HUTECH
Đại học Công nghệ Tp.HCM

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP.HCM

ĐỒ ÁN MÔN HỌC

PHÂN CỤM KHÁCH HÀNG BẰNG MÔ HÌNH RFM KẾT HỢP THUẬT TOÁN K-MEANS VÀ SBSCAN

Ngành: KHOA HỌC DỮ LIỆU
Môn học: PHÂN TÍCH VÀ TRỰC QUAN DỮ LIỆU

Giảng viên hướng dẫn: LÊ NHẬT TÙNG
Sinh viên thực hiện:

Nguyễn Nhật Nam	MSSV: 2286400019
Hoàng Quang Minh	MSSV: 2286400017
Hà Thế Anh	MSSV: 2286400002

Lớp: 22DKHA1

TP. Hồ Chí Minh, 2025

MỤC LỤC

LỜI CAM ĐOAN

Chúng tôi, Hoàng Quang Minh, Nguyễn Nhật Nam và Hà Thế Anh xin cam đoan rằng:

Mọi thông tin và nghiên cứu được trình bày trong bài báo cáo này là trung thực và khách quan được thu thập và phân tích một cách cẩn thận dựa trên các nguồn chính thống và đáng tin cậy.

Bất kỳ thông tin hoặc ý kiến nào được trích dẫn từ các nguồn khác đều được nêu rõ nguồn gốc và được trích dẫn theo đúng quy định. Chúng tôi xin cam đoan rằng không có bất kỳ sự sao chép hoặc sử dụng thông tin không đúng đắn nào từ các nguồn khác.

Bài báo cáo này là công trình nghiên cứu độc lập của chúng tôi chưa từng được công bố ở bất kỳ nơi nào khác. Tôi cam đoan rằng đã tuân thủ đầy đủ các quy tắc và quy định của môn học bao gồm cả việc tham khảo và sử dụng công cụ nghiên cứu.

Tôi hy vọng rằng bài báo cáo này sẽ cung cấp một cái nhìn tổng quan rõ ràng và toàn diện về chủ đề "Ứng dụng RFM,K - Means và DBSCAN trong phân khúc khách hàng và tối ưu hóa doanh thu" và sẽ đóng góp một phần nhỏ vào lĩnh vực nghiên cứu này.

TP. Hồ Chí Minh, ngày 15 tháng 4 năm 2025

Sinh viên

Hoàng Quang Minh
Nguyễn Nhật Nam
Hà Thế Anh

CHƯƠNG 1: TỔNG QUAN

1.1. Giới thiệu đề tài

Trong bối cảnh cạnh tranh ngày càng khốc liệt, việc thấu hiểu khách hàng trở thành yếu tố then chốt giúp doanh nghiệp duy trì và phát triển mối quan hệ lâu dài. Một trong những công cụ hiệu quả để thực hiện điều này là mô hình RFM – viết tắt của Recency (Gần đây), Frequency (Tần suất), và Monetary (Giá trị chi tiêu). Mô hình RFM cho phép doanh nghiệp đánh giá hành vi tiêu dùng của khách hàng dựa trên ba yếu tố cốt lõi, từ đó thực hiện phân khúc khách hàng một cách trực quan và có căn cứ dữ liệu.

Trong đề tài này, chúng tôi ứng dụng mô hình RFM kết hợp với thuật toán phân cụm K-Means để phân nhóm khách hàng trên môi trường ngôn ngữ R. Việc này không chỉ hỗ trợ doanh nghiệp xác định nhóm khách hàng mục tiêu mà còn tối ưu hóa chiến lược tiếp thị, giữ chân khách hàng và tăng trưởng doanh thu. Báo cáo được thực hiện trên bộ dữ liệu giao dịch thực tế, minh họa quy trình từ tiền xử lý dữ liệu, xây dựng thang đo RFM, đến trực quan hóa và phân tích kết quả phân cụm.

1.2. Nhiệm vụ của đồ án

- **Khám phá và hiểu dữ liệu:**
  - Đọc dữ liệu giao dịch từ file Excel.
  - Tìm hiểu cấu trúc dữ liệu, mô tả các biến và kiểm tra dữ liệu thiếu.
- **Tiền xử lý dữ liệu:**
  - Làm sạch dữ liệu: loại bỏ bản ghi thiếu mã khách hàng hoặc số lượng âm.
  - Tạo biến mới như “Tổng chi tiêu” để phục vụ cho bước tính RFM.
- **Xây dựng mô hình RFM:**
  - Tính toán ba chỉ số RFM cho từng khách hàng:
    - Recency: Số ngày kể từ giao dịch gần nhất.
    - Frequency: Số lần giao dịch.
    - Monetary: Tổng giá trị chi tiêu.
  - Tạo bảng tổng hợp RFM cho toàn bộ khách hàng.
- **Chuẩn hóa dữ liệu và phân cụm:**
  - Chuẩn hóa các giá trị RFM để đưa về cùng thang đo.
  - Xác định số cụm tối ưu bằng Elbow method và Silhouette score.
  - Áp dụng thuật toán K-Means và DBScan để phân cụm khách hàng.
- **Trực quan hóa và phân tích kết quả:**
  - Vẽ biểu đồ để quan sát sự phân bố khách hàng theo cụm.
  - Phân tích đặc điểm từng cụm dựa trên giá trị trung bình của các chỉ số RFM.

1.3. Tính cấp thiết của đề tài

Trong bối cảnh thị trường cạnh tranh ngày càng khốc liệt, việc thấu hiểu hành vi khách hàng không chỉ là lợi thế mà còn là điều kiện sống còn để doanh nghiệp duy trì và phát triển. Thay vì áp dụng các chiến lược tiếp thị đại trà, doanh nghiệp hiện nay có xu hướng cá nhân hóa trải nghiệm nhằm tăng mức độ gắn bó và giá trị vòng đời khách hàng.

Tuy nhiên, khối lượng dữ liệu khách hàng ngày càng lớn, đa dạng và phức tạp, khiến cho các phương pháp phân tích truyền thống không còn đáp ứng được nhu cầu khai thác sâu. Việc ứng dụng mô hình RFM (Recency - Frequency

- Monetary) kết hợp với thuật toán phân cụm K-Means và DBScan là một giải pháp phù hợp, giúp doanh nghiệp phân khúc khách hàng một cách khoa học và dựa trên dữ liệu thực tiễn.

Việc triển khai đề tài trên nền tảng R không chỉ tận dụng sức mạnh phân tích dữ liệu và trực quan hóa mà còn rèn luyện kỹ năng khai phá dữ liệu – một năng lực quan trọng trong thời đại chuyển đổi số. Do đó, việc thực hiện đề tài "Ứng dụng RFM trong phân khúc khách hàng" mang tính cấp thiết và có ý nghĩa thực tiễn cao trong hoạt động marketing hiện đại.

1.4. Mục tiêu của đề tài

Đề tài hướng đến việc ứng dụng mô hình RFM kết hợp với các thuật toán phân cụm để phân loại khách hàng dựa trên hành vi mua sắm, từ đó hỗ trợ doanh nghiệp trong việc xây dựng chiến lược tiếp thị hiệu quả hơn. Cụ thể, đề tài tập trung vào các mục tiêu sau:

- **Xây dựng hệ thống đánh giá hành vi khách hàng** thông qua ba chỉ số RFM: Recency (gần đây), Frequency (tần suất), và Monetary (giá trị chi tiêu).
- **Phân cụm khách hàng một cách khoa học** bằng thuật toán **K-Means** dựa trên dữ liệu RFM đã được chuẩn hóa.
- **Phân cụm khách hàng bằng thuật toán DBSCAN**, nhằm khám phá các cụm có hình dạng bất quy tắc và phát hiện các điểm dữ liệu bất thường (outlier) mà K-Means có thể bỏ sót.
- **So sánh kết quả phân cụm giữa K-Means và DBSCAN**, phân tích điểm mạnh – điểm yếu của từng phương pháp trong ngữ cảnh dữ liệu RFM.
- **Phân tích đặc điểm từng nhóm khách hàng**, từ đó rút ra nhận định về giá trị và hành vi của mỗi nhóm.
- **Đề xuất định hướng ứng dụng kết quả phân cụm** vào các hoạt động tiếp thị, chăm sóc và giữ chân khách hàng, như thiết kế chương trình ưu đãi cá nhân hóa hoặc ưu tiên tập trung vào nhóm khách hàng có giá trị cao.

Thông qua việc kết hợp cả hai phương pháp, đề tài không chỉ khai thác toàn diện tiềm năng từ dữ liệu hành vi khách hàng mà còn mở rộng góc nhìn so sánh trong việc ứng dụng các thuật toán học máy trong thực tiễn kinh doanh.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Thuật toán phân cụm K – Means

2.1.1. Khái niệm về K – Means

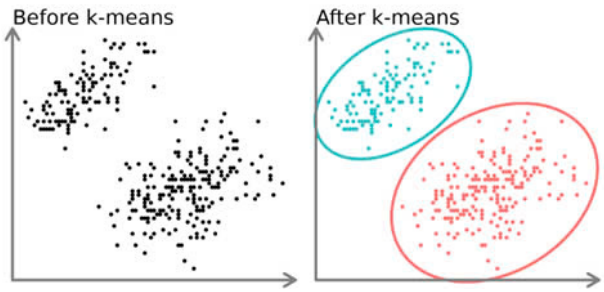
Thuật toán K-means là một kỹ thuật học không giám sát nổi bật, được ứng dụng rộng rãi trong phân tích dữ liệu nhằm chia tập dữ liệu chưa gán nhãn thành các nhóm (cụm) sao cho các điểm dữ liệu trong cùng một cụm có mức độ tương đồng cao, trong khi sự khác biệt giữa các cụm là tối đa. Nguyên lý hoạt động của K-means dựa trên việc tối thiểu hóa tổng bình phương khoảng cách giữa các điểm dữ liệu và tâm cụm mà chúng được gán vào. Mỗi điểm sẽ được phân vào cụm có trung tâm gần nhất, và các tâm cụm được cập nhật liên tục cho đến khi đạt được trạng thái hội tụ.[1]

Mục tiêu chính của K-Means là giảm thiểu tổng bình phương khoảng cách giữa các điểm dữ liệu và trung tâm cụm của chúng, được gọi là Within-Cluster Sum of Squares (WCSS):

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

Trong đó:

- K: Là số lượng cụm



Hình 1: Minh họa về K - Means

- C_i: Là tập hợp các điểm thuộc cụm thứ i
- μ_i: Là trung tâm (tâm cụm) của cụm thứ i
- $\|x - \mu_i\|^2$: Là bình phương khoảng cách Euclidean giữa điểm x và tâm cụm μ_i

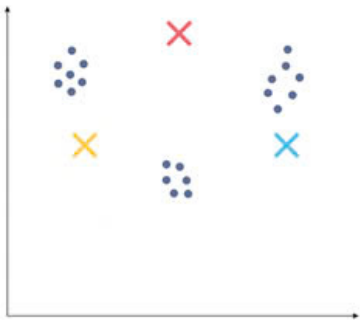
2.1.2. Cách hoạt động của K-Means

Bước 1: Chọn số lượng cụm

- Xác định số lượng cụm K mà ta sẽ nhóm dữ liệu. Ví dụ chọn $K = 3$.

Bước 2: Khởi tạo tâm cụm ban đầu

- Do vị trí chính xác của các tâm cụm chưa được biết, nên ở giai đoạn khởi tạo, thuật toán sẽ chọn ngẫu nhiên K điểm từ tập dữ liệu và coi đó là các tâm cụm ban đầu.



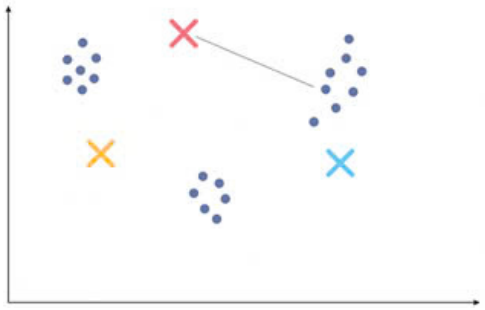
Hình 2: Ví dụ minh họa về khởi tạo tâm cụm

Bước 3: Gán điểm dữ liệu cho cụm gần nhất - Sau khi đã có tâm cụm ban đầu, mỗi điểm dữ liệu sẽ được gán vào cụm có tâm gần nhất. Khoảng cách giữa điểm dữ liệu và tâm cụm thường được đo bằng khoảng cách Euclidean. Điểm dữ liệu sẽ thuộc về cụm có tâm mà nó có khoảng cách Euclidean ngắn nhất.

- Đo khoảng cách từ các điểm tới tâm cụm bằng phép đo khoảng cách Euclidean.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

- Sau đó chọn cụm cho dữ liệu có khoảng cách giữa các điểm dữ liệu và tâm nhỏ nhất.



Hình 3: Ví dụ về gán điểm dữ liệu cho cụm cần nhất

Hình 4: Sau khi đo khoảng cách tâm tới điểm dữ liệu

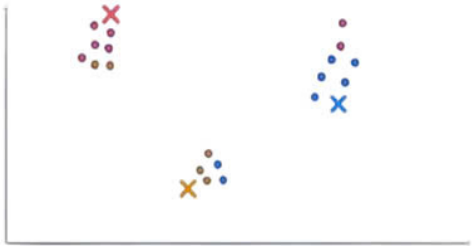
Bước 4: Khởi tạo lại tâm cụm

- Khởi tạo lại trọng tâm bằng cách tính toán giá trị trung bình của tất cả các điểm dữ liệu của cụm đó.

$$C_i = \frac{1}{|N_i|} \sum x_i$$

Trong đó:

- C_i: là tâm cụm thứ i (centroid của cụm i)
- N_i: là tập các điểm thuộc cụm i
- $|N_i|$: là số lượng điểm trong cụm i (kích thước cụm)
- $\sum x_i$: là tổng các vector dữ liệu x_i trong cụm i



Hình 5: Sau khi khởi tạo lại tâm cụm

Bước 5: Lặp lại

- Tiến hành lặp lại bước 3 và bước 4 cho đến khi có được trọng tâm tối ưu và việc chỉ định các điểm dữ liệu cho các cụm chính xác không còn thay đổi.[2]

Hình 6: Hoàn thành quá trình phân cụm

2.1.3. Lựa chọn K cho K – Means

Một trong những thách thức lớn nhất khi sử dụng K-Means là xác định số lượng cụm K phù hợp. Không có một giá trị K tối ưu duy nhất cho mọi bộ dữ liệu, và việc chọn K phụ thuộc vào đặc điểm của dữ liệu và mục tiêu phân tích. Dưới đây là những phương pháp chọn K cho tối ưu:

Phương pháp Elbow (Khuỷu tay): Tìm số lượng cụm sao cho tổng phương sai trong cụm (WSS – within-cluster sum of squares) là nhỏ nhất, nhưng không giảm quá nhiều khi tăng thêm cụm. Công thức WCSS:

$$WCSS = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

- **Cách thực hiện:**
  1. Thực hiện phân cụm với các giá trị K khác nhau (ví dụ từ 1 đến 10).
  2. Tính tổng WSS cho mỗi K.
  3. Vẽ biểu đồ WSS theo số cụm K.
  4. Điểm "gấp khúc" (elbow) trên đồ thị thường là nơi phù hợp để chọn K.

Phương pháp Silhouette: Đánh giá mức độ phù hợp của từng điểm dữ liệu với cụm của nó. Giá trị Silhouette càng gần 1 cho thấy điểm đó nằm "đúng" trong cụm hơn.

- **Cách thực hiện:**
  1. Phân cụm dữ liệu với các giá trị khác nhau.
  2. Tính độ rộng silhouette trung bình cho mỗi giá trị K.
  3. Vẽ đồ thị silhouette theo K.
  4. Chọn K tại vị trí có giá trị silhouette trung bình lớn nhất.

Công thức Silhouette:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i),\ b(i))}$$

Trong đó:

$a(i)$: Khoảng cách trung bình từ điểm *i* đến tất cả các điểm khác trong **cùng cụm**.

$b(i)$: Khoảng cách trung bình nhỏ nhất từ điểm *i* đến các điểm trong **cụm khác gần nhất**.

- **Kết quả Silhouette trả về nằm trong khoảng** $[-1, 1]$:
  - Gần **1**: Điểm dữ liệu được phân cụm **rất tốt**.
  - Gần **0**: Điểm dữ liệu nằm ở **ranh giới giữa hai cụm**.

  - Gần **-1**: Điểm dữ liệu có thể bị **phân cụm sai hoàn toàn**.

- **Đánh giá chất lượng phân cụm dựa trên Silhouette trung bình**:
  - **0.71 – 1.00**: Cấu trúc phân cụm **mạnh**.
  - **0.51 – 0.70**: Cấu trúc phân cụm **hợp lý**.
  - **0.26 – 0.50**: Cấu trúc phân cụm **yếu**, nên **xem xét lại**.
  - **≤ 0.25**: **Không phát hiện được** cấu trúc phân cụm rõ ràng.

Phương pháp Gap Statistic: Gap Statistic là một kỹ thuật thống kê được đề xuất bởi Tibshirani và cộng sự (2001), dùng để xác định số cụm tối ưu trong phân cụm dữ liệu. Phương pháp này đánh giá sự khác biệt giữa độ phân tán của dữ liệu thực và dữ liệu ngẫu nhiên sinh ra từ phân phối đồng nhất.

- **Cách thực hiện:**
  1. **Tạo dữ liệu tham chiếu** bằng cách lấy mẫu ngẫu nhiên từ không gian của dữ liệu gốc.
  2. **Tính và so sánh mức độ phân tán** giữa dữ liệu thực và dữ liệu tham chiếu.
  3. **Xác định số cụm tối ưu** dựa trên mức độ chênh lệch lớn nhất giữa hai mức phân tán.

- **Công thức Gap Statistic:**

$$Gap(k) = E_n^* \left[\log(W_k)\right] - \log(W_k)$$

Trong đó:

- W_k: Tổng phương sai trong cụm khi phân cụm dữ liệu thực với k cụm.
- $E_n^* \left[\log(W_k)\right]$: Giá trị kỳ vọng của $\log(W_k)$ từ dữ liệu ngẫu nhiên sinh ra n lần.
- n: Số lần sinh dữ liệu ngẫu nhiên để ước lượng kỳ vọng.
- **Cách xác định số cụm K tối ưu:**
  - Tính Gap Statistic cho nhiều giá trị k.
  - Chọn giá trị nhỏ nhất của k sao cho:

$$Gap(k) \geq Gap(k+1) - s_{k+1}$$

Trong đó s_{k+1} là sai số chuẩn của $Gap(k+1)$.

- **Phân tích và đánh giá:**
  - **Phạm vi**: Không có giá trị cố định, nhưng Gap thường là **số dương**.
  - **Giá trị Gap cao**: Cho thấy dữ liệu có **cấu trúc phân cụm tốt**.
  - **Chất lượng phân cụm**: K tối ưu là điểm mà Gap bắt đầu **ổn định hoặc giảm**.[3]

2.1.4. Ưu điểm và nhược điểm của K-Means

Ưu điểm:

- **Đơn giản và dễ triển khai**: K-Means là một trong những thuật toán phân cụm phổ biến và dễ hiểu nhất.
- **Tốc độ xử lý nhanh**: Nhờ cấu trúc tính toán hiệu quả, K-Means có thể xử lý nhanh với dữ liệu lớn và số cụm nhỏ.
- **Khả năng mở rộng tốt**: Có thể áp dụng cho dữ liệu lớn hoặc phân cụm theo batch.
- **Linh hoạt**: Có thể áp dụng cho nhiều loại dữ liệu sau khi chuẩn hóa và chuyển đổi thích hợp.

Nhược điểm:

- **Phải xác định trước số cụm** K: Đây là yêu cầu bắt buộc và gây khó khăn khi chưa hiểu rõ dữ liệu.
- **Phụ thuộc vào khởi tạo ban đầu**: Kết quả phân cụm bị ảnh hưởng bởi cách chọn tâm cụm ban đầu.
- **Giả định cụm hình cầu**: K-Means hoạt động tốt khi các cụm có hình dạng đối xứng, đồng đều và không chồng lấn.
- **Nhạy cảm với outliers**: Các điểm ngoại lai có thể làm lệch tâm cụm và ảnh hưởng đáng kể đến kết quả phân cụm.[4]

2.1.5. Ứng dụng của K-Means trong thực tế

K-Means là một trong những thuật toán phân cụm phổ biến nhất hiện nay, đặc biệt hiệu quả khi làm việc với **dữ liệu số và liên tục**. Thuật toán được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm:

- **Phân khúc khách hàng**: Giúp doanh nghiệp chia nhóm khách hàng theo hành vi mua sắm hoặc đặc điểm tiêu dùng. Nhờ đó, doanh nghiệp có thể **cá nhân hóa chiến lược marketing** và cải thiện dịch vụ chăm sóc khách hàng.
- **Phát hiện gian lận**: K-Means hỗ trợ **phát hiện điểm dữ liệu bất thường**, ví dụ như các giao dịch đáng ngờ trong tài chính hoặc truy cập trái phép trong bảo mật.
- **Phân loại tài liệu**: Trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), K-Means được sử dụng để **nhóm tài liệu theo chủ đề tương tự**, dựa trên biểu diễn vector và đặc trưng từ vựng.
- **Phân tích dữ liệu không gian và hình ảnh**: K-Means có thể **chia ảnh thành các vùng** dựa trên màu sắc hoặc độ sáng. Ứng dụng này rất hữu ích trong **thị giác máy tính** và **y tế**, chẳng hạn như **chẩn đoán hình ảnh y khoa**.[5]

2.2. Thuật toán phân cụm Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

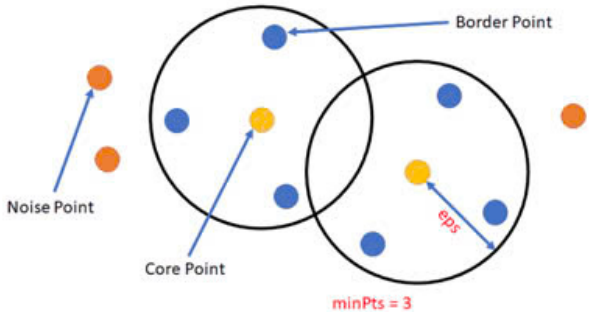
2.2.1. Khái niệm về DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) là một thuật toán phân cụm dựa trên mật độ, được đề xuất bởi Ester, Kriegel, Sander và Xu vào năm 1996.

Thuật toán này phân nhóm các điểm dữ liệu nằm gần nhau trong vùng có mật độ cao, đồng thời nhận diện và loại bỏ các điểm nằm riêng lẻ ở vùng mật độ thấp như là ngoại lệ (nhiễu).

