Analyse de Segmentation Clients avec des Données E-commerce

SIMO KOM Yannick Mandela

2025-03-16

Contents

1 Introduction

Dans ce rapport, nous réalisons une analyse de segmentation clients en utilisant des données de transactions e-commerce. La segmentation client est une technique de marketing qui divise une base de clientèle en groupes de personnes ayant des caractéristiques similaires, permettant aux entreprises de développer des stratégies marketing ciblées.

1.1 Objectifs

- Collecter et prétraiter des données de transactions e-commerce
- Appliquer des algorithmes de clustering pour segmenter les clients
- Visualiser les différents segments de clients
- Interpréter les résultats et proposer des recommandations business

2 Méthodologie

2.1 Collecte de données

Nous utilisons l'ensemble de données "Online Retail" du UCI Machine Learning Repository. Ce jeu de données contient toutes les transactions effectuées par un détaillant en ligne basé au Royaume-Uni entre le 01/12/2010 et le 09/12/2011.

```
# Charger les données
online_retail <- readxl::read_excel("Online_Retail.xlsx")

# Aperçu des données
head(online_retail)
```

```
## # 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>
```

2.2 Prétraitement des données

2.2.1 Nettoyage et transformation

```r
# Structure des données
str(online_retail)
```

```
## 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 CUPI
##  $ 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" .
```

```r
# Résumé statistique
summary(online_retail)
```

```
##   InvoiceNo          StockCode         Description           Quantity
##  Length:541909      Length:541909      Length:541909      Min.   :-80995.00
##  Class :character   Class :character   Class :character   1st Qu.:     1.00
##  Mode  :character   Mode  :character   Mode  :character   Median :     3.00
##                                                           Mean   :     9.55
##                                                           3rd Qu.:    10.00
##                                                           Max.   : 80995.00
##
##   InvoiceDate                    UnitPrice           CustomerID
##  Min.   :2010-12-01 08:26:00   Min.   :-11062.06   Min.   :12346
##  1st Qu.:2011-03-28 11:34:00   1st Qu.:     1.25   1st Qu.:13953
##  Median :2011-07-19 17:17:00   Median :     2.08   Median :15152
##  Mean   :2011-07-04 13:34:57   Mean   :     4.61   Mean   :15288
##  3rd Qu.:2011-10-19 11:27:00   3rd Qu.:     4.13   3rd Qu.:16791
##  Max.   :2011-12-09 12:50:00   Max.   : 38970.00   Max.   :18287
##                                                    NA's   :135080
##    Country
##  Length:541909
##  Class :character
##  Mode  :character
```

```
##
##
##
##
```

```
# Vérification des valeurs manquantes
colSums(is.na(online_retail))
```

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

```
# Supprimer les lignes avec des valeurs manquantes
online_retail_clean <- online_retail %>%
  filter(!is.na(CustomerID)) %>%
  filter(Quantity > 0, UnitPrice > 0)

# Vérifier les doublons
n_distinct(online_retail_clean$InvoiceNo)
```

```
## [1] 18532
```

