và bin cuối trên 75% lứa tuổi còn lại.

Gợi ý: đầu tiên ta cần xem có bao nhiêu phần tử và xác định 4 nhóm. Sau đó chia tất cả các phần tử này vào 4 nhóm sau khi đã sắp xếp chúng.

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Equi-depth histogram:**

VD5:

```
WITH OrderedData AS (
    SELECT *, NTILE(4) OVER (ORDER BY age)
    AS quartile FROM Heights )
SELECT
    CASE
        WHEN quartile = 1 THEN 'bottom25'
        WHEN quartile = 2 THEN '25to50'
        WHEN quartile = 3 THEN '50to75'
        WHEN quartile = 4 THEN 'top25'
    END AS quartile, age, size
FROM OrderedData
ORDER BY quartile, age;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Equi-space histogram:** Với thuộc tính phân loại, mỗi khoảng sẽ chứa cùng số lượng giá trị. Phương pháp này thường chỉ áp dụng cho miền dữ liệu thứ tự và dạng số. Độ rộng của các khoảng sẽ quyết định số lượng bin với một miền dữ liệu số nhất định (D). Nếu h là độ rộng cố định, thì số lượng bin sẽ được xác định bởi công thức:

$$number\ of\ bins = \left\lceil \frac{\max(D) - \min(D)}{h} \right\rceil$$

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Equi-space histogram:**

VD6: bảng Heights(age, size) với age là số nguyên, hãy tạo histogram cho thuộc tính age trong đó mỗi khoảng có độ rộng cố định là 4.

```
SELECT age, size, ceil(((age-minage)+1)/4) AS bin
FROM Heights, (SELECT min(age) AS minage FROM Heights) AS T
ORDER BY bin;
```

→ Ánh xạ mỗi giá trị vào một bin bắt đầu từ bin 1, tuổi nhỏ nhất được ánh xạ đến bin 1 bởi (age-minage)+1), nhỏ thứ hai đến bin 2, …

3.1 THĂM DÒ DỮ LIỆU

3.1.1 PHÂN TÍCH ĐƠN BIẾN

b. VỚI CÁC THUỘC TÍNH KIỂU PHÂN LOẠI

- **Equi-space histogram**:
  - Phân chia các bin quá rộng → một vài bin sẽ ẩn đi các đặc điểm quan trọng của dữ liệu.
  - Phân chia các bin quá hẹp → số lượng bin quá lớn, làm cho dữ liệu khá bất thường nếu nó không phù hợp với bất kỳ phân phối nào đã biết.
- Một số quy tắc có thể sử dụng để chọn số lượng khoảng như sau:
  - Nếu có n điểm DL, chọn $\sqrt{n}$ khoảng cho histogram.
  - Quy tắc Sturges: nếu có n điểm DL , chọn số lượng bin là $\lceil \log_2 n + 1 \rceil$.

3.1 THĂM DÒ DỮ LIỆU

3.1.2 PHÂN TÍCH ĐA BIẾN

- Giả sử A và B là hai thuộc tính trong một bảng và một trong hai cột phụ thuộc (bị ảnh hưởng) vào cột còn lại. Trong trường hợp này ta nói rằng có một thuộc tính là độc lập (trong thống kê nó là biến dự đoán) và một thuộc tính phụ thuộc (biến đầu ra trong thống kê).

- Tùy thuộc kiểu dữ liệu của hai thuộc tính mà ta có 3 trường hợp như sau:
  - Cả hai đều là số.
  - Cả hai đều là phân loại.
  - Một phân loại kết hợp với một số.

- Thực tế, ta không biết liệu hai thuộc tính có độc lập hay không → phân tích với một số thử nghiệm đơn giản để xác định có mối liên hệ nào giữa các thuộc tính hay không.

3.1 THĂM DÒ DỮ LIỆU

3.1.2 PHÂN TÍCH ĐA BIẾN

- Cách đơn giản nhất có thể thử nghiệm được với **mọi loại thuộc tính** đó là sử dụng xác suất của chúng.

- A và B được gọi là độc lập nhau nếu:

$$P(A,B) = P(A)P(B)$$

- Trong SQL ta có thể xác định như sau như sau: giả sử ta có Data(A,B)

  - Xác định P(A):

```
SELECT A, sum(1.0/total) as PrA
FROM Data, (SELECT count(*) as total FROM Data) as T
GROUP BY A;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.2 PHÂN TÍCH ĐA BIẾN

- Trong SQL ta có thể xác định như sau như sau: giả sử ta có Data(A,B)
  - Xác định P(B):

```
SELECT B, sum(1.0/total) as PrB
FROM Data, (SELECT count(*) as total FROM Data) as T
GROUP BY B;
```

  - Xác định P(A, B):

```
SELECT A, B, sum(1.0/total) as PrAB
FROM Data, (SELECT count(*) as total FROM Data) as T
GROUP BY A, B;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.2 PHÂN TÍCH ĐA BIẾN

- Trong SQL ta có thể xác định như sau như sau: giả sử ta có Data(A,B)
  - Tính toán sự phụ thuộc theo công thức xác suất đồng thời:

```
SELECT sum(PrAB - (PrA * PrB))
FROM (      SELECT A, B, PrA, PrB, PrAB
            FROM ProbA, ProbB, ProbAB
            WHERE ProbA.A = ProbAB.A and ProbB.B = ProbAB.B
    ) AS Probabilities;
```

3.1 THĂM DÒ DỮ LIỆU

3.1.2 PHÂN TÍCH ĐA BIẾN

- Trong SQL ta có thể xác định như sau như sau: giả sử ta có Data(A,B)
  - Một cách khác để kiểm tra sự độc lập của các thuộc tính bằng cách sử dụng PMI (Pointwise Mutual Information) được xác định như sau:

$$PMI(A,B) = \log \frac{P(A,B)}{P(A)P(B)}$$

  - Nếu PMI(A, B)=0 thì chúng độc lập với nhau.

3.1 THĂM DÒ DỮ LIỆU

3.1.2 PHÂN TÍCH ĐA BIẾN

a. TRƯỜNG HỢP: CẢ HAI THUỘC TÍNH LÀ KIỂU SỐ

- **Covariance – hiệp phương sai** – là thước đo đơn giản nhất về mối liên hệ có thể có giữa các thuộc tính, nếu Cov(A, B) = 0 thì chúng độc lập với nhau. Hiệp phương sai được xác định bởi:

$$Cov(A,B) = E[(A - E(A))(B - E(B))]$$

với A, B là các thuộc tính, E là trung bình kỳ vọng.

- Một số công thức tính hiệp phương sai tương đương:

$$Cov(A,B) = \frac{\sum_{i=1}^{N}(a_i - \bar{A})(b_i - \bar{B})}{N-1} \quad (1)$$

$$= \frac{\sum_{i=1}^{N}(a_i b_i)}{N} - \frac{\sum_{i=1}^{N}(a_i)\sum_{i=1}^{N}(b_i)}{N(N-1)} \quad (2)$$

3.1 THĂM DÒ DỮ LIỆU

3.1.2 PHÂN TÍCH ĐA BIẾN

a. TRƯỜNG HỢP: CẢ HAI THUỘC TÍNH LÀ KIỂU SỐ

- **Covariance – hiệp phương sai**

VD7: Bảng Data(ID, Num1, Num2) có dữ liệu như sau, hãy tính Cov(Num1, Num2)

ID	Num1	Num2
1	3.5	7.2
2	5.1	8.4
3	7.8	6.9
4	9.0	10.2
5	11.3	4.5
6	13.7	12.8