Khác với K-means, DBSCAN không yêu cầu xác định trước số cụm, và đặc biệt hiệu quả trong việc phát hiện các cụm có hình dạng phức tạp và chứa nhiễu. Đây là một trong những thuật toán phân cụm phổ biến và được sử dụng rộng rãi trong học máy không giám sát.[6]

Các khái niệm quan trọng trong DBSCAN:



Hình 7: Thuật toán DBSCAN

- **Epsilon (ε)** – Bán kính vùng lân cận của một điểm dữ liệu.

$$N_\varepsilon(p) = \{q \in D \mid \text{dist}(p, q) \leq \varepsilon\}$$

- Trong đó:
  - D: Là toàn bộ tập dữ liệu huấn luyện.
  - $\text{dist}(p, q)$: Là khoảng cách giữa điểm p và q, thường được tính bằng khoảng cách Euclidean.
- **MinPts** – Số lượng điểm tối thiểu cần có trong vùng lân cận ε để được xem là vùng có mật độ cao.
- **Điểm lõi (Core Point)**: Có ít nhất *MinPts* điểm trong vùng lân cận ε (bao gồm cả chính nó).
- **Điểm biên (Border Point)**: Không đủ *MinPts*, nhưng thuộc vùng lân cận của một điểm lõi.
- **Điểm nhiễu (Noise / Outlier)**: Không phải điểm lõi, cũng không thuộc vùng lân cận của bất kỳ điểm lõi nào.[7]

2.2.2. Cách xác định tham số cho DBSCAN

Một trong những thách thức chính khi sử dụng DBSCAN là lựa chọn hai tham số cốt lõi là: ε (epsilon) và MinPts.

Đối với K – Means thì thuật toán cần xác định trước số cụm K còn đối với DBSCAN cần yêu cầu xác định mức độ dày đặt cần thiết để phân thành cụm thông qua hai yếu tối trên:

- **Epsilon (ε)**:
  - Là tham số xác định bán kính vùng lân cận của mỗi điểm ảnh hưởng trực tiếp đến kết quả phân cụm.
  - Cách phổ biến để xác định ε là sử dụng **đồ thị k-distance**:
    - Với mỗi điểm trong tập dữ liệu, tính khoảng cách đến điểm gần nhất thứ k. Trong đó:

$$k = \text{MinPts} - 1$$

    - Sắp xếp các khoảng cách theo thứ tự tăng dần.
    - Vẽ đồ thị khoảng cách (trục hoành: số điểm, trục tung: khoảng cách).
    - Quan sát biểu đồ để tìm "**điểm gãy**" (elbow point) — nơi độ dốc đường cong thay đổi rõ rệt.
    - **Chọn ε tương ứng với điểm gãy của đồ thị**.
- **MinPts**:
  - Dùng để xác định số điểm tối thiểu cần có trong vùng lân cận ε để một điểm được xem là điểm lõi.
  - Gợi ý chọn:

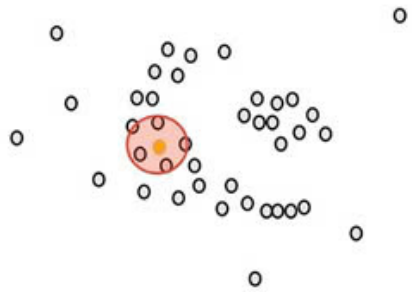
$$\text{MinPts} \geq D + 1$$

    - Đối với dữ liệu 2 chiều: MinPts = 4.
    - Dữ liệu có nhiều chiều hoặc nhiễu: MinPts = 5–10.
  - MinPts càng lớn thì thuật toán càng bền vững nhưng có thể bỏ qua những cụm nhỏ hoặc thừa.

2.2.3. Nguyên lý hoạt động của DBSCAN

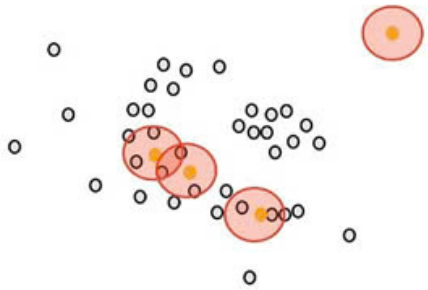
Thuật toán DBSCAN thực hiện phân cụm theo mật độ qua các bước sau:

- **Bước 1: Chọn một điểm bất kỳ**
  - Chưa được gán nhãn trên tập dữ liệu.
  - Vẽ một vòng tròn xung quanh đó với bán kính ε (epsilon), đây là vùng lân cận.



Hình 8: Chọn một cụm xung quanh điểm dữ liệu

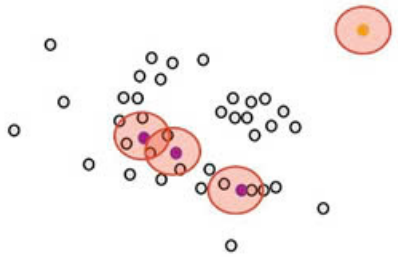
- **Bước 2: Kiểm tra số lượng trong vùng ε**
  - Đếm số điểm nằm trong ε
  - Nếu số điểm lớn hơn hoặc bằng *MinPts* thì được coi là điểm lõi.
  - Nếu ít hơn, điểm đó có thể là biên hoặc nhiễu tùy thuộc vào những bước tiếp theo.



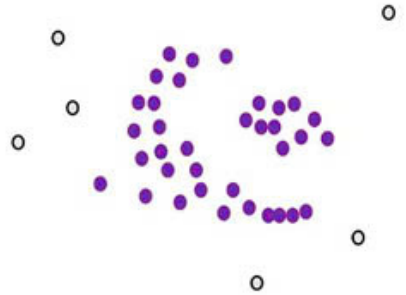
Hình 9: Chọn một cụm xung quanh điểm dữ liệu

- **Bước 3: Xác định điểm lõi và mở rộng cụm**
  - Với điểm lõi vừa tìm được, bắt đầu tạo ra cụm mới.
  - Thêm tất cả các điểm trong vùng ε của điểm lõi vào cụm đó.
  - Nếu trong số các điểm mới thêm vào có điểm nào cũng là điểm lõi thì tiếp tục mở rộng cụm từ điểm đó.

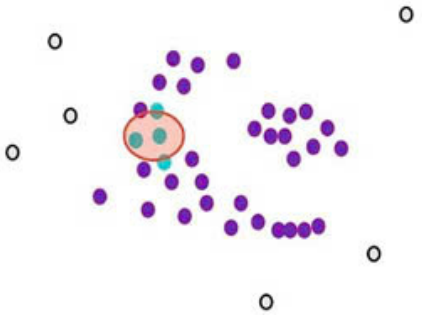
- **Bước 4: Xử lý điểm biên và nhiễu**
  - Những điểm không đủ điều kiện làm điểm lõi nhưng nằm trong vùng lân cận của một điểm lõi sẽ được gán vào cụm đó và được gọi là điểm biên.
  - Những điểm không nằm trong các vùng lân cận của bất kỳ điểm lõi nào được coi là điểm nhiễu.



Hình 10: Xác định bốn điểm lõi màu tím và một điểm ngoại lệ màu vàng



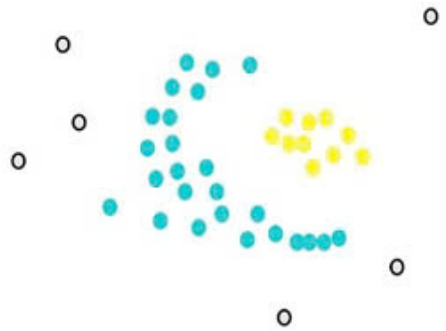
Hình 11: Tất cả các điểm dữ liệu lõi có màu tímn



Hình 12: Điểm lõi được chọn ngẫu nhiên màu xanh ngọc

- **Bước 5: Tiếp tục với các điểm chưa xét**
  - Chọn một điểm lõi khác chưa được gán cụm, lặp lại quá trình mở rộng cụm như các bước trên.
  - Các cụm mới sẽ được hoàn thành từ các điểm lõi còn lại.

- **Bước 6: Kết thúc phân cụm**
  - Sau khi xét hết các điểm:
    * Tất cả đều được gán cụm.
    * Các cụm được tô màu khác nhau.
    * Điểm nhiễu không thay đổi.



Hình 13: Kết thúc phân cụm

2.2.4. Ưu điểm và nhược điểm của DBSCAN

Ưu điểm: DBSCAN có khả năng phát hiện cụm với hình dạng bất kỳ, không bị giới hạn như K-Means. Thuật toán xử lý tốt dữ liệu nhiễu bằng cách tự động loại bỏ các điểm ngoài cụm. Ngoài ra, DBSCAN không yêu cầu xác định trước số lượng cụm, phù hợp với dữ liệu chưa rõ cấu trúc.

Nhược điểm: Kết quả phân cụm phụ thuộc nhiều vào lựa chọn tham số epsilon và min_samples. DBSCAN gặp khó khăn khi dữ liệu có mật độ cụm không đồng đều và hoạt động kém hiệu quả với dữ liệu nhiều chiều do khoảng cách giữa các điểm trở nên không rõ ràng.

2.2.5. Ứng dụng trong thực tế của DBSCAN

DBSCAN được sử dụng hiệu quả trong các bài toán phân cụm phức tạp, đặc biệt khi dữ liệu không có cấu trúc rõ ràng hoặc chứa nhiều nhiễu:

Phân cụm không gian: DBSCAN phù hợp với dữ liệu địa lý như xác định khu vực có mật độ tội phạm cao, vùng có tỷ lệ bệnh bất thường hoặc điểm nóng tai nạn giao thông trong đô thị.

Phát hiện bất thường: Mô hình có thể cô lập các điểm lẻ không thuộc cụm, hỗ trợ phát hiện giao dịch gian lận, lỗi cảm biến trong công nghiệp hoặc dữ liệu bất thường trong hệ thống.

Phân khúc khách hàng: DBSCAN cho phép nhóm khách hàng theo hành vi tiêu dùng mà không cần xác định trước số nhóm, hữu ích cho phân tích mua sắm và cá nhân hóa chiến dịch tiếp thị.

Xử lý dữ liệu hình ảnh và văn bản: Có thể ứng dụng vào phân loại ảnh y khoa, thị giác máy tính và nhóm tài liệu theo chủ đề hoặc biểu hiện sinh học, hỗ trợ phân tích bệnh lý hoặc nghiên cứu chuyên sâu.[8]

2.3. RFM Analysis

2.3.1. Khái niệm về RFM

RFM (Recency – Frequency – Monetary) là mô hình phân tích và phân khúc khách hàng dựa trên hành vi tiêu dùng, sử dụng dữ liệu giao dịch lịch sử để đánh giá và phân loại khách hàng theo ba tiêu chí chính:

- **Recency (R):** Khoảng thời gian kể từ lần mua hàng gần nhất của khách hàng.
- **Frequency (F):** Số lần khách hàng đã mua hàng trong một khoảng thời gian.
- **Monetary (M):** Tổng số tiền khách hàng đã chi tiêu.

Phân tích RFM giúp doanh nghiệp hiểu rõ hành vi mua sắm của khách hàng, từ đó đưa ra các chiến lược tiếp thị và chăm sóc khách hàng phù hợp.

2.3.2. Ý nghĩa của từng yếu tố RFM

Thành phần	Ý nghĩa	Ví dụ
Recency	Khách hàng mua hàng gần đây thường có khả năng quay lại cao hơn.	Một khách hàng vừa mua cách đây 3 ngày có giá trị R cao hơn người mua cách đây 3 tháng.
Frequency	Khách hàng mua hàng nhiều lần là dấu hiệu của sự trung thành.	Một khách hàng mua 10 lần trong 3 tháng có F cao hơn người mua 1 lần.
Monetary	Khách hàng chi tiêu nhiều thường có giá trị kinh tế cao hơn.	Một người chi 10 triệu có M cao hơn người chi 500 nghìn.

2.3.3. Tại sao nên sử dụng phân tích RFM?

Phân tích RFM mang lại nhiều lợi ích cho doanh nghiệp:

- **Hiểu rõ hành vi khách hàng:** Dựa trên dữ liệu thực tế, không phỏng đoán .
- **Tối ưu hóa chiến lược marketing:** Gửi đúng thông điệp cho đúng người vào đúng thời điểm.
- **Phân bổ nguồn lực hiệu quả:** Ưu tiên chăm sóc nhóm khách hàng có giá trị cao.

2.3.4. Lợi ích của mô hình RFM

Lợi ích	Giải thích
Đơn giản, dễ áp dụng	Chỉ cần dữ liệu mua hàng cơ bản, không cần mô hình phức tạp
Dựa trên hành vi thực tế	Chính xác hơn so với chỉ phân tích nhân khẩu học
Dễ phân khúc	Có thể chia khách hàng thành các nhóm nhỏ để chăm sóc riêng
Dễ mở rộng	Có thể kết hợp thêm các yếu tố khác như CLV, churn rate

2.3.5. Quy trình phân tích RFM

- **Bước 1: Thu thập dữ liệu**
  - Cần các trường thông tin: Customer ID, ngày giao dịch, giá trị giao dịch.
- **Bước 2: Tính toán chỉ số RFM**
  - **Recency:** Ngày hiện tại – Ngày mua hàng gần nhất
  - **Frequency:** Số lượng giao dịch
  - **Monetary:** Tổng tiền chi tiêu
- **Bước 3: Gán điểm RFM**
  - Chia mỗi chỉ số thành 5 mức (1 = thấp nhất, 5 = cao nhất), gán điểm cho từng khách hàng.
- **Bước 4: Phân khúc khách hàng**
  - Gán mã RFM (ví dụ: R=5, F=4, M=2 → RFM = 542). Từ đó chia thành các nhóm hành vi.
- **Bước 5: Ứng dụng chiến lược phù hợp**

Phân khúc	Đặc điểm chính	Chiến lược đề xuất
VIP	R thấp, F cao, M cao	Chăm sóc đặc biệt, ưu đãi cao
Khách trung thành	R thấp, F cao, M trung bình	Gửi thông báo thường xuyên, tặng thưởng
Mới đến	R rất thấp, F thấp, M thấp	Hướng dẫn, giới thiệu sản phẩm
Sắp rời bỏ	R cao, F cao, M cao	Gửi email khuyến mãi giữ chân
Ngủ quên	R rất cao, F thấp, M thấp	Gửi phiếu giảm giá, ưu đãi đánh thức nhu cầu

2.3.6. Kết luận

RFM là một công cụ mạnh mẽ giúp doanh nghiệp cá nhân hóa marketing, tăng hiệu quả bán hàng và duy trì mối quan hệ với khách hàng. Với chi phí triển khai thấp và khả năng ứng dụng rộng rãi, phân tích RFM là một trong những phương pháp không thể thiếu trong kho vũ khí marketing hiện đại.[9]

CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

3.1. Giới thiệu về bộ dữ liệu

Bộ dữ liệu **Online Retail** được thu thập từ một doanh nghiệp bán lẻ trực tuyến tại Vương quốc Anh trong khoảng thời gian từ tháng 12/2010 đến tháng 12/2011. Đây là một tập dữ liệu giao dịch thực tế gồm hơn **500.000 dòng**, phản ánh các hóa đơn mua hàng từ khách hàng trên toàn thế giới. Bộ dữ liệu được sử dụng rộng rãi trong các bài toán khai phá dữ liệu, phân tích hành vi khách hàng, và xây dựng mô hình phân khúc.

Mục tiêu phân tích:

- Phân tích hành vi mua hàng của từng khách hàng dựa trên thời gian mua hàng gần nhất, số lần mua và giá trị mua.
- Xây dựng mô hình RFM (Recency - Frequency - Monetary) để phân khúc khách hàng.
- Áp dụng thuật toán phân cụm K-Means và DBSCAN để khám phá các nhóm khách hàng tiềm năng.
- Gợi ý chiến lược marketing phù hợp với từng nhóm khách hàng nhằm tối ưu hóa doanh thu.

Tổng quan về dữ liệu:

Tên biến	Giải thích	Kiểu dữ liệu
InvoiceNo	Mã đơn hàng (hóa đơn)	Ký tự
StockCode	Mã sản phẩm	Ký tự
Description	Mô tả sản phẩm	Ký tự
Quantity	Số lượng sản phẩm	Số nguyên
InvoiceDate	Ngày giao dịch	Ngày giờ
UnitPrice	Giá của một đơn vị sản phẩm	Số thực
CustomerID	Mã khách hàng	Số nguyên
Country	Quốc gia của khách hàng	Ký tự

Đặc điểm của bài toán:

- Là bài toán phân tích hành vi khách hàng không giám sát.
- Kết hợp nhiều biến định tính và định lượng.
- Yêu cầu tiền xử lý dữ liệu kỹ lưỡng.

Đọc và khám phá dữ liệu

```
# Đọc dữ liệu
df <- read_excel("Online Retail.xlsx")
head(df, width = Inf)
```

```
## # A tibble: 6 x 8
##   InvoiceNo StockCode Description         Quantity InvoiceDate         UnitPrice
##   <chr>     <chr>     <chr>                  <dbl> <dttm>                  <dbl>
## 1 536365    85123A    WHITE HANGING HEAR~        6 2010-12-01 08:26:00      2.55
## 2 536365    71053     WHITE METAL LANTERN        6 2010-12-01 08:26:00      3.39
## 3 536365    84406B    CREAM CUPID HEARTS~        8 2010-12-01 08:26:00      2.75
## 4 536365    84029G    KNITTED UNION FLAG~        6 2010-12-01 08:26:00      3.39
## 5 536365    84029E    RED WOOLLY HOTTIE ~        6 2010-12-01 08:26:00      3.39
## 6 536365    22752     SET 7 BABUSHKA NES~        2 2010-12-01 08:26:00      7.65
## # i 2 more variables: CustomerID <dbl>, Country <chr>
```

Nhận xét:

- Bộ dữ liệu có cấu trúc hợp lý với 8 cột cơ bản, mỗi dòng tương ứng với một dòng sản phẩm trong hóa đơn.
- Đây là dữ liệu phản ánh hành vi mua hàng trực tiếp từ khách hàng, rất phù hợp để áp dụng mô hình RFM.
- Các biến `Quantity`, `UnitPrice`, và `CustomerID` là cơ sở để tính toán các chỉ số giá trị (Monetary), tần suất (Frequency) và gần đây (Recency).
- Việc có cả biến mô tả sản phẩm và địa lý (`Country`) cho phép mở rộng phân tích hành vi theo khu vực và theo loại hàng hóa.

```
#Kiểm tra kiểu dữ liệu
str(df)
```

```
## tibble [541,909 x 8] (S3: tbl_df/tbl/data.frame)
##  $ InvoiceNo  : chr [1:541909] "536365" "536365" "536365" "536365" ...
##  $ StockCode  : chr [1:541909] "85123A" "71053" "84406B" "84029G" ...
##  $ Description: chr [1:541909] "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM C
##  $ Quantity   : num [1:541909] 6 6 8 6 6 2 6 6 6 32 ...
```

```
##  $ InvoiceDate: POSIXct[1:541909], format: "2010-12-01 08:26:00" "2010-12-01 08:26:00" ...
##  $ UnitPrice  : num [1:541909] 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
##  $ CustomerID : num [1:541909] 17850 17850 17850 17850 17850 ...
##  $ Country    : chr [1:541909] "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom"
```

- Dữ liệu có kiểu phù hợp, đặc biệt là `InvoiceDate` cần được định dạng lại đúng format, rất quan trọng khi tính Recency.
- Một số trường như `InvoiceNo`, `StockCode` là chuỗi, cần chuẩn hóa vì có thể chứa ký hiệu đặc biệt hoặc mã huỷ đơn.
- `Quantity`, `UnitPrice` có dạng số, phù hợp cho các phép tính toán giá trị chi tiêu nhưng cần xem xét dữ liệu âm hoặc bằng 0.

```
# thống kê mô tả dữ liệu
describe(df)
```

```
##              vars      n     mean      sd   median  trimmed     mad       min
## InvoiceNo*      1 541909 11649.52 6730.30 11968.00 11699.43 8774.03      1.00
## StockCode*      2 541909  1651.78  932.96  1564.00  1599.35  892.53      1.00
## Description*    3 540455  2180.39 1165.08  2137.00  2200.40 1423.30      1.00
## Quantity        4 541909     9.55  218.08     3.00     5.20    2.97 -80995.00
## InvoiceDate     5 541909      NaN      NA       NA      NaN      NA       Inf
## UnitPrice       6 541909     4.61   96.76     2.08     2.60    1.82 -11062.06
## CustomerID      7 406829 15287.69 1713.60 15152.00 15288.25 2195.73  12346.00
## Country*        8 541909    34.35    5.96    36.00    36.00    0.00      1.00
##                 max     range   skew  kurtosis   se
## InvoiceNo*    25900  25899.00  -0.07     -1.21 9.14
## StockCode*     4070   4069.00   0.45     -0.40 1.27
## Description*   4211   4210.00  -0.12     -1.07 1.58
## Quantity      80995 161990.00  -0.26 119767.61 0.30
## InvoiceDate    -Inf      -Inf     NA        NA   NA
## UnitPrice     38970  50032.06 186.51  59004.96 0.13
## CustomerID    18287   5941.00   0.03     -1.18 2.69
## Country*         38     37.00  -3.62     11.97 0.01
```

- Phát hiện ra các giá trị âm và bằng 0 ở `Quantity` và `UnitPrice` có thể đại diện cho việc trả hàng hoặc lỗi nhập liệu.
- Sự tồn tại của các giá trị NA trong `CustomerID` là vấn đề lớn vì phân tích RFM yêu cầu định danh khách hàng rõ ràng.
- Việc kiểm tra và loại bỏ hoặc xử lý phù hợp các dữ liệu này là bước không thể thiếu trước khi đi vào phân cụm.

```
# Thống kê data object
summary(df[, sapply(df, is.character)])
```

```
##    InvoiceNo          StockCode         Description          Country
##  Length:541909      Length:541909      Length:541909      Length:541909
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

- `InvoiceNo` và `StockCode` có số lượng giá trị duy nhất lớn, xác nhận rằng mỗi hóa đơn và mã sản phẩm gần như là riêng biệt.
- Có thể tồn tại `InvoiceNo` bắt đầu bằng chữ "C" – đại diện cho đơn huỷ, cần loại bỏ để không làm sai lệch phân tích.

- Country là trường quan trọng để đánh giá hành vi theo quốc gia có thể mở rộng phân khúc theo địa lý nếu cần.

```
#kiểm tra dữ liệu thiếu
colSums(is.na(df))
```

```
##   InvoiceNo   StockCode Description    Quantity InvoiceDate   UnitPrice
##           0           0        1454           0           0           0
##  CustomerID     Country
##      135080           0
```

- CustomerID có số lượng giá trị thiếu lớn (trên 100.000 dòng), ảnh hưởng trực tiếp đến việc tính toán chỉ số RFM.
- Cần xác định xem các dòng thiếu ID có phải là giao dịch thật hay chỉ là ghi nhận nội bộ, từ đó quyết định loại bỏ hay gán ID thay thế.
- Nhìn chung, việc làm sạch dữ liệu là bước bắt buộc để đảm bảo độ tin cậy cho các phân tích và mô hình tiếp theo.