3 Analyses complémentaires

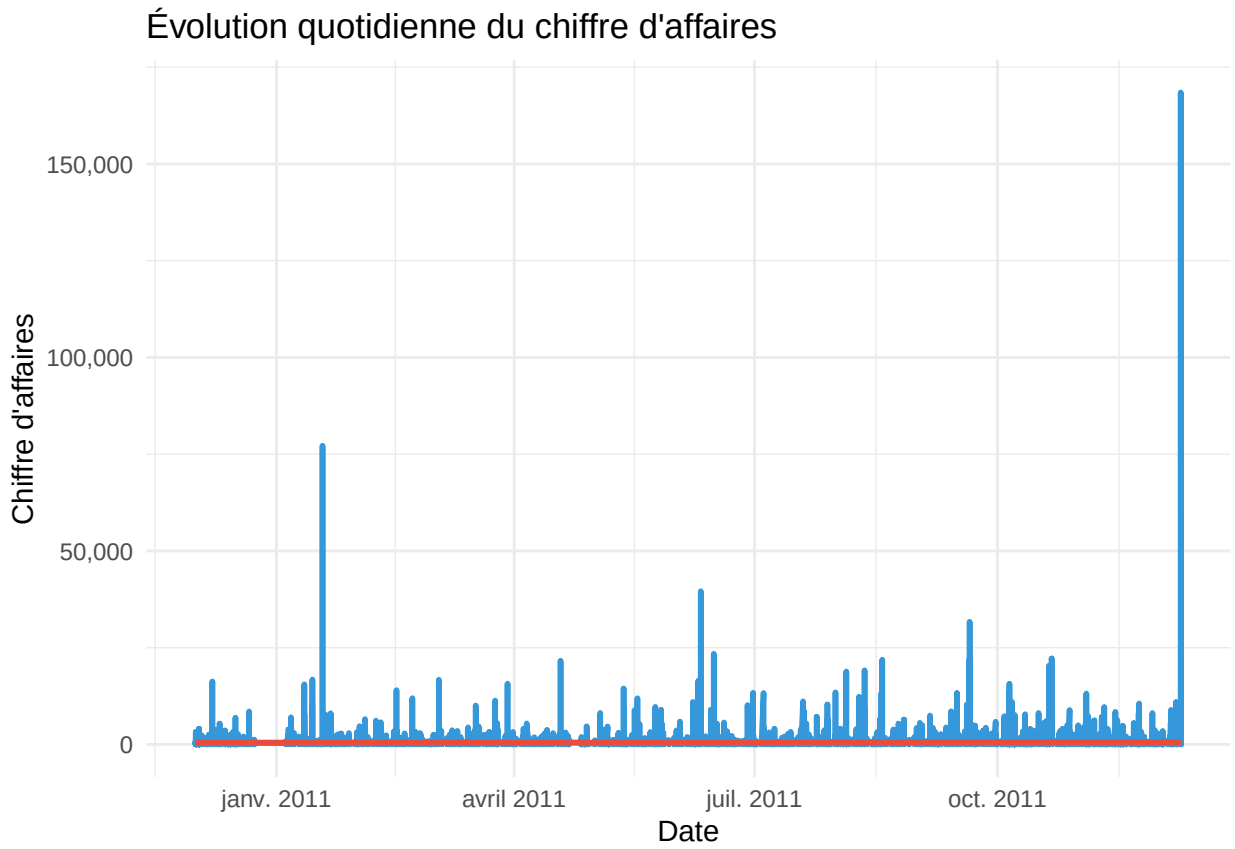
3.1 Analyse des ventes

```
# S'assurer que TotalPrice est bien calculé
if(!"TotalPrice" %in% names(online_retail_clean)) {
  online_retail_clean$TotalPrice <- online_retail_clean$Quantity * online_retail_clean$UnitPrice
}

# Préparation des données pour les analyses de ventes
online_retail_clean$YearMonth <- format(online_retail_clean$InvoiceDate, "%Y-%m")
online_retail_clean$Month <- format(online_retail_clean$InvoiceDate, "%m")
online_retail_clean$Year <- format(online_retail_clean$InvoiceDate, "%Y")
online_retail_clean$WeekDay <- weekdays(online_retail_clean$InvoiceDate)
online_retail_clean$Hour <- format(as.POSIXct(online_retail_clean$InvoiceDate), "%H")

# Convertir les mois en facteur ordonné
online_retail_clean$Month <- factor(online_retail_clean$Month,
                                    levels = c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10
                                    labels = c("Jan", "Fév", "Mar", "Avr", "Mai", "Juin", "Juil", "Août"

# Convertir les jours de la semaine en facteur ordonné
online_retail_clean$WeekDay <- factor(online_retail_clean$WeekDay,
                                      levels = c("lundi", "mardi", "mercredi", "jeudi", "vendredi", "sam

# Ventes par mois
sales_by_month <- online_retail_clean %>%
```

```r
  group_by(Year, Month) %>%
  summarise(TotalSales = sum(TotalPrice)) %>%
  arrange(Year, Month)

# Évolution des ventes au cours du temps
sales_over_time <- online_retail_clean %>%
  group_by(InvoiceDate) %>%
  summarise(TotalSales = sum(TotalPrice)) %>%
  arrange(InvoiceDate)
```