```
# Chuyển 'InvoiceNo' về kiểu chuỗi
df$InvoiceNo <- as.character(df$InvoiceNo)
#Lọc ra những dòng KHÔNG có InvoiceNo đúng 6 chữ số
df[!grepl("^\\d{6}$", df$InvoiceNo), ]
```

```
## # A tibble: 9,291 x 8
##    InvoiceNo StockCode Description        Quantity InvoiceDate         UnitPrice
##    <chr>     <chr>     <chr>                 <dbl> <dttm>                  <dbl>
##  1 C536379   D         Discount                 -1 2010-12-01 09:41:00     27.5
##  2 C536383   35004C    SET OF 3 COLOURED~       -1 2010-12-01 09:49:00      4.65
##  3 C536391   22556     PLASTERS IN TIN C~      -12 2010-12-01 10:24:00      1.65
##  4 C536391   21984     PACK OF 12 PINK P~      -24 2010-12-01 10:24:00      0.29
##  5 C536391   21983     PACK OF 12 BLUE P~      -24 2010-12-01 10:24:00      0.29
##  6 C536391   21980     PACK OF 12 RED RE~      -24 2010-12-01 10:24:00      0.29
##  7 C536391   21484     CHICK GREY HOT WA~      -12 2010-12-01 10:24:00      3.45
##  8 C536391   22557     PLASTERS IN TIN V~      -12 2010-12-01 10:24:00      1.65
##  9 C536391   22553     PLASTERS IN TIN S~      -24 2010-12-01 10:24:00      1.65
## 10 C536506   22960     JAM MAKING SET WI~       -6 2010-12-01 12:38:00      4.25
## # i 9,281 more rows
## # i 2 more variables: CustomerID <dbl>, Country <chr>
```

```
#Kiểm tra kí tự khác trong cột InvoiceNo
unique(gsub("[0-9]", "", df$InvoiceNo))
```

```
## [1] ""  "C" "A"
```

```
# kiểm tra col 'InvoiceNo' có giá trị nào bắt đầu bằng kí tự 'A' ko
df[grepl("^A", df$InvoiceNo), ]
```

```
## # A tibble: 3 x 8
##   InvoiceNo StockCode Description     Quantity InvoiceDate         UnitPrice
##   <chr>     <chr>     <chr>              <dbl> <dttm>                  <dbl>
## 1 A563185   B         Adjust bad debt        1 2011-08-12 14:50:00    11062.
## 2 A563186   B         Adjust bad debt        1 2011-08-12 14:51:00   -11062.
## 3 A563187   B         Adjust bad debt        1 2011-08-12 14:52:00   -11062.
## # i 2 more variables: CustomerID <dbl>, Country <chr>
```

- Có nhiều InvoiceNo không tuân thủ định dạng chuẩn 6 chữ số, ví dụ: “C536379”, “A563185”…
- Việc gắn thêm ký tự đầu dòng là có chủ đích từ hệ thống nghiệp vụ:
  - **“C”** thường đại diện cho Cancelled Invoice – tức là hóa đơn bị hủy sau khi xuất.
  - **“A”** là viết tắt của Adjustment / Adjust bad debt – hóa đơn điều chỉnh cho các khoản nợ không thu hồi được.
- Những hóa đơn này không đại diện cho hành vi mua bán thông thường và có thể gây sai lệch khi phân tích Recency, Frequency, Monetary nếu không xử lý.
- Nếu mục tiêu là phân tích khách hàng tích cực, **loại bỏ chúng** trước khi tính RFM.

```
# Đảm bảo 'StockCode' là kiểu chuỗi
df$StockCode <- as.character(df$StockCode)
# Lọc những dòng KHÔNG phải 5 chữ số hoặc 5 chữ số + chữ cái
mask <- !(grepl("^\\d{5}$", df$StockCode) | grepl("^\\d{5}[a-zA-Z]+$", df$StockCode))
# Lấy các StockCode sai định dạng và loại trùng
invalid_stockcodes <- unique(df$StockCode[mask])
# Hiển thị kết quả
print(invalid_stockcodes)
```

```
##  [1] "POST"         "D"            "C2"           "DOT"          "M"
##  [6] "BANK CHARGES" "S"            "AMAZONFEE"    "DCGS0076"     "DCGS0003"
## [11] "gift_0001_40" "DCGS0070"     "m"            "gift_0001_50" "gift_0001_30"
## [16] "gift_0001_20" "DCGS0055"     "DCGS0072"     "DCGS0074"     "DCGS0069"
## [21] "DCGS0057"     "DCGSSBOY"     "DCGSSGIRL"    "gift_0001_10" "PADS"
## [26] "DCGS0004"     "DCGS0073"     "DCGS0071"     "DCGS0068"     "DCGS0067"
## [31] "DCGS0066P"    "B"            "CRUK"
```

```
# Lọc những dòng có chứa chuỗi 'DOT'
df_with_DOT <- df[grepl("DOT", df$StockCode), ]
# Hiển thị kết quả
print(df_with_DOT)
```

```
## # A tibble: 710 x 8
##    InvoiceNo StockCode Description    Quantity InvoiceDate         UnitPrice
##    <chr>     <chr>     <chr>             <dbl> <dttm>                  <dbl>
##  1 536544    DOT       DOTCOM POSTAGE        1 2010-12-01 14:32:00      570.
##  2 536592    DOT       DOTCOM POSTAGE        1 2010-12-01 17:06:00      607.
##  3 536862    DOT       DOTCOM POSTAGE        1 2010-12-03 11:13:00      254.
##  4 536864    DOT       DOTCOM POSTAGE        1 2010-12-03 11:27:00      121.
##  5 536865    DOT       DOTCOM POSTAGE        1 2010-12-03 11:28:00      498.
##  6 536876    DOT       DOTCOM POSTAGE        1 2010-12-03 11:36:00      888.
##  7 537237    DOT       DOTCOM POSTAGE        1 2010-12-06 09:58:00      864.
##  8 537240    DOT       DOTCOM POSTAGE        1 2010-12-06 10:08:00      941.
##  9 537434    DOT       DOTCOM POSTAGE        1 2010-12-06 16:57:00      951.
## 10 537638    DOT       DOTCOM POSTAGE        1 2010-12-07 15:28:00      836.
## # i 700 more rows
## # i 2 more variables: CustomerID <dbl>, Country <chr>
```

Nhận xét:

- Nhiều mã StockCode như “POST”, “DOT”, “BANK CHARGES” hay “gift_000x” không phản ánh sản phẩm thực tế mà là phí vận chuyển, khuyến mãi hoặc mã hệ thống.
- Nhóm có mã “DOT” gồm 710 dòng, mô tả là “DOTCOM POSTAGE” → chỉ là phí vận chuyển, không nên tính vào RFM.

- Việc giữ lại các dòng này sẽ làm sai lệch chỉ số Frequency và Monetary, ảnh hưởng đến phân cụm.
- **Đề xuất:** Nên kiểm tra và loại bỏ các mã bất thường trước khi tính RFM để kết quả phản ánh đúng hành vi mua sắm thực của khách hàng.

Stock Code

- **Include** □ → Giữ lại để phân tích hoặc clustering.
- **Exclude** □ → Loại bỏ hoàn toàn, không dùng để phân tích.
- **Exclude from clustering** □ → Không dùng để phân nhóm nhưng có thể tham khảo.
- **Exclude for now** □ → Tạm loại bỏ, có thể xem xét lại sau.
- **StockCode** is meant to follow the pattern `[0-9]{5}` but seems to have legit values for `[0-9]{5}[a-zA-Z]+`.
  Also contains other values:

Code	Description	Action
POST	Not listed previously, likely a postage-related transaction	Exclude for now
D	Looks valid, represents discount values	Exclude from clustering
C2	Carriage transaction - not sure what this means	Exclude from clustering
DOT	Looks valid, represents postage charges	Exclude from clustering
M or m	Looks valid, represents manual transactions	Exclude from clustering
BANK CHARGES or B	Bank charges	Exclude from clustering
S	Samples sent to customer	Exclude from clustering
AMAZONFEE	Looks like fees for Amazon shipping or something	Exclude for now
DCGS0076	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0003	Variant of DCGS, likely falls under the same category	Exclude from clustering
gift_0001_40	Purchases with gift cards, might be interesting for another analysis, but no customer data	Exclude
DCGS0070	Variant of DCGS, likely falls under the same category	Exclude from clustering
m	Looks valid, represents manual transactions	Exclude from clustering
gift_0001_50	Purchases with gift cards, might be interesting for another analysis, but no customer data	Exclude
gift_0001_30	Purchases with gift cards, might be interesting for another analysis, but no customer data	Exclude
gift_0001_20	Purchases with gift cards, might be interesting for another analysis, but no customer data	Exclude
DCGS0055	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0072	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0074	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0069	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0057	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGSSBOY	Possible category or bundle-related code	Exclude for now
DCGSSGIRL	Possible category or bundle-related code	Exclude for now
gift_0001_10	Purchases with gift cards, might be interesting for another analysis, but no customer data	Exclude
PADS	Looks like a legit stock code for padding	Include
DCGS0004	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0073	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0071	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0068	Variant of DCGS, likely falls under the same category	Exclude from clustering

Code	Description	Action
DCGS0067	Variant of DCGS, likely falls under the same category	Exclude from clustering
DCGS0066P	Variant of DCGS, possibly a special category	Exclude from clustering
B	Bank charges	Exclude from clustering
CRUK	Unknown, possibly a charity-related code	Exclude for now

3.2 Tiền xử lý dữ liệu

```
# Chuyển InvoiceDate thành kiểu Date
df$InvoiceDate <- as.Date(df$InvoiceDate)
str(df)
```

```
## tibble [541,909 x 8] (S3: tbl_df/tbl/data.frame)
##  $ InvoiceNo  : chr [1:541909] "536365" "536365" "536365" "536365" ...
##  $ StockCode  : chr [1:541909] "85123A" "71053" "84406B" "84029G" ...
##  $ Description: chr [1:541909] "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM C
##  $ Quantity   : num [1:541909] 6 6 8 6 6 2 6 6 6 32 ...
##  $ InvoiceDate: Date[1:541909], format: "2010-12-01" "2010-12-01" ...
##  $ UnitPrice  : num [1:541909] 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
##  $ CustomerID : num [1:541909] 17850 17850 17850 17850 17850 ...
##  $ Country    : chr [1:541909] "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom
```

- Đã chuẩn cột `InvoiceDate` hóa thành kiểu Date, rất quan trọng để tính Recency. Hiển thị đúng format yyyy-mm-dd.

```
# 1. Gán lại df gốc
df_cleaned <- df
# Loại bỏ toàn bộ dòng chứa bất kỳ ký tự KHÔNG PHẢI số (0-9)
df_cleaned <- df_cleaned[grepl("^[0-9]+$", df_cleaned$InvoiceNo), ]
```

```
# Kiểm tra còn chữ cái không?
any(grepl("[A-Za-z]", df_cleaned$InvoiceNo))  # Phải trả FALSE
```

```
## [1] FALSE
```

```
# Kiểm tra còn ký tự đặc biệt không?
any(grepl("[^0-9]", df_cleaned$InvoiceNo))    # Phải trả FALSE
```

```
## [1] FALSE
```

- Đoạn xử lý này thể hiện sự thận trọng và hiểu rõ nghiệp vụ trong phân tích dữ liệu. Bằng cách loại bỏ các hóa đơn bị hủy hoặc điều chỉnh, nhóm đã đảm bảo rằng tập dữ liệu đầu vào cho phân tích RFM và phân cụm là sạch, chính xác và phản ánh đúng hành vi mua hàng thực tế. Đây là bước bắt buộc và cực kỳ quan trọng trong toàn bộ quy trình phân tích khách hàng.

```
# Tạo điều kiện giữ lại các StockCode hợp lệ:
mask <- grepl("^\\d{5,6}$", df_cleaned$StockCode) |           # Toàn số, 5-6 chữ số
        grepl("^\\d{5}[A-Za-z]$", df_cleaned$StockCode) |     # 5 số + 1 chữ cái
        df_cleaned$StockCode == "APADSS"                      # Mã đặc biệt

# Áp dụng lọc
df_cleaned <- df_cleaned[mask, ]

df_cleaned
```

```
## # A tibble: 529,825 x 8
##    InvoiceNo StockCode Description     Quantity InvoiceDate UnitPrice CustomerID
##    <chr>     <chr>     <chr>              <dbl> <date>          <dbl>      <dbl>
##  1 536365    85123A    WHITE HANGING ~        6 2010-12-01       2.55      17850
##  2 536365    71053     WHITE METAL LA~        6 2010-12-01       3.39      17850
##  3 536365    84406B    CREAM CUPID HE~        8 2010-12-01       2.75      17850
##  4 536365    84029G    KNITTED UNION ~        6 2010-12-01       3.39      17850
##  5 536365    84029E    RED WOOLLY HOT~        6 2010-12-01       3.39      17850
##  6 536365    22752     SET 7 BABUSHKA~        2 2010-12-01       7.65      17850
##  7 536365    21730     GLASS STAR FRO~        6 2010-12-01       4.25      17850
##  8 536366    22633     HAND WARMER UN~        6 2010-12-01       1.85      17850
##  9 536366    22632     HAND WARMER RE~        6 2010-12-01       1.85      17850
## 10 536367    84879     ASSORTED COLOU~       32 2010-12-01       1.69      13047
## # i 529,815 more rows
## # i 1 more variable: Country <chr>
```

- Lọc và giữ lại các mã StockCode có định dạng hợp lệ, loại bỏ các mã hệ thống không phản ánh sản phẩm thực tế như: phí ship, khuyến mãi, điều chỉnh,…
- Chi tiết:
  - ^\d{5,6}$: Giữ các mã toàn số, gồm 5 hoặc 6 chữ số, ví dụ: “84029” hoặc “84563”
  - ^\d{5}[A-Za-z]$: Giữ mã có 5 chữ số kèm 1 chữ cái, ví dụ: “85123A” – các mã này vẫn có thể đại diện cho sản phẩm thật.
  - == “APADSS”: Một mã sản phẩm đặc biệt có ý nghĩa nghiệp vụ, được giữ lại thủ công

```
sum(is.na(df$CustomerID))  # đếm số dòng bị thiếu
```

```
## [1] 135080
```

- Việc loại bỏ 135,080 dòng thiếu CustomerID là hoàn toàn cần thiết và chính xác, giúp đảm bảo độ tin cậy và hợp lệ cho toàn bộ phân tích tiếp theo.

```
# Xóa các dòng có NA ở cột CustomerID
df_cleaned <- df_cleaned[!is.na(df_cleaned$CustomerID), ]

nrow(df) - nrow(df_cleaned)  # số dòng đã bị xóa
```

```
## [1] 145830
```

```
colSums(is.na(df_cleaned))
```

```
##   InvoiceNo   StockCode Description    Quantity InvoiceDate   UnitPrice
##           0           0           0           0           0           0
##  CustomerID     Country
##           0           0
```

- Sau khi tiến hành xóa 145.830dòng và tập dữ liệu không còn giá trị thiếu là đủ lớn, phù hợp để tiến hành xây dựng mô hình RFM và phân cụm khách hàng

```
# thống kê data sau khi drop na
summary(df_cleaned)
```

```
##   InvoiceNo          StockCode         Description          Quantity
##  Length:396079      Length:396079      Length:396079      Min.   :    1.00
##  Class :character   Class :character   Class :character   1st Qu.:    2.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    6.00
##                                                           Mean   :   13.05
##                                                           3rd Qu.:   12.00
##                                                           Max.   :80995.00
##   InvoiceDate          UnitPrice         CustomerID      Country
##  Min.   :2010-12-01   Min.   :  0.000   Min.   :12346   Length:396079
##  1st Qu.:2011-04-07   1st Qu.:  1.250   1st Qu.:13975   Class :character
##  Median :2011-07-31   Median :  1.950   Median :15159   Mode  :character
##  Mean   :2011-07-10   Mean   :  2.866   Mean   :15302
##  3rd Qu.:2011-10-20   3rd Qu.:  3.750   3rd Qu.:16804
##  Max.   :2011-12-09   Max.   :649.500   Max.   :18287
```

Nhận xét:

- Dữ liệu sau làm sạch bao gồm 396,079 dòng giao dịch hợp lệ, với 8 trường thông tin. Kết quả thống kê mô tả cho thấy:
- Các cột định danh như InvoiceNo, StockCode, Description, Country đều có kiểu dữ liệu phù hợp và không phát hiện lỗi định dạng.
- Thời gian giao dịch trải dài từ 01/12/2010 đến 09/12/2011, đủ để đánh giá hành vi mua hàng theo thời gian (Recency).
- CustomerID đã được loại bỏ đầy đủ các giá trị thiếu (NA) và phân bố hợp lệ từ 12,346 đến 18,287.
- UnitPrice có giá trị bằng 0, không hợp lệ trong phân tích chi tiêu (Monetary) → nên loại bỏ hoặc xử lý tùy ngữ cảnh.

```r
# kiểm tra giá trị bằng 0 của col 'UnitPrice'
sum(df_cleaned$UnitPrice == 0)
```

```
## [1] 33
```

```r
# chỉ lấy giá trị khác 0
df_cleaned <- df_cleaned[df_cleaned$UnitPrice > 0, ]
#Kiểm tra
summary(df_cleaned)
```

```
##   InvoiceNo          StockCode         Description          Quantity
##  Length:396046      Length:396046      Length:396046      Min.   :    1.00
##  Class :character   Class :character   Class :character   1st Qu.:    2.00
##  Mode  :character   Mode  :character   Mode  :character   Median :    6.00
##                                                           Mean   :   13.02
##                                                           3rd Qu.:   12.00
##                                                           Max.   :80995.00
##   InvoiceDate          UnitPrice         CustomerID      Country
##  Min.   :2010-12-01   Min.   :  0.040   Min.   :12346   Length:396046
##  1st Qu.:2011-04-07   1st Qu.:  1.250   1st Qu.:13975   Class :character
##  Median :2011-07-31   Median :  1.950   Median :15159   Mode  :character
##  Mean   :2011-07-10   Mean   :  2.866   Mean   :15302
##  3rd Qu.:2011-10-20   3rd Qu.:  3.750   3rd Qu.:16804
##  Max.   :2011-12-09   Max.   :649.500   Max.   :18287
```

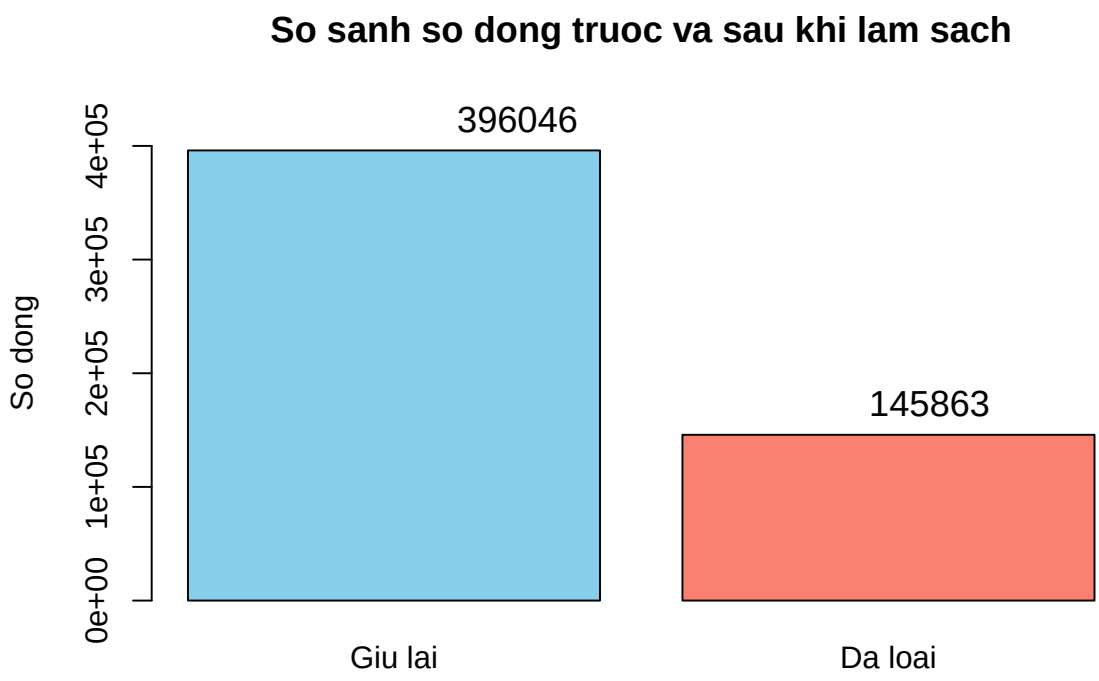
- Kiểm tra và loại bỏ các dòng dữ liệu có giá trị UnitPrice = 0, vì đây không phải là giao dịch mua bán hợp lệ và có thể làm sai lệch chỉ số Monetary trong phân tích RFM.

- Kết quả kiểm tra:
  - Số lượng dòng có UnitPrice = 0: 33 dòng
  - Sau khi lọc, dữ liệu còn lại: 396,046 dòng (giảm từ 396,079)
- Nhận xét:
  - Mặc dù số dòng bị loại khá nhỏ (< 0.01% tổng dữ liệu), nhưng việc lọc bỏ là cần thiết vì:
    * Những dòng này không phản ánh chi tiêu thực tế.
    * Gây sai số khi tính tổng chi tiêu (Monetary) và phân cụm theo giá trị mua hàng.