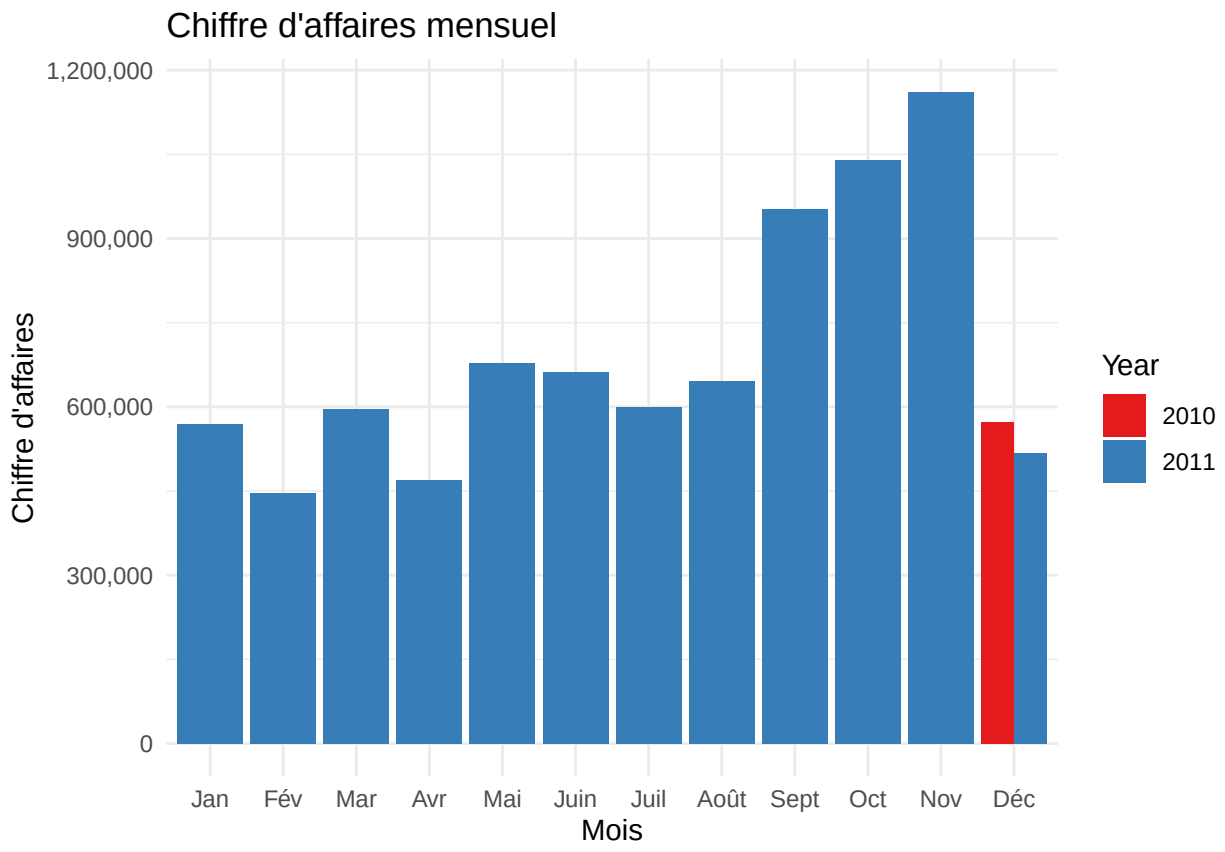
3.1.1 Évolution des ventes au cours du temps

```r
# Graphique d'évolution des ventes
ggplot(sales_over_time, aes(x = InvoiceDate, y = TotalSales)) +
  geom_line(color = "#3498db", size = 1) +
  geom_smooth(method = "loess", color = "#e74c3c", fill = "#e74c3c20", se = TRUE) +
  labs(title = "Évolution quotidienne du chiffre d'affaires",
       x = "Date",
       y = "Chiffre d'affaires") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)
```