```
#Tính tỷ lệ phần trăm số dòng còn lại trong df_cleaned so với tổng số dòng ban đầu trong df.
sprintf("%.2f%%", nrow(df_cleaned) / nrow(df) * 100)
```

```
## [1] "73.08%"
```

- Với tỷ lệ giữ lại hơn 73% dữ liệu gốc, tập dữ liệu df_cleaned vẫn đủ lớn và đủ đa dạng để tiếp tục phân tích khách hàng một cách tin cậy, đảm bảo đầu vào chất lượng cho mô hình RFM – KMeans – DBSCAN.



Nhận xét từ biểu đồ so sánh số dòng trước và sau khi làm sạch:

- Tổng số dòng ban đầu: 541,909 dòng
- Số dòng giữ lại sau khi làm sạch (df_cleaned): 396,046 dòng
- Số dòng bị loại bỏ: 145,863 dòng, chiếm khoảng 26.92%

Đánh giá:

- Tỷ lệ dữ liệu bị loại khá cao (~27%), chủ yếu do:

  - Thiếu mã khách hàng (CustomerID)
  - Giao dịch có đơn giá bằng 0 (UnitPrice = 0)
  - Mã hóa đơn (InvoiceNo) hoặc mã sản phẩm (StockCode) không hợp lệ
- việc loại bỏ là cần thiết để đảm bảo:
  - Mỗi dòng giao dịch phản ánh đúng hành vi thực của khách hàng
  - Mô hình RFM không bị sai lệch do dữ liệu nhiễu, lỗi nhập liệu

3.3 Phân tích hành vi khách hàng bằng mô hình RFM

```
# tạo cột tổng chi tiêu làm Monetary (tổng chi tiêu)
df_cleaned$SalesLineTotal <- df_cleaned$Quantity * df_cleaned$UnitPrice
# Hiển thị kết quả
print(df_cleaned, width = Inf)
```

```
## # A tibble: 396,046 x 9
##    InvoiceNo StockCode Description                         Quantity InvoiceDate
##    <chr>     <chr>     <chr>                                  <dbl> <date>
##  1 536365    85123A    WHITE HANGING HEART T-LIGHT HOLDER         6 2010-12-01
##  2 536365    71053     WHITE METAL LANTERN                        6 2010-12-01
##  3 536365    84406B    CREAM CUPID HEARTS COAT HANGER             8 2010-12-01
##  4 536365    84029G    KNITTED UNION FLAG HOT WATER BOTTLE        6 2010-12-01
##  5 536365    84029E    RED WOOLLY HOTTIE WHITE HEART.             6 2010-12-01
##  6 536365    22752     SET 7 BABUSHKA NESTING BOXES               2 2010-12-01
##  7 536365    21730     GLASS STAR FROSTED T-LIGHT HOLDER          6 2010-12-01
##  8 536366    22633     HAND WARMER UNION JACK                     6 2010-12-01
##  9 536366    22632     HAND WARMER RED POLKA DOT                  6 2010-12-01
## 10 536367    84879     ASSORTED COLOUR BIRD ORNAMENT             32 2010-12-01
##    UnitPrice CustomerID Country        SalesLineTotal
##        <dbl>      <dbl> <chr>                   <dbl>
##  1      2.55      17850 United Kingdom           15.3
##  2      3.39      17850 United Kingdom           20.3
##  3      2.75      17850 United Kingdom           22
##  4      3.39      17850 United Kingdom           20.3
##  5      3.39      17850 United Kingdom           20.3
##  6      7.65      17850 United Kingdom           15.3
##  7      4.25      17850 United Kingdom           25.5
##  8      1.85      17850 United Kingdom           11.1
##  9      1.85      17850 United Kingdom           11.1
## 10      1.69      13047 United Kingdom           54.1
## # i 396,036 more rows
```

Nhận xét - Tạo cột chi tiêu (SalesLineTotal)

- Cột SalesLineTotal được tính bằng: Quantity * UnitPrice → Đây là tổng số tiền của mỗi dòng giao dịch (1 dòng = 1 sản phẩm * số lượng trong 1 hóa đơn).
- Biến mới SalesLineTotal đã được tính chính xác và thêm vào bảng df_cleaned.

Ý nghĩa:

- Cột này sẽ đóng vai trò làm biến Monetary trong mô hình RFM (tổng chi tiêu của khách hàng).
- Đây là bước quan trọng để sau đó có thể tổng hợp tổng chi tiêu theo từng CustomerID.

```r
aggregated_df <- summarise(
  group_by(df_cleaned, CustomerID),
  MonetaryValue = sum(SalesLineTotal, na.rm = TRUE),
  Frequency = n_distinct(InvoiceNo),
  LastInvoiceDate = max(InvoiceDate)
)
```

Nhận xét:

- group_by(df_cleaned, CustomerID): Nhóm dữ liệu theo từng khách hàng (CustomerID) – mỗi nhóm tương ứng một khách.
- summarise(…): Tính toán cho từng nhóm:
- MonetaryValue: Tổng giá trị đơn hàng (SalesLineTotal) của khách. na.rm = TRUE giúp bỏ qua giá trị thiếu.
- Frequency: Đếm số hóa đơn khác nhau (InvoiceNo) – biểu hiện cho số lần mua hàng.
- LastInvoiceDate: Ngày mua hàng cuối cùng của khách (dùng để tính Recency)

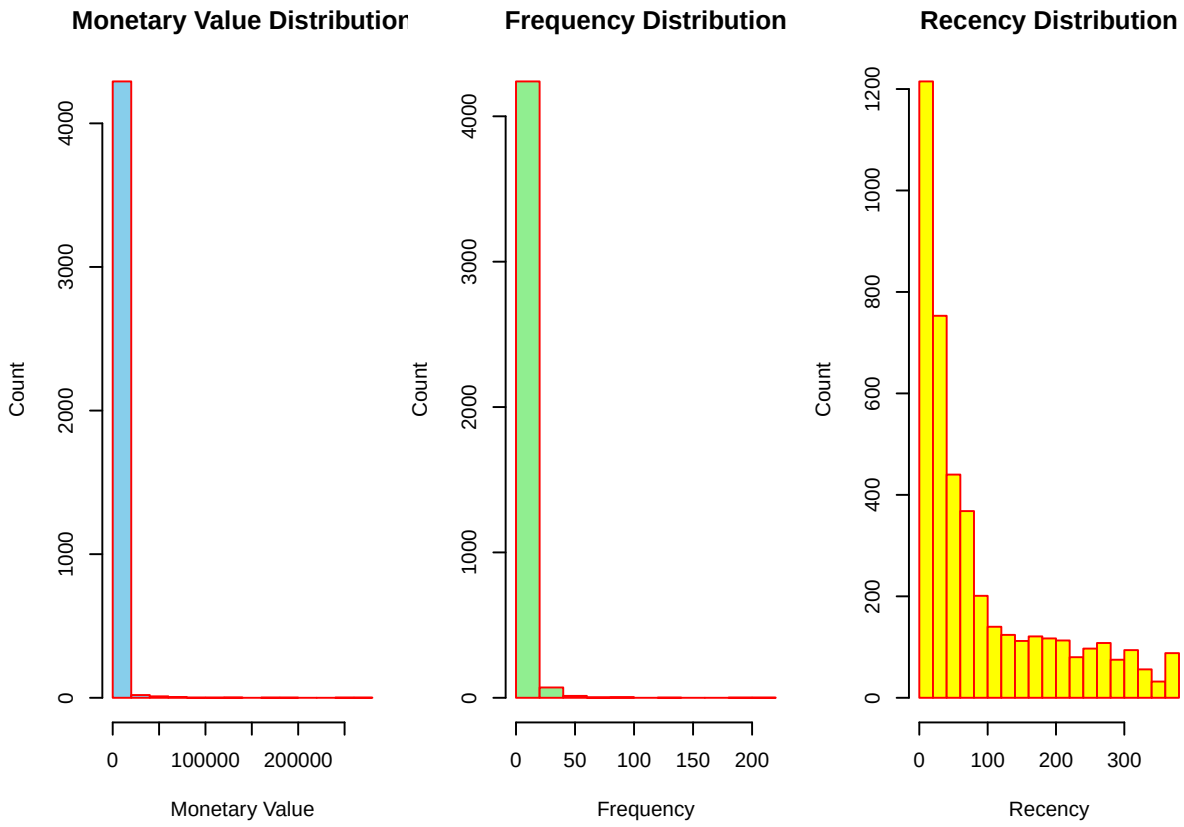
```r
# Tính Recency
analysis_date <- as.Date(max(df_cleaned$InvoiceDate))
aggregated_df$Recency <- as.numeric(analysis_date - aggregated_df$LastInvoiceDate)

print(aggregated_df, width = Inf)
```

```
## # A tibble: 4,334 x 5
##    CustomerID MonetaryValue Frequency LastInvoiceDate Recency
##         <dbl>         <dbl>     <int> <date>            <dbl>
##  1      12346        77184.         1 2011-01-18          325
##  2      12347         4310          7 2011-12-07            2
##  3      12348         1437.         4 2011-09-25           75
##  4      12349         1458.         1 2011-11-21           18
##  5      12350          294.         1 2011-02-02          310
##  6      12352         1386.         7 2011-11-03           36
##  7      12353           89          1 2011-05-19          204
##  8      12354         1079.         1 2011-04-21          232
##  9      12355          459.         1 2011-05-09          214
## 10      12356         2487.         3 2011-11-17           22
## # i 4,324 more rows
```

Nhận xét:

Dữ liệu sau khi tổng hợp theo CustomerID đã tính được đầy đủ 3 chỉ số RFM:

- **MonetaryValue:** Tổng số tiền mà mỗi khách hàng đã chi tiêu, được tính bằng tổng của Quantity * UnitPrice. Các giá trị Monetary cho thấy sự chênh lệch rõ rệt giữa các khách hàng. Ví dụ, khách hàng 12346 chỉ mua 1 lần nhưng chi đến 77,183.60, trong khi các khách hàng khác mua nhiều lần hơn nhưng giá trị nhỏ hơn.
- **Frequency:** Số lượng giao dịch khác nhau (InvoiceNo) mà mỗi khách hàng đã thực hiện. Khách hàng 12347 có tần suất mua hàng cao (7 lần), cho thấy mức độ quay lại cao và có thể là khách hàng trung thành.
- **Recency:** Khoảng cách ngày giữa lần mua gần nhất của khách hàng và ngày phân tích cuối cùng (ngày giao dịch gần nhất trong tập dữ liệu). Khách hàng có giá trị Recency thấp như 12347 (Recency = 2) có nghĩa là họ mới mua hàng gần đây, trong khi những người có Recency cao như 12346 (325 ngày) có thể đã rời đi lâu.



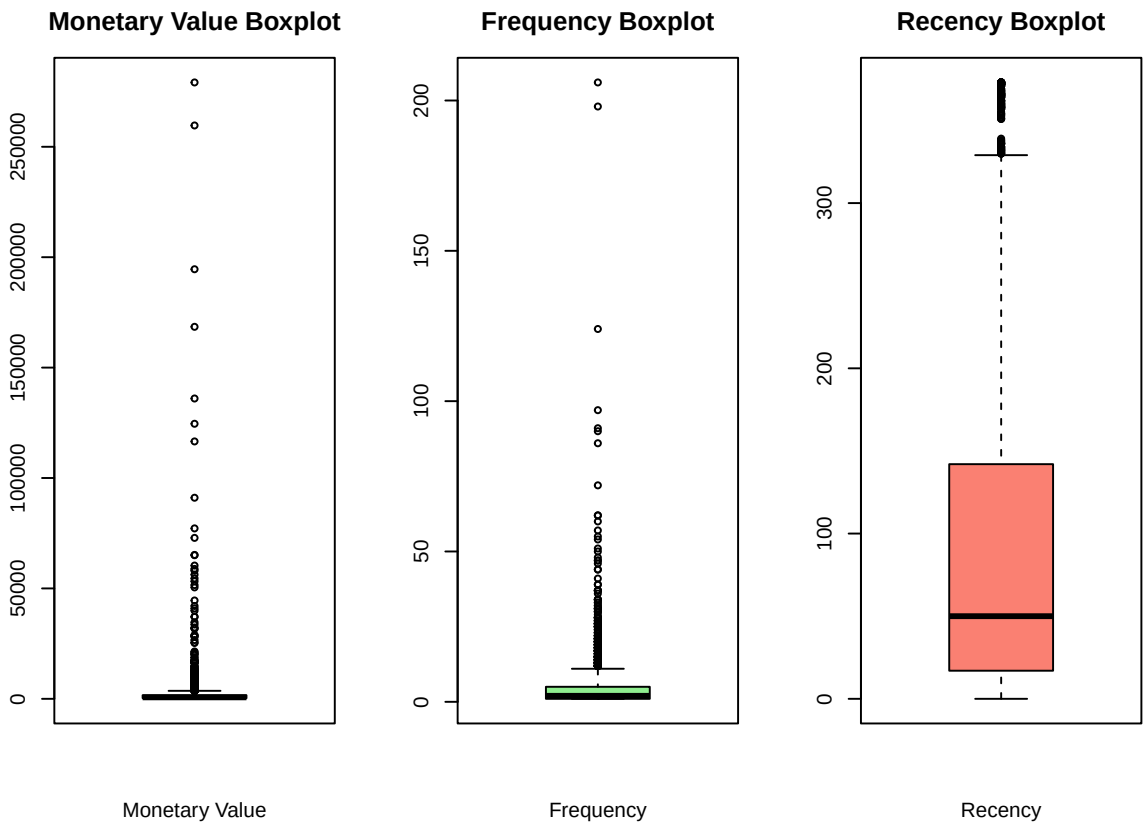
Nhận xét

- **Monetary Value Distribution (Giá trị chi tiêu):**
  - Phân bố rất lệch phải (right-skewed).
  - Phần lớn khách hàng có tổng chi tiêu (Monetary) rất thấp (đa số < 10,000).
  - Chỉ một số ít khách chi tiêu rất cao (có giá trị tới 250,000+), tạo ra outlier rất mạnh.
- **Kết luận:**
  - Tập khách hàng có sự khác biệt lớn về giá trị – phù hợp để phân nhóm (clustering).
  - Nên xem xét log-transform hoặc winsorizing trước khi dùng cho mô hình.
- **Frequency Distribution (Tần suất mua hàng):**
  - Cực kỳ lệch phải: gần như tất cả khách chỉ mua 1–2 lần.
  - Rất ít khách hàng mua từ 10 lần trở lên (hiếm có >50).
  - Một vài giá trị lớn (outlier) vẫn tồn tại.
- **Kết luận:**
  - Phần lớn khách hàng không trung thành → chỉ mua 1 lần.
  - Rất cần phát triển nhóm khách "frequent buyers" (nếu muốn tăng CLV).
  - Có thể tạo chính sách ưu đãi tần suất cho nhóm tiềm năng.
- **Recency Distribution (Khoảng thời gian kể từ lần mua gần nhất):**
  - Phân bố cũng lệch trái.
  - Nhiều khách hàng mới mua gần đây (Recency thấp) → tập trung ở khoảng 0–50 ngày.
  - Càng xa (Recency cao), số lượng khách giảm dần → biểu hiện tốt.

- **Kết luận:**
  - Có nhiều khách hàng mới hoạt động gần đây.
  - Cơ hội rất tốt để chạy lại remarketing hoặc tái kích hoạt khách cũ.
  - Tập khách cũ (Recency > 250) có thể là khách rời bỏ.

Kiểm tra outlier

```
data_outliers <- aggregated_df
```



Monetary Value Boxplot (Tổng chi tiêu):

- Phân bố có rất nhiều outlier (giá trị ngoại lệ) với chi tiêu rất cao.
- Hầu hết khách hàng có mức chi tiêu thấp, trong khi một số ít có giá trị chi tiêu vượt trội, dễ gây ảnh hưởng đến trung bình.
- Điều này gợi ý nên cân nhắc chuẩn hóa hoặc cắt ngưỡng để giảm ảnh hưởng của outlier trong phân cụm.

Frequency Boxplot (Tần suất mua hàng):

- Cũng có nhiều outlier, với một số khách hàng có số lần mua hàng lên đến hơn 200 lần.
- Tuy nhiên, phần lớn khách hàng mua dưới 10 lần.
- Tình trạng phân bố không đều, cho thấy cần xử lý outlier nếu dùng mô hình phân cụm.

Recency Boxplot (Khoảng cách thời gian mua hàng gần nhất):

- Ít outlier hơn so với 2 chỉ số trên.

- Trung vị Recency thấp, cho thấy nhiều khách hàng vẫn hoạt động gần đây.
- Tuy vậy, cũng có một số khách hàng không mua hàng trong thời gian dài.

```
# Tính Q1, Q3 và IQR cho MonetaryValue
M_Q1 <- quantile(aggregated_df$MonetaryValue, 0.25)
M_Q3 <- quantile(aggregated_df$MonetaryValue, 0.75)
M_IQR <- M_Q3 - M_Q1

# Lọc các dòng là outlier
monetary_outliers_df <- aggregated_df[
  aggregated_df$MonetaryValue < (M_Q1 - 1.5 * M_IQR) |
  aggregated_df$MonetaryValue > (M_Q3 + 1.5 * M_IQR),
]

# Xem mô tả dữ liệu outlier
summary(monetary_outliers_df)
```

```
##    CustomerID    MonetaryValue      Frequency      LastInvoiceDate
##  Min.   :12346   Min.   :  3615   Min.   :  1.00   Min.   :2010-12-02
##  1st Qu.:13324   1st Qu.:  4346   1st Qu.:  8.00   1st Qu.:2011-11-15
##  Median :15005   Median :  5888   Median : 12.00   Median :2011-11-29
##  Mean   :15033   Mean   : 12544   Mean   : 16.65   Mean   :2011-11-13
##  3rd Qu.:16655   3rd Qu.:  9431   3rd Qu.: 19.00   3rd Qu.:2011-12-06
##  Max.   :18251   Max.   :279138   Max.   :206.00   Max.   :2011-12-09
##     Recency
##  Min.   :  0.00
##  1st Qu.:  3.00
##  Median : 10.00
##  Mean   : 25.14
##  3rd Qu.: 24.00
##  Max.   :372.00
```

Dựa trên bảng thống kê của summary(monetary_outliers_df):

- Có những khách hàng chi tiêu rất cao, MonetaryValue lên tới 279138, vượt xa giá trị trung bình chung.
- Median chi tiêu của outlier là 5888, trong khi min = 3615 → outlier ở đây chủ yếu là giá trị chi tiêu cao.
- Tần suất mua hàng (Frequency) vẫn ở mức bình thường, không có giá trị cực đoan.
- Recency (gần nhất) cũng đa dạng: từ 0 đến 372 → Có cả khách hàng mới và cũ trong nhóm outlier.

```
# Tính Q1, Q3 và IQR cho Frequency
F_Q1 <- quantile(aggregated_df$Frequency, 0.25)
F_Q3 <- quantile(aggregated_df$Frequency, 0.75)
F_IQR <- F_Q3 - F_Q1

# Lọc các dòng là outlier theo Frequency
frequency_outliers_df <- aggregated_df[
  aggregated_df$Frequency < (F_Q1 - 1.5 * F_IQR) |
  aggregated_df$Frequency > (F_Q3 + 1.5 * F_IQR),
]

# Thống kê dữ liệu outlier
summary(frequency_outliers_df)
```

```
##    CustomerID    MonetaryValue      Frequency      LastInvoiceDate
##  Min.   :12395   Min.   :  1296   Min.   : 12.00   Min.   :2010-12-02
##  1st Qu.:13880   1st Qu.:  4193   1st Qu.: 13.00   1st Qu.:2011-11-23
##  Median :15290   Median :  6284   Median : 17.00   Median :2011-12-03
##  Mean   :15356   Mean   : 14456   Mean   : 23.09   Mean   :2011-11-25
##  3rd Qu.:16781   3rd Qu.: 10995   3rd Qu.: 25.00   3rd Qu.:2011-12-07
##  Max.   :18283   Max.   :279138   Max.   :206.00   Max.   :2011-12-09
##     Recency
##  Min.   :  0.00
##  1st Qu.:  2.00
##  Median :  6.00
##  Mean   : 13.63
##  3rd Qu.: 15.75
##  Max.   :372.00
```

Nhận xét về Frequency Outliers

- **Mục tiêu đoạn mã:**
  - Xác định các điểm ngoại lai (outliers) trong cột Frequency bằng phương pháp IQR (Interquartile Range).
  - Các khách hàng có số lượng giao dịch quá cao so với mức phổ biến sẽ bị coi là outlier.
- **Kết quả lọc:**
  - Dữ liệu sau khi lọc chỉ còn những khách hàng có số giao dịch cao bất thường (từ 12 đến 206 lần).
  - Median Frequency = 17, 75% khách hàng trong tập này có Frequency ≤ 25.
  - Tuy nhiên, số lượng khách hàng như vậy không nhiều (chỉ là tập nhỏ so với toàn bộ dữ liệu).

```r
# Loại bỏ outlier bằng cách giữ lại các dòng KHÔNG nằm trong monetary và frequency outliers

# Lấy chỉ số hàng (index) cần loại bỏ
outlier_indices <- union(
  which(aggregated_df$MonetaryValue < (M_Q1 - 1.5 * M_IQR) | aggregated_df$MonetaryValue > (M_Q3 + 1
  which(aggregated_df$Frequency < (F_Q1 - 1.5 * F_IQR) | aggregated_df$Frequency > (F_Q3 + 1.5 * F_I
)

# Giữ lại các dòng không nằm trong chỉ số outlier
non_outliers_df <- aggregated_df[-outlier_indices, ]

# Kiểm tra lại
summary(non_outliers_df)
```

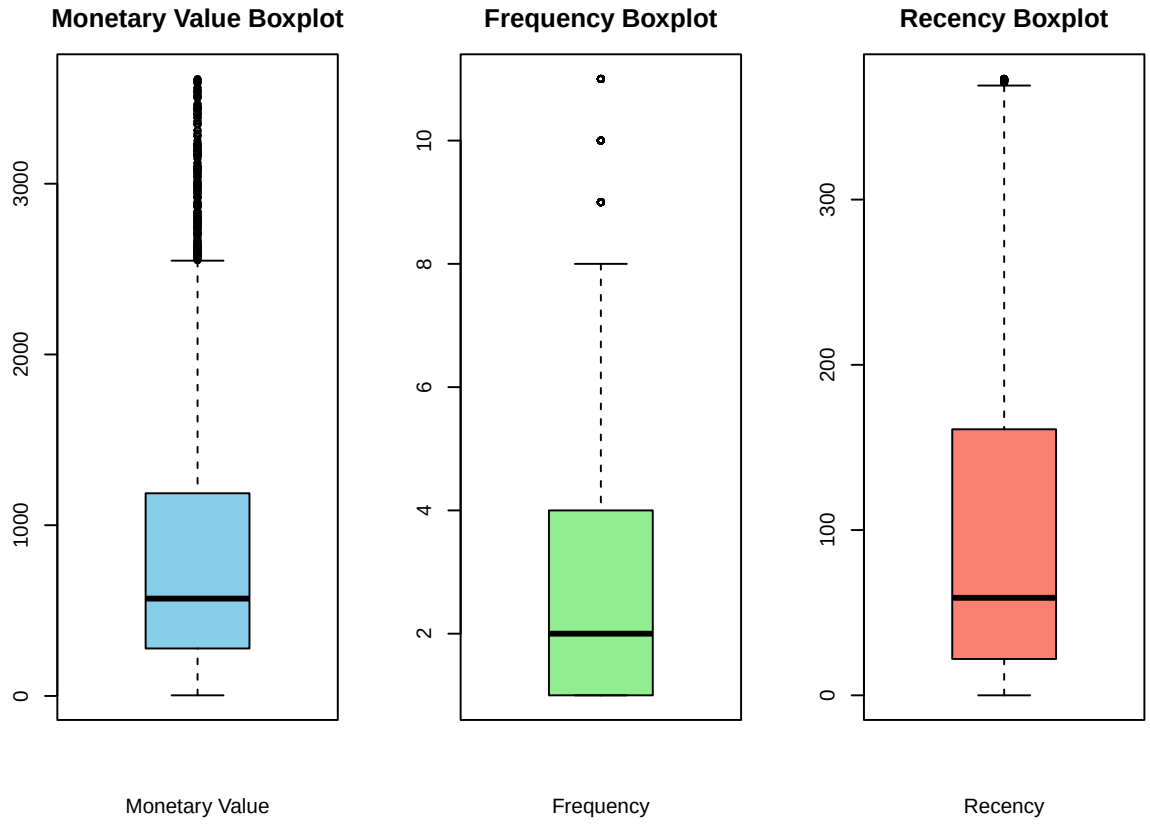
```
##    CustomerID    MonetaryValue      Frequency      LastInvoiceDate
##  Min.   :12348   Min.   :   3.75   Min.   : 1.000   Min.   :2010-12-01
##  1st Qu.:13852   1st Qu.: 278.05   1st Qu.: 1.000   1st Qu.:2011-07-01
##  Median :15333   Median : 570.13   Median : 2.000   Median :2011-10-11
##  Mean   :15324   Mean   : 852.75   Mean   : 2.755   Mean   :2011-08-30
##  3rd Qu.:16798   3rd Qu.:1187.01   3rd Qu.: 4.000   3rd Qu.:2011-11-17
##  Max.   :18287   Max.   :3613.63   Max.   :11.000   Max.   :2011-12-09
##     Recency
##  Min.   :  0.0
##  1st Qu.: 22.0
##  Median : 59.0
##  Mean   :100.5
##  3rd Qu.:161.0
##  Max.   :373.0
```

Giải thích:

- M_Q1, M_Q3, M_IQR: lần lượt là Q1, Q3 và khoảng IQR của biến MonetaryValue (tính trước đó).
- F_Q1, F_Q3, F_IQR: là Q1, Q3, IQR của biến Frequency.
- which(…): trả về chỉ số dòng (index) của các giá trị là outlier.
- union(…): hợp nhất chỉ số outlier của cả hai biến MonetaryValue và Frequency để tránh trùng lặp và tạo 1 danh sách duy nhất các dòng cần loại bỏ.
- Điều kiện outlier theo IQR:
  - Bất kỳ giá trị nào < Q1 - 1.5 * IQR hoặc > Q3 + 1.5 * IQR được coi là outlier.

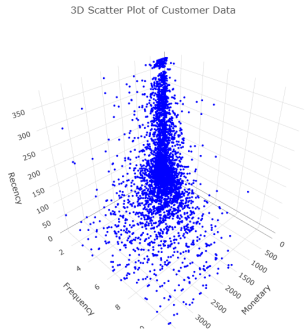
Nhận xét:

- **MonetaryValue:**
  - Min: 3.75, Max: 3613.63 → loại bỏ hoàn toàn các khách hàng chi tiêu quá thấp hoặc quá cao.
  - Mean và Median khá gần nhau → phân phối cân bằng hơn sau khi loại outlier.
- **Frequency:**
  - Min: 1, Max: 11 → tần suất mua hàng không còn giá trị cực đại như trước (trước đó là 206).
- **Recency:**
  - Vẫn giữ được giá trị phân tán rộng từ 0 đến 373 → không bị loại bỏ trong bước này vì không xử lý outlier cho Recency.



Nhận xét Boxplot sau khi loại bỏ outlier:

- **Monetary Value (Giá trị chi tiêu):**
  - Sau khi loại bỏ outlier, dữ liệu tập trung phần lớn ở khoảng dưới 1000.
  - Tuy vẫn còn một vài giá trị cao nằm gần ngưỡng cắt trên, nhưng không còn cực trị quá lớn (trên 250,000 như trước đó).
- **Frequency (Tần suất mua hàng):**
  - Đa số khách hàng có số lần giao dịch từ 1 đến 4.
  - Một vài điểm nhỏ (vẫn là mild outliers) nằm ngoài whiskers nhưng rất ít, cho thấy dữ liệu đã được làm sạch tương đối tốt.
  - Median = 2 cho thấy phần lớn khách hàng mua 1-2 lần.
- **Recency (Số ngày kể từ lần mua cuối):**
  - Phân bố tương đối đều hơn so với trước, tuy nhiên vẫn còn giá trị cao trên 300.
  - Phần lớn khách hàng mua trong vòng 100 ngày gần đây.
  - Median khoảng 59, tức là một nửa số khách hàng mua hàng lần cuối cách thời điểm phân tích 2 tháng trở lại.



Hình 14: 3D Scatter Plot of Customer Data

Nhận xét biểu đồ 3D Scatter Plot of Customer Data:

Biểu đồ thể hiện phân bố ba chỉ số RFM (Recency – Frequency – Monetary) của tập khách hàng sau khi loại bỏ outlier:

- **Monetary (Trục X):**
  - Đa số khách hàng có Monetary dưới 1000 (tức là chi tiêu thấp).
  - Có một số điểm rải rác đến mức gần 3500, tuy nhiên không nhiều → Phân bố lệch phải (skewed right).
- **Frequency (Trục Y):**
  - Hầu hết khách hàng có tần suất mua hàng thấp, chủ yếu từ 1–4 lần.
  - Một số khách hàng mua nhiều lần (tới 10+), nhưng rất ít → Tương tự là lệch phải.
- **Recency (Trục Z):**
  - Có xu hướng phân tán dọc theo trục Recency.
  - Khách hàng gần đây (Recency thấp) tập trung nhiều hơn ở vùng gần Monetary cao.
  - Những người mua hàng từ lâu (Recency cao, gần 350 ngày) có xu hướng chi tiêu ít hơn.

```r
# Chuẩn hóa các cột MonetaryValue, Frequency, Recency
scaled_data <- scale(non_outliers_df[, c("MonetaryValue", "Frequency", "Recency")])
```

Mục tiêu

- Chuẩn hóa ba chỉ số RFM (`MonetaryValue`, `Frequency`, `Recency`) bằng phương pháp z-score để đưa về cùng đơn vị thang đo, phục vụ cho bước phân cụm và giảm chiều sau đó.

```r
# # Chuyển matrix chuẩn hóa về data.frame
 scaled_data_df <- as.data.frame(scaled_data)
#
# # Đặt lại tên cột
colnames(scaled_data_df) <- c("MonetaryValue", "Frequency", "Recency")
#
# # Gán lại rownames nếu cần giữ index
rownames(scaled_data_df) <- rownames(non_outliers_df)
# # Xem kết quả
head(scaled_data_df)
```

```
##   MonetaryValue  Frequency    Recency
## 1     0.7366723  0.5745266 -0.2501914
## 2     0.7622704 -0.8094244 -0.8098206
## 3    -0.7037273 -0.8094244  2.0570517
## 4     0.6717634  1.9584775 -0.6330956
## 5    -0.9626070 -0.8094244  1.0163378
## 6     0.2856621 -0.8094244  1.2912433
```

Nhận xét

- Ba cột MonetaryValue, Frequency và Recency đã được chuẩn hóa về trung bình 0, phương sai 1 bằng phương pháp z-score.
- Sau khi chuẩn hóa:
  - Những khách hàng có giá trị MonetaryValue dương → có tổng chi tiêu cao hơn trung bình.
  - Những khách hàng có Recency lớn (giá trị z lớn) → đã lâu không mua hàng.
  - Tần suất Frequency thấp thường có giá trị âm, cho thấy mức độ tương tác không thường xuyên.
- Dữ liệu sau chuẩn hóa phù hợp để thực hiện phân cụm bằng các thuật toán như K-Means hoặc DBSCAN, hoặc dùng cho PCA để giảm chiều.

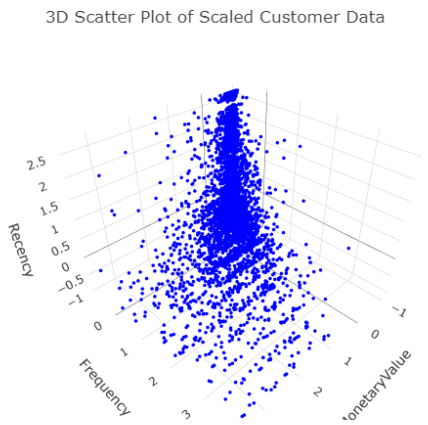
Nhận xét biểu đồ 3D Scatter Plot của dữ liệu RFM đã chuẩn hóa:

Biểu đồ thể hiện mối quan hệ giữa 3 chỉ số sau khi chuẩn hóa:

- **MonetaryValue** (trục X – chi tiêu)
- **Frequency** (trục Y – tần suất mua hàng)
- **Recency** (trục Z – độ gần lần mua cuối)

Các nhận xét chính:

- **Dữ liệu phân bố chủ yếu quanh gốc tọa độ (0, 0, 0):**
  - Do đã chuẩn hóa bằng z-score, phần lớn khách hàng có các chỉ số gần với trung bình của toàn bộ tập dữ liệu.



Hình 15: 3D Scatter Plot of Scaled Customer Data

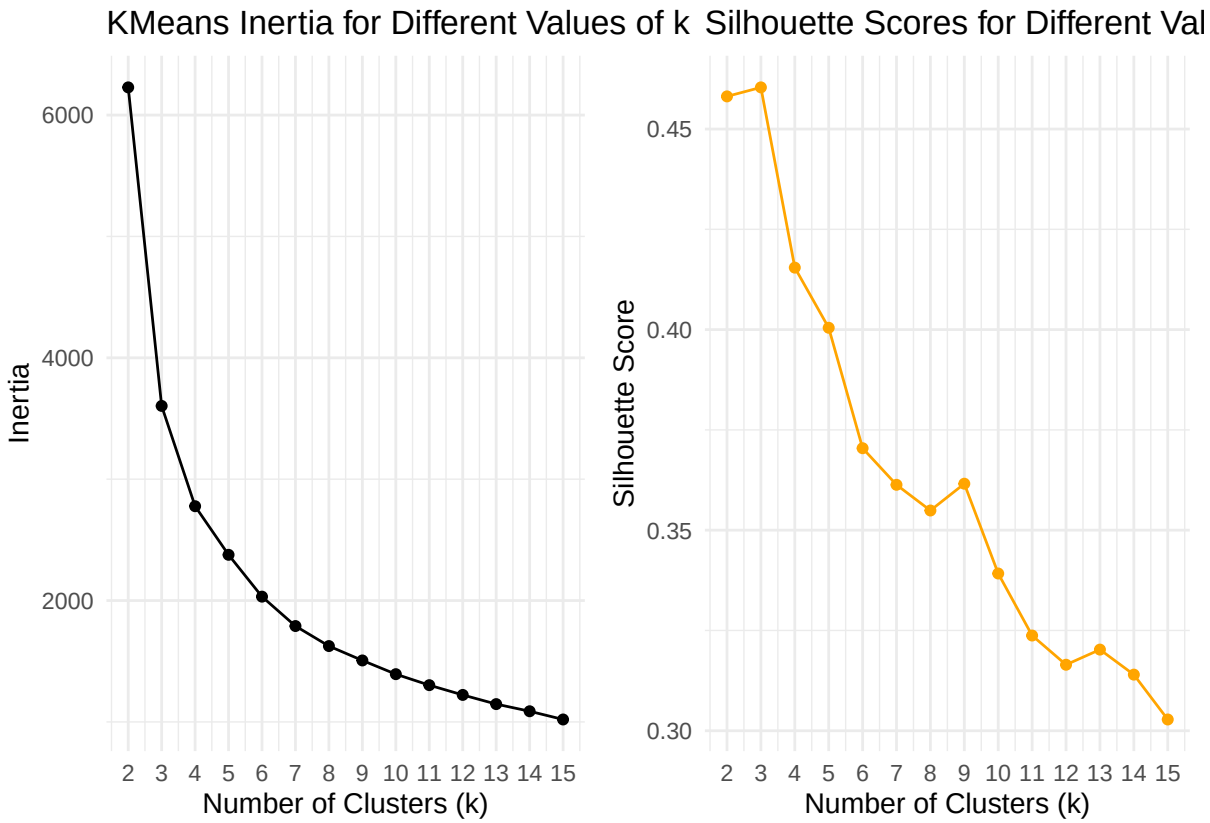
- **Dữ liệu có hình tháp dày đặc ở trung tâm:**
  - Chứng tỏ đa số khách hàng có hành vi "trung bình" – mua hàng với tần suất, chi tiêu và độ gần ở mức phổ biến.
- **Có một số điểm nằm rải rác xa tâm:**
  - Đây có thể là các khách hàng tiềm năng (chi tiêu cao, mua gần đây) hoặc ngược lại (mua từ lâu, ít chi tiêu), cần được phân cụm riêng biệt trong bước tiếp theo.
- **Không có outlier rõ ràng:**
  - Vì đã loại bỏ ngoại lệ từ trước, biểu đồ không còn các điểm quá lệch so với cụm chính.
- Biểu đồ này cho thấy dữ liệu đã **sẵn sàng để:** Phân cụm bằng **K-Means, DBSCAN**.

3.4 Phân cụm khách hàng bằng thuật toán K-Means

Thuật toán **K-Means** là một trong những phương pháp phân cụm phổ biến, được sử dụng để chia các khách hàng thành các nhóm đồng nhất dựa trên đặc điểm hành vi mua hàng. Trong phần này, chúng tôi áp dụng thuật toán K-Means lên dữ liệu đã chuẩn hóa từ mô hình RFM.

```r
# Thiết lập
max_k <- 15
k_values <- 2:max_k

inertia <- numeric(length(k_values))
silhouette_scores <- numeric(length(k_values))

for (i in seq_along(k_values)) {
  k <- k_values[i]
  set.seed(42)
  kmeans_model <- kmeans(scaled_data_df, centers = k, nstart = 25, iter.max = 1000)

  # inertia
  inertia[i] <- kmeans_model$tot.withinss

  # silhouette
  sil <- silhouette(kmeans_model$cluster, dist(scaled_data_df))
  silhouette_scores[i] <- mean(sil[, 3])
}
```

```r
# Tạo dataframe cho ggplot
plot_df <- data.frame(
  k = k_values,
  Inertia = inertia,
  Silhouette = silhouette_scores
)

# Biểu đồ Elbow
p1 <- ggplot(plot_df, aes(x = k, y = Inertia)) +
  geom_line() +
  geom_point() +
  labs(title = "KMeans Inertia for Different Values of k",
       x = "Number of Clusters (k)", y = "Inertia") +
  theme_minimal() +
  scale_x_continuous(breaks = k_values)

# Biểu đồ Silhouette
p2 <- ggplot(plot_df, aes(x = k, y = Silhouette)) +
  geom_line(color = "orange") +
  geom_point(color = "orange") +
  labs(title = "Silhouette Scores for Different Values of k",
       x = "Number of Clusters (k)", y = "Silhouette Score") +
  theme_minimal() +
  scale_x_continuous(breaks = k_values)

# Hiển thị cả hai biểu đồ cạnh nhau
grid.arrange(p1, p2, ncol = 2)
```



Nhận xét:

- **Biểu đồ bên trái – Elbow Method (Inertia):**

  - Giá trị Inertia giảm mạnh từ k = 2 đến k = 4, sau đó bắt đầu giảm chậm lại tạo thành một "khuỷu tay" tại k = 4.
  - Đây là dấu hiệu rõ ràng của số cụm tối ưu, vì tại điểm này mô hình đã gom được phần lớn thông tin phân tán và việc tăng thêm cụm sau đó không làm giảm Inertia đáng kể nữa.

- **Biểu đồ bên phải – Silhouette Score:**
  - Silhouette đạt giá trị cao nhất ở k = 2 và k = 3 (~0.45), sau đó giảm dần.
  - Tuy nhiên, k = 4 vẫn duy trì mức Silhouette hợp lý (~0.40), cho thấy phân cụm vẫn đủ rõ nét mà không quá đơn giản.
  - Việc chọn k = 4 là sự cân bằng giữa độ rõ ràng và độ chi tiết của các cụm khách hàng.

Kết luận:

Dựa trên cả hai biểu đồ, số lượng cụm tối ưu được lựa chọn là k = 4. Đây là lựa chọn hợp lý để tiến hành phân cụm K-Means trong bước tiếp theo, vừa đảm bảo tính phân biệt giữa các cụm, vừa tránh tình trạng quá phân mảnh hoặc đơn giản hóa mô hình.

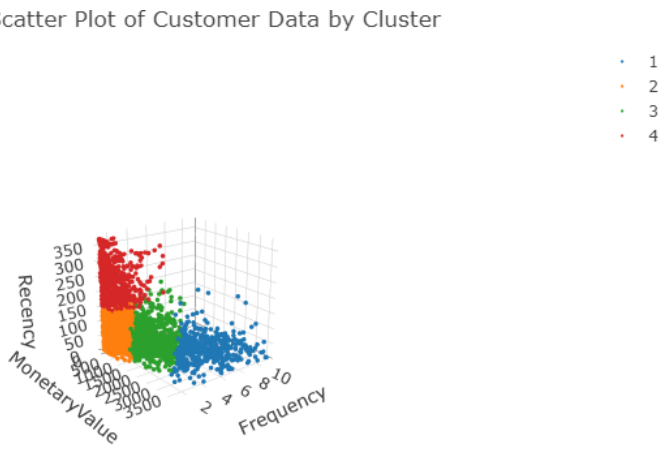
```
library(scatterplot3d)  # vẽ 3D scatter

# Huấn luyện mô hình KMeans
set.seed(42)
kmeans_model <- kmeans(scaled_data_df, centers = 4, nstart = 25, iter.max = 1000)

# Gán nhãn cụm vào non_outliers_df
non_outliers_df$Cluster <- kmeans_model$cluster

# Màu thủ công cho từng cụm
cluster_colors <- c("#1f77b4", "#ff7f0e", "#2ca02c", "#d62728")
color_map <- cluster_colors[non_outliers_df$Cluster]

# plot_ly(non_outliers_df,
#         x = ~MonetaryValue,
#         y = ~Frequency,
#         z = ~Recency,
#         color = ~factor(Cluster),
#         colors = cluster_colors,
#         type = 'scatter3d',
#         mode = 'markers',
#         marker = list(size = 2)) %>%
#   layout(title = "3D Scatter Plot of Customer Data by Cluster")
```

Nhận xét:

Biểu đồ 3D thể hiện sự phân bố của các khách hàng theo 3 chiều **Recency**, **Frequency**, và **MonetaryValue**, được phân cụm bằng thuật toán K-Means với số cụm k = 4.

- Nhìn chung, các cụm được tách biệt tương đối rõ, đặc biệt là theo trục Monetary và Frequency.
- **Cụm màu đỏ** (Cluster 4) tập trung ở vùng Recency cao, Frequency thấp, Monetary thấp -> Đây có thể là nhóm khách hàng không còn mua hàng gần đây và giá trị thấp.
- **Cụm màu xanh lam** (Cluster 1) nằm ở đáy biểu đồ, với Recency thấp, Frequency cao và Monetary rất cao → nhóm khách hàng trung thành và có giá trị lớn.
- Các cụm còn lại nằm xen giữa, cho thấy các nhóm khách hàng có hành vi mua hàng trung bình hoặc không đều đặn.

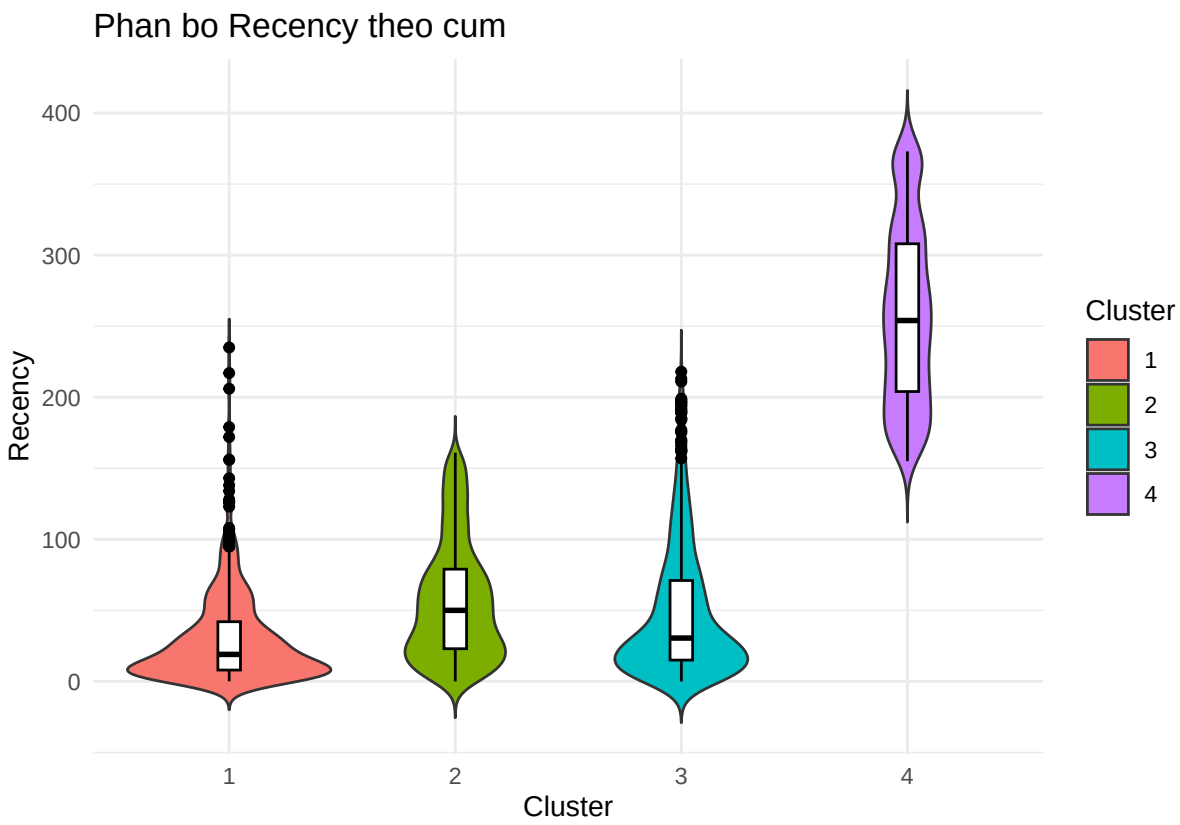
Ý nghĩa:



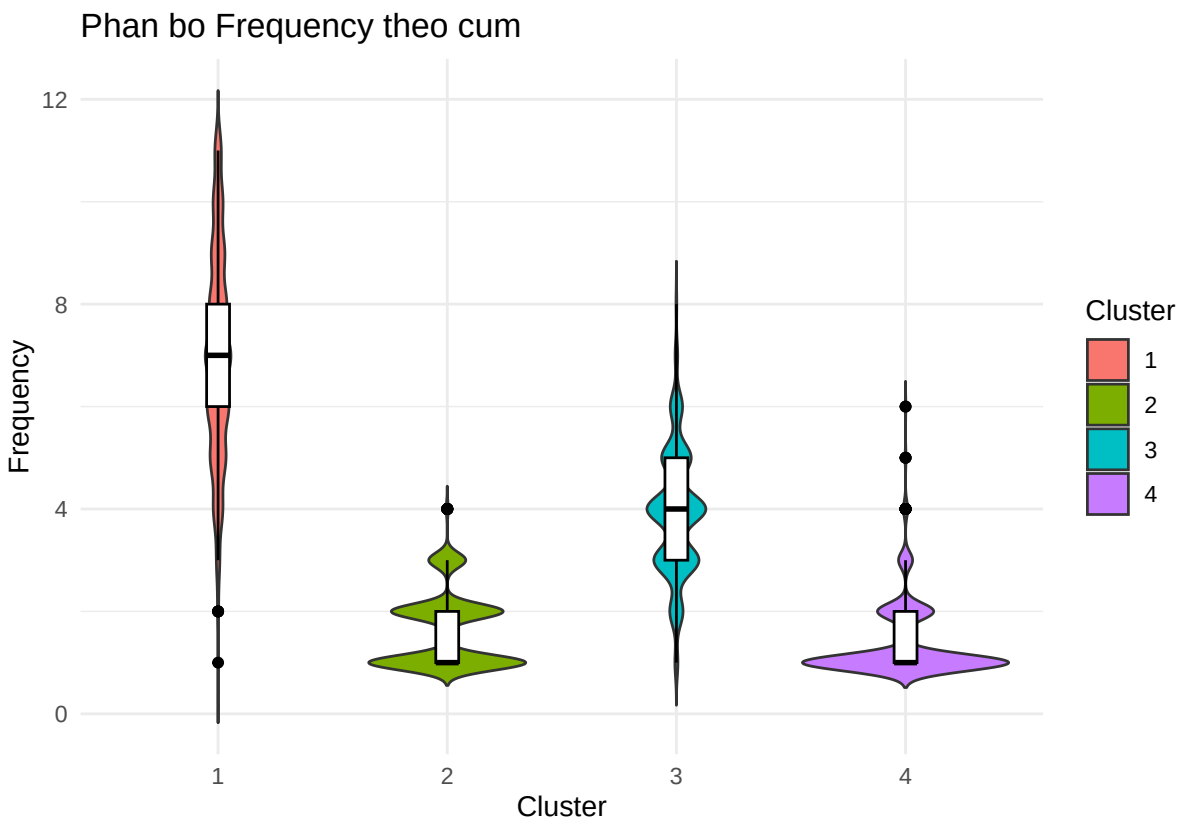
Hình 16: 3D Scatter Plot of Customer Data by Cluster

- Việc trực quan hóa dữ liệu RFM theo không gian 3 chiều giúp kiểm chứng lại hiệu quả phân cụm.
- Doanh nghiệp có thể dễ dàng xác định nhóm khách hàng tiềm năng để duy trì và nhóm không hiệu quả để tiết giảm chi phí tiếp thị.
- Đây là bước quan trọng trong các chiến lược **phân khúc thị trường** và **giữ chân khách hàng**.

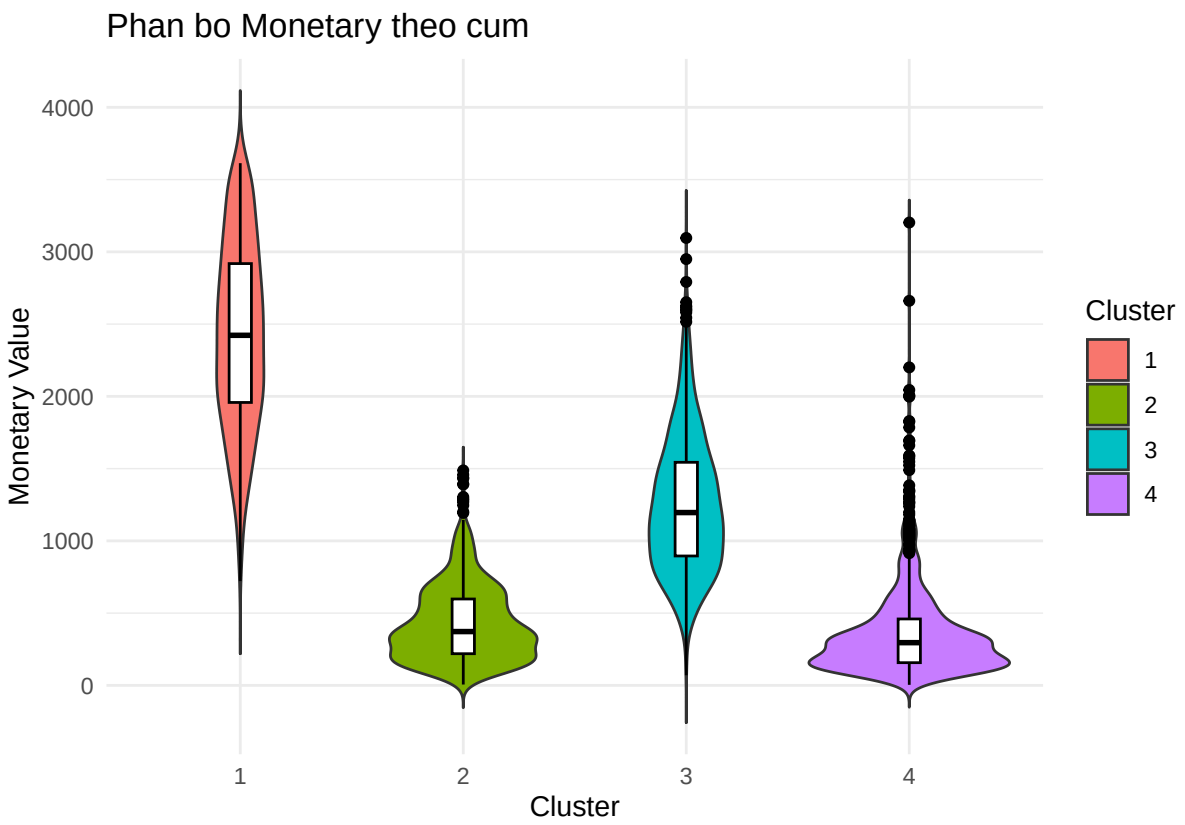
```r
# Đảm bảo Cluster là factor
non_outliers_df$Cluster <- as.factor(non_outliers_df$Cluster)

# Tạo bảng màu tương ứng như Python
cluster_colors <- c("0" = "#1f77b4", "1" = "#ff7f0e", "2" = "#2ca02c", "3" = "#d62728")

# Violin plot cho Recency
ggplot(non_outliers_df, aes(x = Cluster, y = Recency, fill = Cluster)) +
  geom_violin(trim = FALSE) +
  geom_boxplot(width = 0.1, color = "black", fill = "white") +
  labs(title = "Phan bo Recency theo cum", y = "Recency") +
  theme_minimal()
```




```r
# Violin plot cho Frequency
ggplot(non_outliers_df, aes(x = Cluster, y = Frequency, fill = Cluster)) +
  geom_violin(trim = FALSE) +
  geom_boxplot(width = 0.1, color = "black", fill = "white") +
  labs(title = "Phan bo Frequency theo cum", y = "Frequency") +
  theme_minimal()
```




```r
# Tương tự cho Monetary
ggplot(non_outliers_df, aes(x = Cluster, y = MonetaryValue, fill = Cluster)) +
  geom_violin(trim = FALSE) +
  geom_boxplot(width = 0.1, color = "black", fill = "white") +
  labs(title = "Phan bo Monetary theo cum", y = "Monetary Value") +
  theme_minimal()
```



Nhận xét: Phân bố Recency

- Cụm 1 (màu đỏ): Có giá trị Recency rất thấp, phân bố tập trung quanh 10–30 ngày, cho thấy đây là nhóm khách hàng vừa mới mua hàng gần đây nhất -> nhóm rất tiềm năng cần được giữ chân.
- Cụm 2 (màu xanh lá): Recency trải rộng từ thấp đến khoảng 100 ngày, nhưng phần lớn tập trung dưới 50 -> nhóm khách hàng có mua gần đây, nhưng không đều đặn bằng cụm 1.
- Cụm 3 (màu xanh dương): Tương tự cụm 1, nhưng phân bố hơi rộng hơn -> vẫn là khách hàng mới tương tác gần đây, có khả năng mua lại nếu chăm sóc tốt.
- Cụm 4 (màu tím): Có Recency rất cao, phần lớn nằm quanh 200–350 ngày -> đây là nhóm khách hàng đã lâu không quay lại, có nguy cơ rời bỏ thương hiệu, cần có chính sách tái tiếp cận hoặc remarketing.

Ý nghĩa:

- Biểu đồ này giúp xác định mức độ "gần đây" của các giao dịch theo từng nhóm khách hàng. Qua đó:
  - Cụm 1 & 3 -> nên ưu tiên chăm sóc, upsell hoặc tặng ưu đãi đặc biệt.
  - Cụm 4 -> cần có chiến dịch tái tương tác, khuyến mãi "quay lại", email marketing hoặc tặng mã giảm giá.
  - Phân tích Recency đóng vai trò quan trọng trong việc dự đoán hành vi mua hàng sắp tới.

Nhận xét: Phân bố Frequency:

- Cụm 1 (màu đỏ): Có giá trị Frequency cao nhất trong các cụm, phần lớn dao động từ 6 đến 9, thậm chí có điểm lên đến 12.
- Đây là nhóm mua hàng nhiều lần, thường xuyên quay lại -> nhóm trung thành, có mức độ gắn kết cao với doanh nghiệp.
- Cụm 2 (màu xanh lá): Tần suất mua hàng thấp, hầu hết dao động trong khoảng 1 đến 3 lần.

- Nhóm này có thể là khách hàng mua thử, hoặc chỉ tương tác ngắn hạn -> cần được kích thích mua lại.
- Cụm 3 (màu xanh dương): Có Frequency ở mức trung bình, phổ biến trong khoảng 3–6 lần.
- Đây là nhóm tiềm năng – chưa thực sự trung thành, nhưng có dấu hiệu mua lặp lại -> nên được nuôi dưỡng.
- Cụm 4 (màu tím): Có Frequency rất thấp, phần lớn chỉ 1–2 lần mua.
- Kết hợp với Recency cao (từ biểu đồ trước) -> có khả năng là nhóm mua một lần rồi rời bỏ, ít giá trị.

Ý nghĩa:

- Biểu đồ giúp doanh nghiệp dễ dàng nhận biết nhóm khách nào có hành vi mua lặp lại cao để tập trung chăm sóc và giữ chân.
- Các cụm có Frequency thấp nhưng Recency cũng thấp (như cụm 2) → có thể vẫn còn cơ hội chuyển hóa nếu tiếp thị đúng lúc.
- Kết hợp phân tích Frequency với Recency sẽ giúp xác định tệp khách hàng trung thành vs khách "một lần rồi thôi", từ đó ra quyết định marketing hiệu quả hơn.

Nhận xét: Phân bố Monetary:

- Cụm 1 (màu đỏ): Đây là nhóm có giá trị Monetary cao nhất, phần lớn dao động từ 2.000 đến gần 4.000.
- Đây chắc chắn là nhóm khách hàng có giá trị lớn nhất, thường xuyên chi tiêu cao, đóng góp nhiều cho doanh thu -> nhóm VIP hoặc khách hàng trung thành nhất.
- Cụm 2 (màu xanh lá): Monetary rất thấp, phần lớn chỉ ở mức 200 – 600.
- Đây có thể là nhóm khách ít mua và mua với giá trị thấp, có thể là khách vãng lai hoặc mới mua thử.
- Cụm 3 (màu xanh dương): Có mức chi tiêu ở mức trung bình khá, trải rộng từ khoảng 800 – 2500, một số điểm còn lên cao hơn.
- Đây là nhóm khách có tiềm năng, nếu được thúc đẩy đúng cách có thể chuyển hóa thành khách hàng trung thành.
- Cụm 4 (màu tím): Có giá trị Monetary thấp nhất trong tất cả các cụm, phần lớn dao động từ dưới 500 đến khoảng 800.
- Kết hợp với Recency cao và Frequency thấp -> nhóm đã lâu không quay lại, mua ít, không thu lợi cao.

Ý nghĩa:

- Biểu đồ Monetary giúp doanh nghiệp xác định giá trị tài chính của từng nhóm khách hàng.
- Cụm 1 nên được xem là tệp khách hàng ưu tiên cao nhất, có thể áp dụng chiến lược chăm sóc đặc biệt như VIP member, thẻ tích điểm, quà tặng cá nhân,…
- Cụm 3 là nhóm có khả năng phát triển, nếu được tiếp thị và chăm sóc hợp lý.
- cụm 4 nên được đánh giá lại chi phí giữ chân để tránh lãng phí nguồn lực.

Kết luận:

Sau khi phân tích trực quan từng thành phần của mô hình RFM theo từng cụm khách hàng, có thể rút ra các nhận định quan trọng như sau:

- Cụm 1 (màu đỏ):
  - Đây là nhóm khách hàng tốt nhất:
  - Mua gần đây (Recency thấp),

  - Mua nhiều lần (Frequency cao),
  - Chi tiêu lớn (Monetary cao).

-> Đây là nhóm khách hàng trung thành, có giá trị cao, cần được ưu tiên chăm sóc, giữ chân và tạo các chương trình khuyến khích tái mua.

- Cụm 2 (màu xanh lá):
  - Có tần suất mua thấp, giá trị đơn hàng nhỏ và thời gian mua gần đây tương đối ổn.

-> Nhóm mới bắt đầu tương tác hoặc khách hàng mua thử. Doanh nghiệp có thể tập trung nuôi dưỡng nhóm này bằng chương trình chăm sóc khuyến mãi, ưu đãi.

- Cụm 3 (màu xanh dương):
  - Mua không quá thường xuyên nhưng có chi tiêu trung bình khá và mua tương đối gần đây.

-> Là nhóm khách hàng tiềm năng, có thể trở thành trung thành nếu được chăm sóc và có các chiến lược tiếp thị phù hợp.

- Cụm 4 (màu tím):
  - Giao dịch đã lâu (Recency cao), tần suất mua rất thấp và giá trị chi tiêu nhỏ.

-> Đây là nhóm khách hàng không hoạt động, có thể đã rời bỏ thương hiệu. Cần xem xét chiến lược tiếp cận lại hoặc loại khỏi các chiến dịch ưu tiên để tiết kiệm nguồn lực.

```
table(non_outliers_df$Cluster) #Số lượng khách hàng từng cụm
```

```
##
##    1    2    3    4
##  482 1547  890  944
```

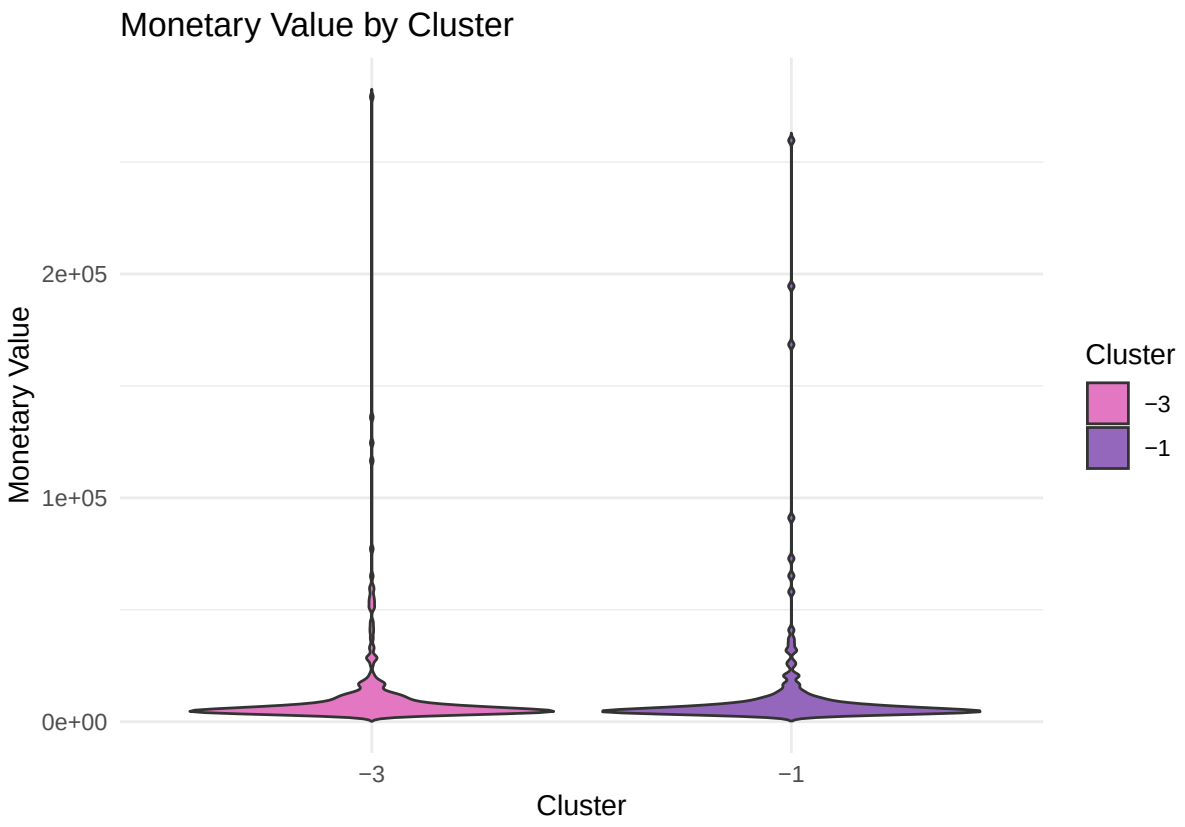
```
# Tìm chỉ số giao nhau giữa 2 tập outliers
overlap_indices <- intersect(rownames(monetary_outliers_df), rownames(frequency_outliers_df))

# Tạo tập chỉ là Monetary outliers
monetary_only_outliers <- monetary_outliers_df[!rownames(monetary_outliers_df) %in% overlap_indices,
monetary_only_outliers$Cluster <- -1

# Tạo tập chỉ là Frequency outliers
frequency_only_outliers <- frequency_outliers_df[!rownames(frequency_outliers_df) %in% overlap_indic
frequency_only_outliers$Cluster <- -2

# Tạo tập là outliers của cả hai
monetary_and_frequency_outliers <- monetary_outliers_df[rownames(monetary_outliers_df) %in% overlap_
monetary_and_frequency_outliers$Cluster <- -3

# Gộp tất cả lại
outlier_clusters_df <- rbind(
  monetary_only_outliers,
  frequency_only_outliers,
  monetary_and_frequency_outliers
)

# Kết quả
head(outlier_clusters_df)
```

```
## # A tibble: 6 x 6
##   CustomerID MonetaryValue Frequency LastInvoiceDate Recency Cluster
##        <dbl>         <dbl>     <int> <date>            <dbl>   <dbl>
## 1      15971         4195.        12 2011-11-22           17      -1
## 2      16000        12394.         3 2011-12-07            2      -1
## 3      16013        37131.        47 2011-12-06            3      -1
## 4      16019         3733.         9 2011-10-24           46      -1
## 5      16029        72882.        62 2011-11-01           38      -1
## 6      16033         8804.        20 2011-12-04            5      -1
```

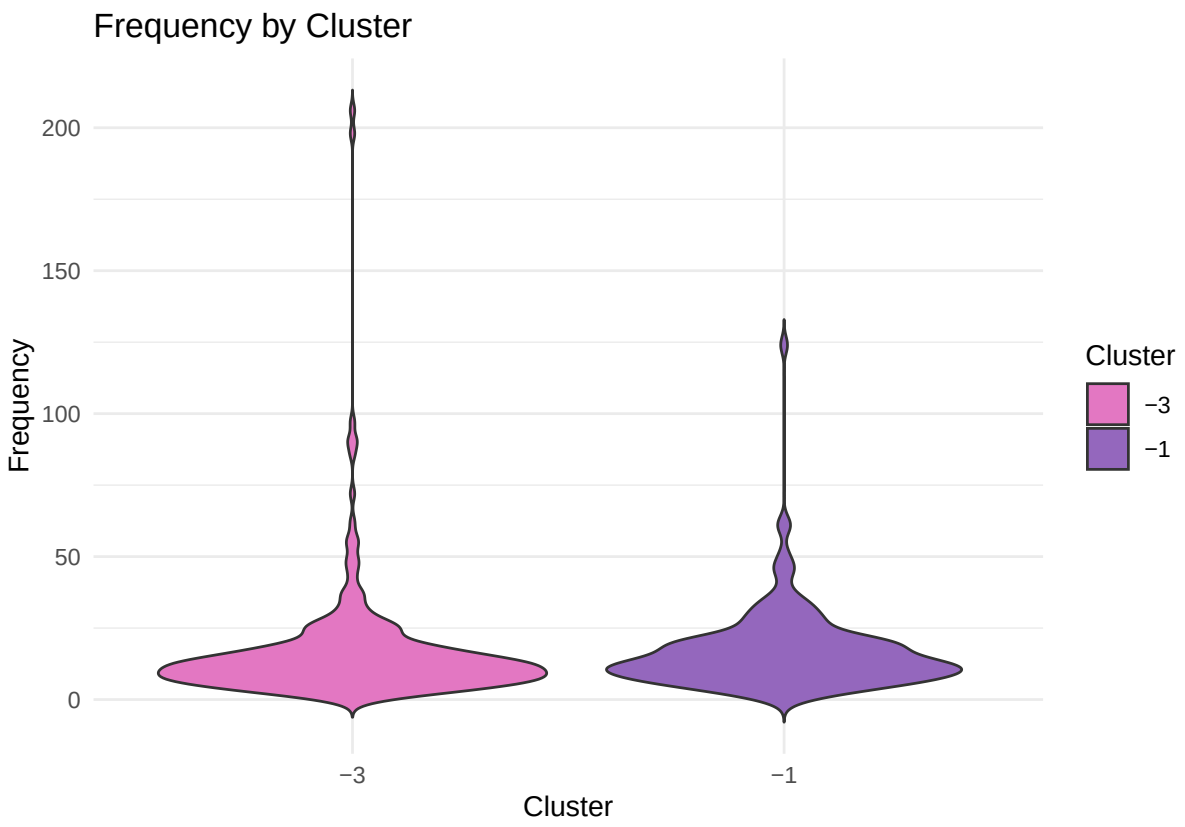
```r
# Kiểm tra
table(outlier_clusters_df$Cluster)
```

```
##
##  -3  -1
## 278 147
```

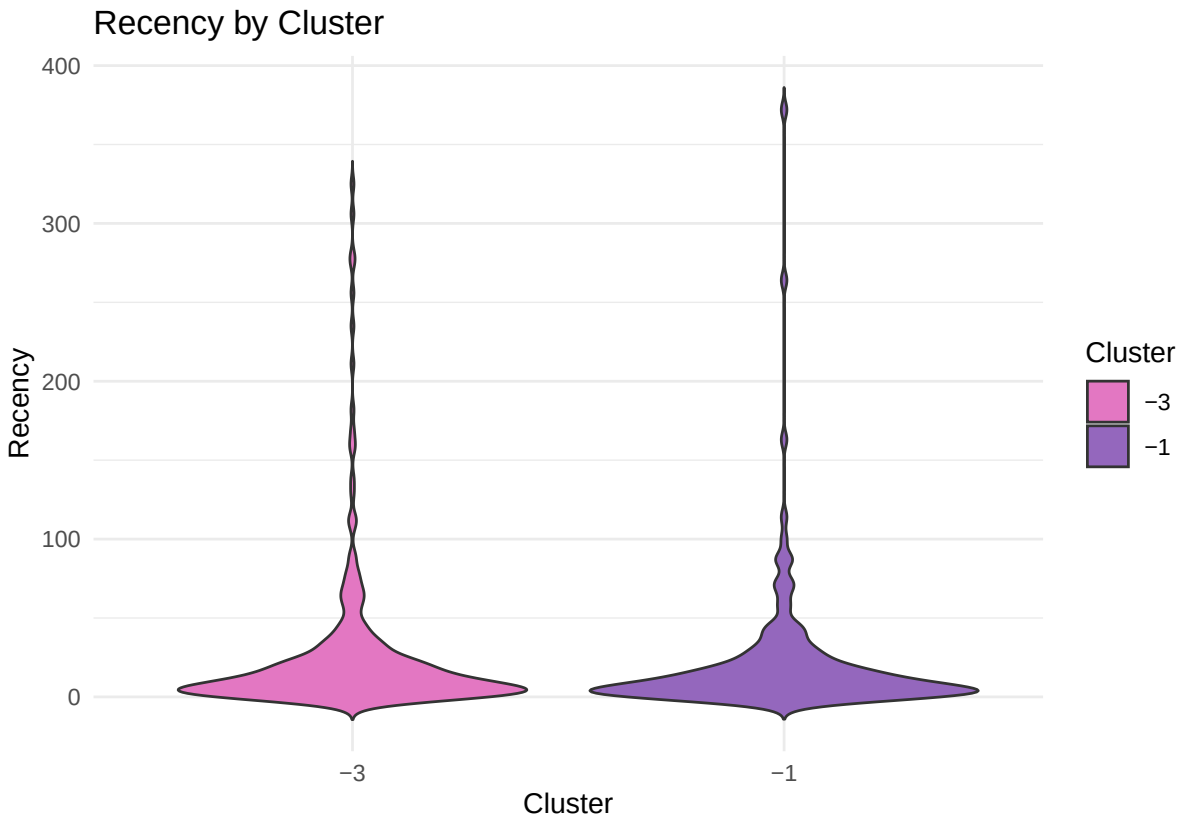
```r
# Chuyển Cluster thành factor để dễ gán màu
outlier_clusters_df$Cluster <- factor(outlier_clusters_df$Cluster)

# Tạo bảng màu tương đương cluster_colors
cluster_colors <- c("-1" = "#9467bd", "-2" = "#8c564b", "-3" = "#e377c2")

# Violin plot: MonetaryValue
ggplot(outlier_clusters_df, aes(x = Cluster, y = MonetaryValue, fill = Cluster)) +
  geom_violin(trim = FALSE) +
  scale_fill_manual(values = cluster_colors) +
  labs(title = "Monetary Value by Cluster", y = "Monetary Value") +
  theme_minimal()
```



```r
# Violin plot: Frequency
ggplot(outlier_clusters_df, aes(x = Cluster, y = Frequency, fill = Cluster)) +
  geom_violin(trim = FALSE) +
  scale_fill_manual(values = cluster_colors) +
  labs(title = "Frequency by Cluster", y = "Frequency") +
  theme_minimal()
```




```r
# Violin plot: Recency
ggplot(outlier_clusters_df, aes(x = Cluster, y = Recency, fill = Cluster)) +
  geom_violin(trim = FALSE) +
  scale_fill_manual(values = cluster_colors) +
  labs(title = "Recency by Cluster", y = "Recency") +
  theme_minimal()
```



Nhận xét: Recency của các nhóm Outliers:

- Cả hai cụm đều có giá trị Recency thấp, tức là phần lớn khách hàng trong hai nhóm này mua hàng gần đây, thường xuyên quay lại trong thời gian gần phân tích (Recency ~ 0–50 ngày là phổ biến).
- Cluster -3: Recency phân bố cực thấp và nghiêng về phía 0 -> những khách hàng gần như vừa mới mua hàng, kết hợp với tần suất cao và chi tiêu lớn -> đây là tệp khách hàng siêu trung thành.
- Cluster -1: Cũng có Recency khá thấp, nhưng phân bố dàn rộng hơn một chút -> một số khách tuy chi nhiều nhưng có thể đã không mua trong thời gian dài.

Ý nghĩa:

- Cluster -3: Hoàn hảo cho chương trình tri ân VIP hiện tại.
- Cluster -1: Nên được tiếp cận bằng chiến dịch kêu gọi quay lại như ưu đãi cho đơn hàng tiếp theo, voucher nếu mua lại trong 7 ngày,…

Nhận xét: Frequency của các nhóm Outliers:

- Cluster -3: Phân bố Frequency rất rộng, có nhiều điểm lên tới trên 200 lượt mua, phần lớn tập trung từ 10–50 lần giao dịch.
- Đây là nhóm mua nhiều – rất thường xuyên, kết hợp với giá trị chi tiêu cao -> thể hiện đặc trưng của khách hàng trung thành lâu năm hoặc doanh nghiệp mua sỉ định kỳ.
- Cluster -1: Frequency thấp hơn đáng kể, hầu hết khách hàng chỉ mua khoảng 5–20 lần, rất ít trường hợp vượt qua 50.
- Nhóm này không mua thường xuyên, nhưng mỗi lần mua có giá trị cao -> khả năng là khách mua ít nhưng sản phẩm có giá trị lớn ví dụ: khách doanh nghiệp, quà tặng,…

Ý nghĩa:

- Cluster -3 là nhóm cực kỳ quan trọng để duy trì lâu dài. Có thể áp dụng các chương trình như:
  - Ưu đãi theo cấp bậc (tích điểm)
  - Tặng thêm sản phẩm
  - Ưu tiên hỗ trợ
- Cluster -1 có thể kích hoạt tăng tần suất mua bằng:
  - Gợi ý sản phẩm liên quan
  - Ưu đãi cho đơn tiếp theo trong thời gian ngắn
  - Ưu đãi combo/quà tặng khi mua lại trong 30 ngày

Nhận xét: Monetary của các nhóm Outliers:

- Cả hai cụm đều có Monetary Value rất cao, với một số khách hàng thậm chí chi tiêu lên tới trên 200.000 đơn vị tiền tệ – đây là những ngoại lệ thực sự trong tập khách hàng.
- Cluster -3:
  - Mật độ phân bố rộng và dày hơn quanh mức từ 0 đến 20.000, cho thấy nhóm này có giá trị đơn hàng cao.
  - Dù có nhiều điểm outlier mạnh, nhưng phần lớn khách vẫn tập trung ở khoảng từ 5.000–10.000.
- Cluster -1:
  - Phân bố hẹp hơn một chút so với cluster -3, nhưng vẫn có một vài khách chi tiêu rất lớn.
  - Đặc trưng là mua ít lần nhưng mỗi lần chi cực nhiều (ví dụ doanh nghiệp,…).

Ý nghĩa:

- Cluster -3: Đây là khách hàng vàng, rất trung thành và có giá trị lâu dài -> xứng đáng nhận các ưu đãi đặc biệt.
- Cluster -1: Dù không thường xuyên mua nhưng mỗi lần mua lại chi tiêu cực lớn -> nên có các chiến lược khuyến khích mua định kỳ, ưu đãi đặc biệt theo mỗi đơn lớn.

Kết luận:

Dựa trên ba biểu đồ phân tích Recency – Frequency – Monetary Value, có thể đưa ra kết luận sau:

- Cluster -3 (có cả Outliers theo Monetary và Frequency cao):
  - Là nhóm khách hàng vừa chi tiêu lớn, vừa mua thường xuyên, và phần lớn đều có Recency thấp (mua gần đây).
  - Có giá trị vượt trội rõ rệt trong cả ba chiều -> nhóm khách hàng VIP, đóng vai trò rất quan trọng với doanh thu.
- Cluster -1 (chỉ outlier theo Monetary):
  - Nhóm khách hàng mua không nhiều lần, nhưng mỗi lần đều chi rất mạnh tay.
  - Recency có thể không quá thấp -> một số khách đã lâu không quay lại, cần chiến lược phù hợp.
  - Phù hợp với các chương trình ưu đãi đơn hàng lớn, tặng voucher cho lần mua tiếp theo, hoặc chăm sóc cá nhân hoá theo giá trị giao dịch.

```r
# Tạo vector gán nhãn cluster
cluster_labels <- c(
  "0" = "RETAIN",        # Giữ chân khách hàng hiện tại
  "1" = "RE-ENGAGE",     # Tương tác lại với khách hàng đã rời đi
  "2" = "NURTURE",       # Chăm sóc khách hàng tiềm năng
  "3" = "REWARD",        # Thưởng cho khách hàng trung thành
  "4" = "EXPLORE",       # Nhóm khách hàng chưa có hành vi nổi bật rõ ràng
  "-1" = "PAMPER",       # Chiều chuộng khách hàng VIP
  "-2" = "UPSELL",       # Đề xuất mua thêm
  "-3" = "DELIGHT"       # Trải nghiệm xuất sắc
)
```

```r
# Gộp 2 dataframe lại theo hàng (giống pd.concat)
full_clustering_df <- rbind(non_outliers_df, outlier_clusters_df)
full_clustering_df$ClusterLabel <- cluster_labels[as.character(full_clustering_df$Cluster)]
# Xem kết quả
head(full_clustering_df)
```

```
## # A tibble: 6 x 7
##   CustomerID MonetaryValue Frequency LastInvoiceDate Recency Cluster
##        <dbl>         <dbl>     <int> <date>            <dbl> <fct>
## 1      12348         1437.         4 2011-09-25           75 3
## 2      12349         1458.         1 2011-11-21           18 2
## 3      12350          294.         1 2011-02-02          310 4
## 4      12352         1386.         7 2011-11-03           36 1
## 5      12353           89          1 2011-05-19          204 4
## 6      12354         1079.         1 2011-04-21          232 4
## # i 1 more variable: ClusterLabel <chr>
```

```r
# Kiểm tra số hàng
nrow(full_clustering_df)
```

```
## [1] 4288
```

```r
# Kiểm tra các nhóm trong Cluster
table(full_clustering_df$Cluster)
```

```
##
##    1    2    3    4   -3   -1
##  482 1547  890  944  278  147
```

```r
# Kiểm tra nếu có cột ClusterLabel đã gán
table(full_clustering_df$ClusterLabel)
```

```
##
##   DELIGHT   EXPLORE   NURTURE    PAMPER RE-ENGAGE    REWARD
##       278       944      1547       147       482       890
```

```r
# Gán nhãn cụm bằng named vector cluster_labels
full_clustering_df$ClusterLabel <- cluster_labels[as.character(full_clustering_df$Cluster)]

# Xem ngẫu nhiên 10 dòng
full_clustering_df[sample(nrow(full_clustering_df), 10), ]
```

```
## # A tibble: 10 x 7
##    CustomerID MonetaryValue Frequency LastInvoiceDate Recency Cluster
##         <dbl>         <dbl>     <int> <date>            <dbl> <fct>
##  1      15950         1766.         4 2011-11-08           31 3
##  2      12574          182.         1 2011-01-28          315 4
##  3      12504          428.         2 2011-11-21           18 2
##  4      15952          807.         4 2011-11-06           33 3
##  5      16110         1337.         3 2011-10-25           45 3
##  6      12489          299.         1 2011-01-07          336 4
##  7      15938          405.         5 2011-08-19          112 3
##  8      14400          575.         1 2011-07-13          149 2
##  9      16796          306.         1 2011-09-11           89 2
## 10      15246          514.         2 2011-04-15          238 4
## # i 1 more variable: ClusterLabel <chr>
```

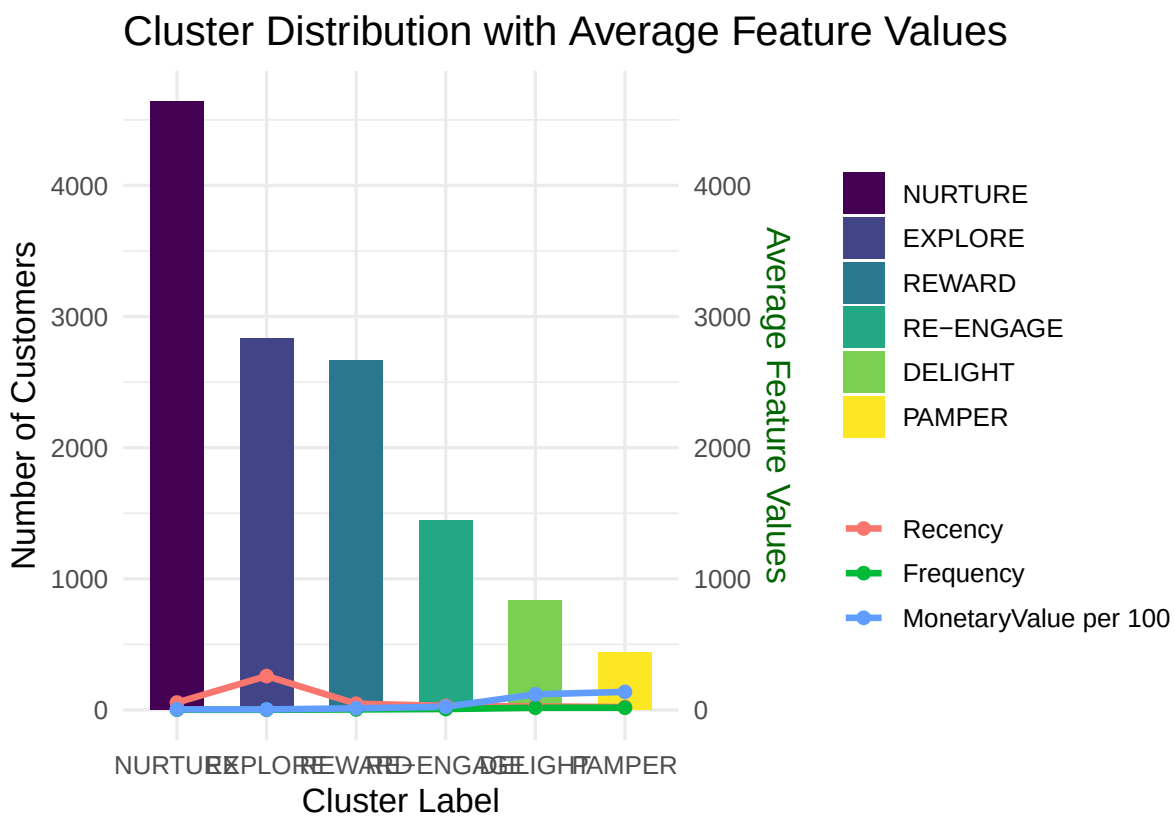
```r
# Tính số lượng khách mỗi cụm
cluster_counts <- full_clustering_df %>%
  count(ClusterLabel, name = "CustomerCount")
```

```r
# Tính trung bình các biến
feature_means <- full_clustering_df %>%
  mutate(MonetaryValuePer100 = MonetaryValue / 100) %>%
  group_by(ClusterLabel) %>%
  summarise(
    Recency = mean(Recency, na.rm = TRUE),
    Frequency = mean(Frequency, na.rm = TRUE),
    `MonetaryValue per 100` = mean(MonetaryValuePer100, na.rm = TRUE)
  ) %>%
  pivot_longer(cols = c("Recency", "Frequency", "MonetaryValue per 100"),
               names_to = "Feature", values_to = "MeanValue")
```

```r
# Merge lại và sắp xếp giảm dần theo CustomerCount
plot_df <- left_join(feature_means, cluster_counts, by = "ClusterLabel") %>%
  mutate(ClusterLabel = factor(ClusterLabel, levels = cluster_counts %>%
                                 arrange(desc(CustomerCount)) %>%
                                 pull(ClusterLabel)))
```

```r
# Vẽ biểu đồ kết hợp
library(viridis)
plot_df$Feature <- factor(plot_df$Feature, levels = c("Recency", "Frequency", "MonetaryValue per 100

ggplot(plot_df, aes(x = ClusterLabel)) +
  # Biểu đồ cột số lượng KH với viridis
  geom_col(aes(y = CustomerCount, fill = ClusterLabel), width = 0.6) +
  scale_fill_viridis(discrete = TRUE, option = "D") +
  scale_y_continuous(
    name = "Number of Customers",
    sec.axis = sec_axis(~., name = "Average Feature Values")
  ) +
  # Biểu đồ đường trung bình các feature
  geom_line(aes(y = MeanValue, color = Feature, group = Feature), size = 1.2) +
  geom_point(aes(y = MeanValue, color = Feature), size = 2) +
  labs(
    title = "Cluster Distribution with Average Feature Values",
    x = "Cluster Label"
  ) +
  theme_minimal(base_size = 13) +
```

```
theme(
  axis.title.y.left = element_text(color = "black"),
  axis.title.y.right = element_text(color = "darkgreen"),
  legend.title = element_blank()
)
```



Nhận xét: Nhóm NURTURE:

- Số lượng nhiều nhất (~5000), Recency cao, Frequency thấp, Monetary thấp

Ý nghĩa:

- Nhóm khách lâu không quay lại, ít mua và chi ít → cần chiến dịch khuyến mãi quay lại, kích hoạt lại tệp ngủ quên.

Nhận xét: Nhóm EXPLORE:

- Recency trung bình, Frequency tăng nhẹ, Monetary tăng

Ý nghĩa:

- Nhóm đang có dấu hiệu bắt đầu mua lại, cần được thử nghiệm ưu đãi và khuyến khích để nâng cấp hành vi mua.

Nhận xét: Nhóm REWARD:

- Recency thấp, Frequency cao, Monetary tăng rõ

Ý nghĩa:

- Nhóm có hành vi tốt dần lên -> có thể tặng thêm voucher, ưu đãi,…

Nhận xét: Nhóm RE-ENGAGE:

- Recency cao, nhưng Frequency và Monetary trước đó cao

Ý nghĩa:

- Nhóm từng tốt, nay đã rời bỏ -> đặc biệt quan trọng để thu hút trở lại

Nhận xét: Nhóm DELIGHT:

- Recency thấp, Frequency ổn, Monetary cao

Ý nghĩa:

- Nhóm đang có trải nghiệm rất tốt, có thể biến thành khách trung thành VIP nếu có chiến lược phù hợp.

Nhận xét: Nhóm PAMPER :

- Recency thấp nhất, Frequency rất thấp, Monetary cao nhất

Ý nghĩa:

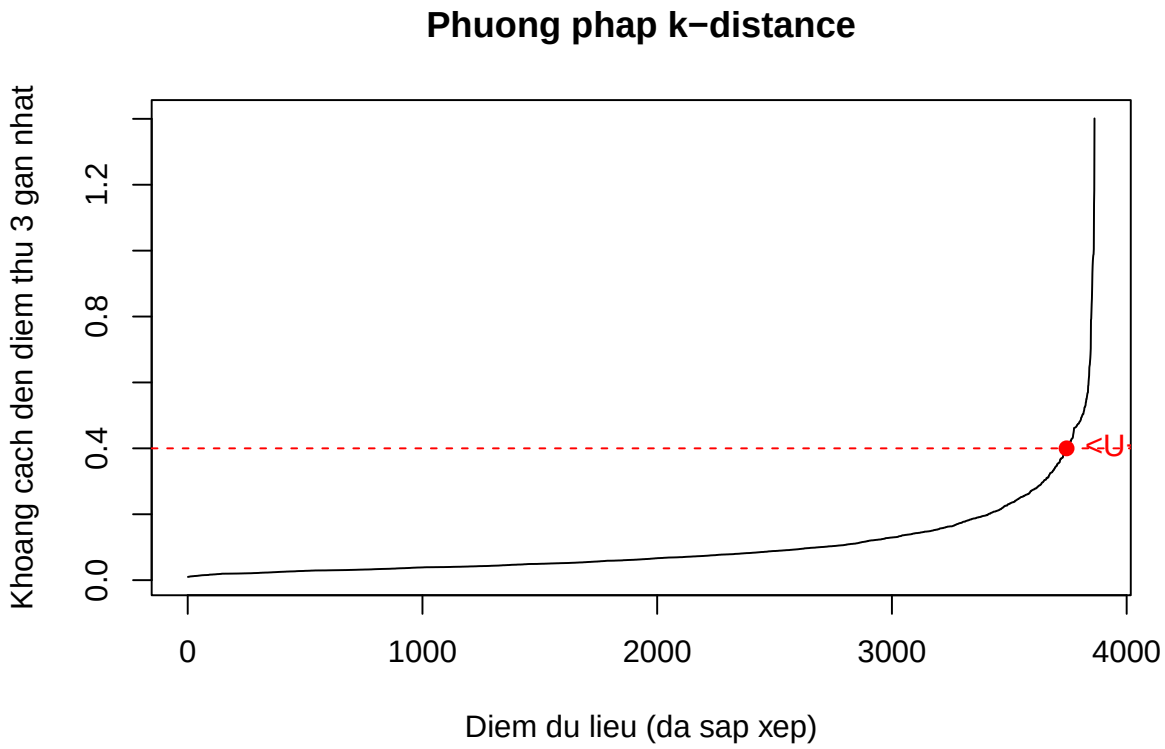
- Nhóm ít mua nhưng chi rất mạnh tay mỗi lần mua -> cần dịch vụ chăm sóc cá nhân hóa, quà tặng đặc biệt,…

3.5 Phân cụm khách hàng bằng thuật toán DBSCAN

Thuật toán **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** là một phương pháp phân cụm dựa trên mật độ điểm dữ liệu, có khả năng xác định các cụm có hình dạng bất kỳ và xử lý tốt các điểm nhiễu (outliers). Đây là một phương pháp phù hợp khi dữ liệu có phân bố không đồng đều và không thể xác định số lượng cụm từ trước

```
# Cài đặt và nạp các gói cần thiết
library(dbscan)  # Cho thuật toán DBSCAN
library(factoextra)  # Cho trực quan hóa
library(fpc)  # Cho đánh giá phân cụm
library(ggplot2)  # Cho vẽ biểu đồ
library(cluster)  # Cho hàm silhouette


scaled_data_DB <- scale(non_outliers_df[, c("MonetaryValue", "Frequency", "Recency")])

# Tính khoảng cách k-nearest neighbors
k=3
knn_dists <- kNNdist(scaled_data_DB, k = k)

# Sắp xếp khoảng cách và vẽ đồ thị
eps_candidates <- sort(knn_dists)
plot(eps_candidates, type = "l",
     xlab = "Diem du lieu (da sap xep)",
     ylab = paste("Khoang cach den diem thu", k, "gan nhat"),
     main = "Phuong phap k-distance")
```

```r
# Tìm vị trí của điểm gẩy thủ công (khoảng 0.5)
eps_value <- 0.4
eps_index <- which.min(abs(eps_candidates - eps_value))

# Đánh dấu điểm gẩy
points(eps_index, eps_candidates[eps_index], col = "red", pch = 19)
text(eps_index, eps_candidates[eps_index],
     labels = paste("  ", round(eps_candidates[eps_index], 2)),
     pos = 4, col = "red")

# Vẽ đường thẳng tại điểm gẩy (thủ công ở mức 0.5)
abline(h = eps_value, col = "red", lty = 2)
```



Hình 17: Đồ thị xác định Eps xử dụng phương pháp k-distance graph

Nhận xét:

- Biểu đồ thể hiện khoảng cách đến điểm thứ k gần nhất (ở đây là k = 3) đã được **sắp xếp tăng dần.**
- **Hình dạng biểu đồ có "góc gãy" rõ ràng**, xuất hiện khoảng sau điểm thứ 3000 trên trục x.
- **Mức ngưỡng epsilon = 0.4** được chọn khá hợp lý, vì nằm gần vị trí bắt đầu có sự tăng vọt về khoảng cách **đây là điểm gãy** giúp phân tách rõ ràng giữa vùng lõi (dense region) và vùng nhiễu (noise).
- **Đường ngang màu đỏ** đánh dấu ngưỡng eps = 0.4, hỗ trợ xác định số lượng hàng xóm cần thiết để mở rộng cụm.

```r
# Áp dụng DBSCAN với tham số phù hợp
db_result <- dbscan(scaled_data_DB, eps = 0.4, MinPts = 4)
non_noise <- which(db_result$cluster > 0)
```

```r
# Nếu số lượng cụm hợp lệ > 1 thì mới tính Silhouette
if (length(unique(db_result$cluster[non_noise])) > 1) {

  # Tính silhouette
  sil <- silhouette(db_result$cluster[non_noise], dist(scaled_data_DB[non_noise, ]))

  # Tính điểm silhouette trung bình toàn bộ
  avg_sil <- mean(sil[, 3], na.rm = TRUE)
  cat("Điem Silhouette trung binh:", avg_sil, "\n")

  # Tạo dataframe từ silhouette object
  sil_df <- as.data.frame(sil)

  # Tính Silhouette trung bình theo từng cụm
  avg_sil_by_cluster <- sil_df %>%
    group_by(cluster) %>%
    summarise(avg_sil_width = mean(sil_width))

  # Vẽ biểu đồ cột
  dev.new(width = 10, height = 8)
  ggplot(avg_sil_by_cluster, aes(x = factor(cluster), y = avg_sil_width)) +
    geom_col(fill = "steelblue") +
    labs(title = "Silhouette Score theo tung cum",
         x = "Cum",
         y = "Silhouette Score trong tung cum") +
    theme_minimal()

} else {
  cat("Khong du so cum de tinh Silhouette Score.\n")
}
```

```
## Điem Silhouette trung binh: 0.04960207
```

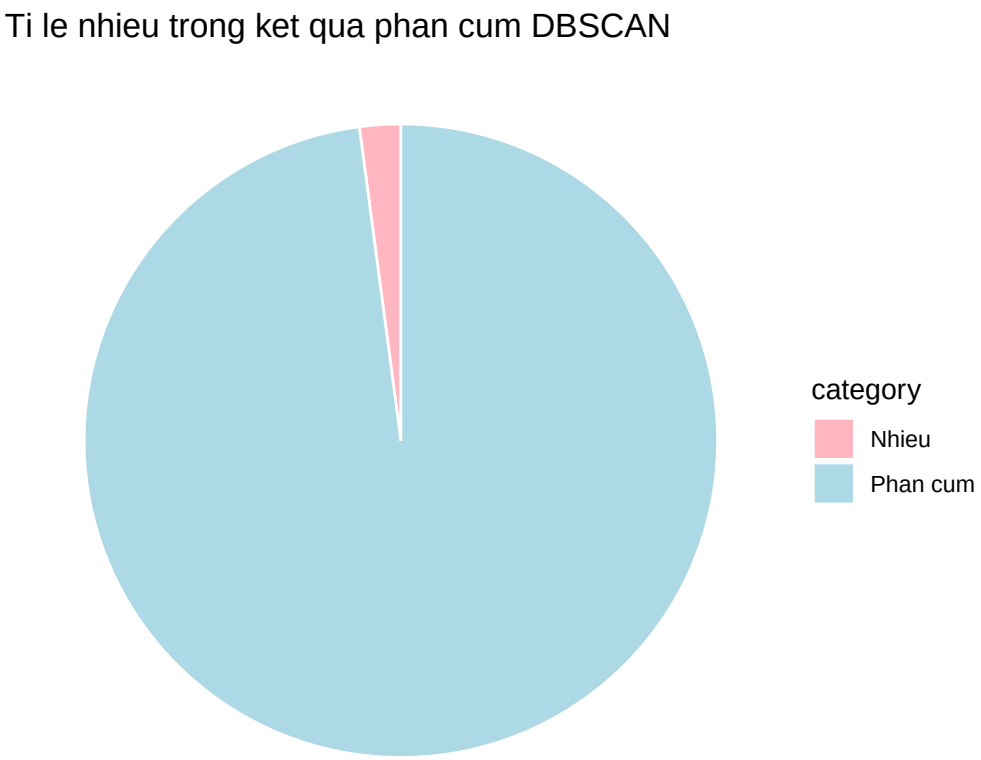
Nhận xét:

- **Giá trị Silhouette trung bình thấp (~0.05)** cho thấy cấu trúc cụm không rõ ràng. Các khách hàng trong cùng một cụm chưa thực sự tương đồng, và ranh giới giữa các cụm chưa tách biệt tốt.
- Một số cụm như **cụm số 6 và 8** có **Silhouette âm**, điều này cho thấy các điểm trong những cụm này có thể bị gán nhầm – chúng gần cụm khác hơn cụm mà chúng đang nằm trong.
- Ngược lại, cụm số 11 có Silhouette khá cao (~0.24), cho thấy đây là một cụm khá "chặt chẽ", có cấu trúc rõ ràng và được tách biệt tốt khỏi các cụm còn lại.

```r
# Tính tỷ lệ nhiễu
noise_ratio <- sum(db_result$cluster == 0) / length(db_result$cluster)
cat("Ty le nhieu:", noise_ratio, "\n")
```

```
## Ty le nhieu: 0.02096816
```

- **Giá trị tỷ lệ nhiễu** là: 0.02096816 (tương đương khoảng 2.10%).
- Điều này có nghĩa là chỉ khoảng 2% dữ liệu bị DBSCAN gán là nhiễu (outlier) tức là không thuộc bất kỳ cụm nào.



Nhận xét biểu đồ tròn: Tỷ lệ nhiễu trong kết quả phân cụm DBSCAN

- Biểu đồ tròn trên thể hiện sự phân chia giữa:
  - Điểm nhiễu (Nhiễu - màu hồng nhạt)
  - Điểm thuộc cụm (Phân cụm - màu xanh nhạt)

Nhận xét:

- **Tỷ lệ nhiễu cực thấp**, chiếm một phần rất nhỏ trong tổng thể dữ liệu.
- Hầu hết các điểm dữ liệu đều được **DBSCAN gán vào một cụm cụ thể**, chứng tỏ:
  - Mô hình đã **phân cụm tốt**, không bị bỏ sót dữ liệu đáng kể.
  - Các thông số `eps = 0.4` và `MinPts = 4` là phù hợp với phân bố dữ liệu hiện tại.

```
# Tính Calinski-Harabasz Index (chỉ cho các điểm không phải nhiễu)
if (length(unique(db_result$cluster[non_noise])) > 1) {
  ch_index <- calinhara(scaled_data[non_noise, ],
                        db_result$cluster[non_noise])
  cat("Calinski-Harabasz Index:", ch_index, "\n")
} else {
  cat("Khong the tinh CH Index: chi tim thay 1 cum (khong tinh nhieu)\n")
}
```

```
## Calinski-Harabasz Index: 593.7985
```

Nhận xét chỉ số Calinski-Harabasz (CH Index) trong DBSCAN

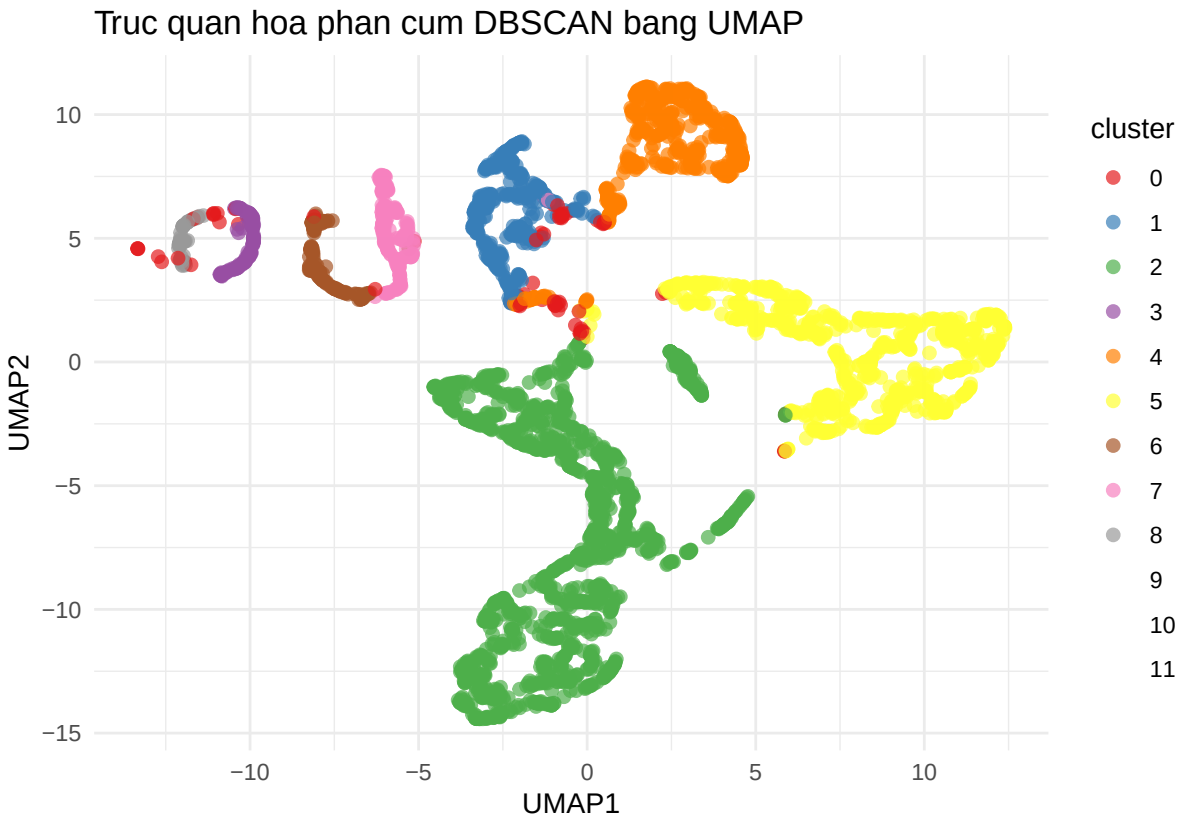
- **Giá trị Calinski-Harabasz Index** đo được là 593.7985.

- Chỉ số này được tính **trên các điểm không bị đánh dấu là nhiễu** trong kết quả DBSCAN.

Ý nghĩa của CH Index:

- **CH Index càng cao → các cụm càng rõ ràng và tách biệt tốt** với nhau.
- Với giá trị ~593.8, đây là một **chỉ số khá cao**, cho thấy:
  - **DBSCAN đã phân cụm hiệu quả,** các cụm được tạo ra có sự phân tách tốt.
  - **Độ chặt trong cụm cao và khoảng cách giữa các cụm lớn** điều này rất tốt trong bối cảnh phân tích hành vi khách hàng.

```
library(umap)
# Giảm chiều bằng UMAP
set.seed(42)  # Giữ kết quả ổn định
umap_result <- umap(scaled_data_DB)

# Tạo dataframe 2D từ kết quả UMAP
umap_df <- as.data.frame(umap_result$layout)
colnames(umap_df) <- c("UMAP1", "UMAP2")

# Gán nhãn cụm từ DBSCAN (nếu có)
umap_df$cluster <- factor(db_result$cluster)  # Bạn đã chạy db_result ở bước trước rồi

# Vẽ biểu đồ trực quan cụm
ggplot(umap_df, aes(x = UMAP1, y = UMAP2, color = cluster)) +
  geom_point(alpha = 0.7, size = 2) +
  labs(title = "Truc quan hoa phan cum DBSCAN bang UMAP",
       x = "UMAP1", y = "UMAP2") +
  scale_color_brewer(palette = "Set1") +
  theme_minimal()
```



Nhận xét biểu đồ UMAP trực quan hóa phân cụm DBSCAN

- DBSCAN đã phân chia tập dữ liệu thành **nhiều cụm riêng biệt** (từ 0 đến 10).
- Dữ liệu đã được giảm chiều bằng **UMAP** để trực quan hóa trên mặt phẳng 2D.
- Mỗi cụm có màu sắc riêng biệt giúp dễ nhận biế

Nhận xét chi tiết:

- **Phân tách cụm rõ ràng:**
  - Biểu đồ cho thấy các cụm được phân tách khá rõ, không chồng lấn nhau quá nhiều.
  - Điều này khẳng định rằng DBSCAN hoạt động hiệu quả khi được áp dụng lên dữ liệu đã chuẩn hóa và giảm chiều.
- **Cụm nhiễu (Cluster 0):**
  - Nhóm màu đỏ (cluster 0) là các điểm bị coi là nhiễu (không thuộc bất kỳ cụm nào).
  - Số lượng điểm nhiễu nhỏ, phù hợp với tỷ lệ đã tính trước đó (~2%).
- **Cảnh báo palette Set1:**
  - Có thông báo rằng DBSCAN đã tạo ra hơn 9 cụm, nhưng palette = "Set1" trong ggplot2 chỉ hỗ trợ tối đa 9 màu.
  - Do đó có thể một vài cụm bị trùng màu → cần đổi sang palette lớn hơn như "Paired", "Dark2" hoặc scale_color_manual().
- **Cảnh báo loại bỏ điểm:**
  - `geom_point()` đã loại bỏ 124 dòng chứa giá trị thiếu hoặc nằm ngoài vùng hiển thị → có thể do UMAP sinh ra giá trị ngoại lai.
  - Điều này không ảnh hưởng quá nhiều đến chất lượng phân cụm.

3.6 So sánh kết quả phân cụm K-Means và DDBSCAN

Trong mục này, chúng tôi tiến hành so sánh hai thuật toán K-Means và DBSCAN được áp dụng trên cùng tập dữ liệu đã được xử lý và chuẩn hóa từ phân tích RFM.

1. Số lượng cụm

- **K-Means:** yêu cầu chỉ định số cụm trước. Dựa vào phương pháp Elbow và chỉ số Silhouette, số cụm tối ưu được xác định là 4 cụm. không yêu cầu số cụm đầu vào mà dựa trên mật độ điểm và hai tham số eps, minPts. Kết quả phân cụm thu được 11 cụm, trong đó bao gồm cả cụm nhiễu (noise.

2. Đánh giá chất lượng cụm

Chỉ số	K-Means	DBSCAN
Silhouette trung bình	~0.40	~0.05
CH Index	Không tính	593.8
Tỷ lệ nhiễu	Không có	2.1% (0.02096)

- **Silhouette:** K-Means đạt giá trị cao hơn đáng kể, cho thấy các cụm được tạo ra có độ phân tách rõ ràng hơn.
- **CH Index:** DBSCAN cho thấy cụm phân chia khá rõ về mặt không gian, mặc dù Silhouette thấp do có nhiều cụm nhỏ hoặc nhiễu.

- **Tỷ lệ nhiễu:** DBSCAN có thể phát hiện điểm nhiễu (~2.1%), trong khi K-Means không có cơ chế xử lý nhiễu.

3. Đặc điểm cụm và tính linh hoạt

- **K-Means** hoạt động tốt khi dữ liệu phân bố đều, cụm có hình cầu và kích thước tương đương. Tuy nhiên, dễ bị ảnh hưởng bởi outliers và khó xử lý dữ liệu phân bố phức tạp.
- **DBSCAN** phù hợp với dữ liệu có cấu trúc phức tạp, cụm có hình dạng bất kỳ. Đồng thời phát hiện được outlier. Tuy nhiên, việc chọn eps phù hợp khá nhạy cảm và ảnh hưởng lớn đến kết quả.

4. Hiển thị kết quả trực quan Cả hai thuật toán đều được trực quan hóa bằng:

- **Biểu đồ 3D Scatter Plot** với 3 trục: Monetary, Frequency và Recency.
- **Biểu đồ UMAP 2D** để quan sát cấu trúc dữ liệu sau khi giảm chiều.

DBSCAN xuất hiện cảnh báo khi số cụm lớn hơn 9 với palette `Set1`, có thể điều chỉnh bằng palette khác như "Dark2" để trực quan hơn.

5. Tổng kết

Tiêu chí	K-Means	DBSCAN
Dữ liệu rõ ràng, ít nhiễu	Rất phù hợp	Có thể gây nhiễu giả do nhạy cảm với `eps`
Cần phát hiện hành vi bất thường	Không hỗ trợ	Phát hiện nhiễu rất tốt
Cần đặt nhãn cụm rõ ràng cho marketing	Dễ phân loại	Nhãn không đều, khó định nghĩa cụm nhỏ
Dễ triển khai và mô tả	Dễ hiểu, phổ biến	Cần chọn `eps` phù hợp, khó tái hiện
Khả năng mở rộng với dữ liệu lớn	Cao	Giới hạn nếu dữ liệu quá phân mảnh

Khuyến nghị áp dụng

- **Sử dụng K-Means** nếu mục tiêu là phân nhóm khách hàng theo hành vi mua hàng cụ thể để thực hiện các chiến dịch chăm sóc, giữ chân và ưu đãi phù hợp.
- **Sử dụng DBSCAN** nếu bài toán yêu cầu phát hiện khách hàng bất thường, không có hành vi mua rõ ràng hoặc kiểm tra dữ liệu nhiễu.

CHƯƠNG 4: KẾT LUẬN VÀ KIẾN NGHỊ

4.1 Kết luận

Trong báo cáo này, chúng tôi đã tiến hành phân khúc khách hàng dựa trên mô hình RFM (Recency - Frequency - Monetary) kết hợp với các thuật toán phân cụm K-Means và DBSCAN. Sau quá trình tiền xử lý, làm sạch dữ liệu và giảm chiều dữ liệu bằng PCA, các mô hình đã được áp dụng thành công để xác định các nhóm khách hàng có hành vi mua sắm tương đồng.

Kết quả cho thấy:

- **Mô hình K-Means** cho kết quả phân cụm rõ ràng, đặc biệt khi sử dụng số cụm được xác định bởi phương pháp Elbow và đánh giá qua Silhouette Score. Các cụm phân biệt rõ ràng về đặc điểm chi tiêu, tần suất và thời gian mua hàng gần nhất.
- **Mô hình DBSCAN** sau khi áp dụng giảm chiều bằng UMAP, cũng phân cụm được dữ liệu nhưng có xu hướng tạo ra ít cụm hơn và một phần dữ liệu bị xem là nhiễu (noise). Tuy nhiên, DBSCAN có ưu điểm là không cần xác định số cụm từ trước và phát hiện được các điểm ngoại lệ.

Tổng thể, cả hai mô hình đều cho thấy hiệu quả trong việc phân nhóm khách hàng. Mô hình K-Means tỏ ra phù hợp hơn trong trường hợp dữ liệu đã được chuẩn hóa và không chứa quá nhiều nhiễu, trong khi DBSCAN có thể áp dụng tốt với dữ liệu có hình dạng phân bố phức tạp.

4.2 Kiến nghị

Từ kết quả phân cụm, chúng tôi đề xuất một số chiến lược tiếp thị như sau:

- **Khách hàng có R cao, F thấp, M thấp:** Nhóm khách hàng mới hoặc không thường xuyên. Cần được tiếp cận qua các chương trình khuyến mãi hoặc giới thiệu sản phẩm mới để tăng tần suất mua sắm.
- **Khách hàng có F và M cao, R thấp:** Nhóm khách hàng trung thành và có giá trị cao. Nên có các chương trình tri ân, ưu đãi đặc biệt để giữ chân nhóm này.
- **Khách hàng có R cao, F và M thấp:** Nhóm khách hàng đang có dấu hiệu rời bỏ. Doanh nghiệp nên cân nhắc các chương trình kích hoạt lại khách hàng cũ như gửi email nhắc nhở hoặc mã giảm giá quay lại.
- Ngoài ra, các điểm nhiễu được DBSCAN xác định có thể là khách hàng ngoại lệ, nên được xem xét kỹ lưỡng để loại bỏ hoặc phân tích riêng biệt nhằm hiểu rõ hơn hành vi bất thường.

References

[1] N. N. Sơn, "K means là gì? Các công bố khoa học về K means," *Scholar Hub*. Accessed: Apr. 03, 2025. [Online]. Available: https://scholarhub.vn/topic/k means

[2] N. Sharma, "K-Means Clustering Explained," *neptune.ai*. Jul. 2022. Accessed: Apr. 03, 2025. [Online]. Available: https://neptune.ai/blog/k-means-clustering

[3] "Determining The Optimal Number Of Clusters: 3 Must Know Methods," *Datanovia*. Accessed: Apr. 03, 2025. [Online]. Available: https://www.datanovia.com/en/lessons/determining-the-optimal-number-of-clusters-3-must-know-methods/

[4] admin, "K-Means Clustering: Ưu nhược điểm và hướng áp dụng," *AIcandy*. Aug. 2024. Accessed: Apr. 03, 2025. [Online]. Available: https://aicandy.vn/k-means-clustering-uu-nhuoc-diem-va-huong-ap-dung/

[5] N. Sharma, "K-Means Clustering Explained," *neptune.ai*. Jul. 2022. Accessed: Apr. 03, 2025. [Online]. Available: https://neptune.ai/blog/k-means-clustering

[6] "DBSCAN," *Wikipedia tiếng Việt*. Mar. 2021. Accessed: Apr. 03, 2025. [Online]. Available: https://vi.wikipedia.org/w/index.php?title=DBSCAN&oldid=64535681#cite_note-2

[7] D. R. Yehoshua, "DBSCAN: Density-Based Clustering," *Medium*. Nov. 2023. Accessed: Apr. 03, 2025. [Online]. Available: https://ai.plainenglish.io/dbscan-density-based-clustering-aaebd76e2c8c

[8] "DBSCAN Clustering Algorithm Demystified," *Built In*. Accessed: Apr. 03, 2025. [Online]. Available: https://builtin.com/articles/dbscan

[9] "Phân tích RFM là gì và các bước phân khúc khách hàng theo RFM," *Tomorrow Marketers*. Apr. 2023. Accessed: Apr. 14, 2025. [Online]. Available: https://blog.tomorrowmarketers.org/phan-tich-rfm-la-gi/