3.1.2 Ventes par mois

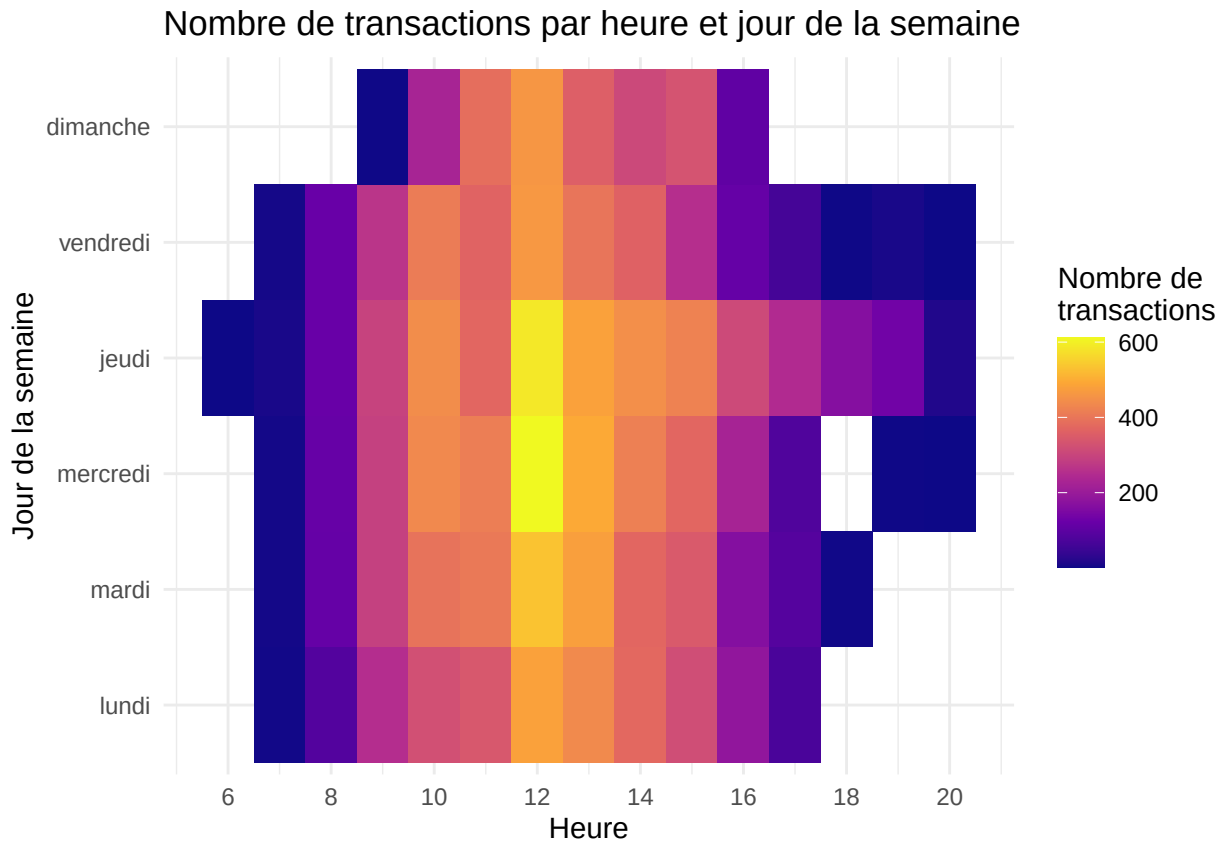
```
# Graphique des ventes mensuelles
ggplot(sales_by_month, aes(x = Month, y = TotalSales, fill = Year)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Chiffre d'affaires mensuel",
       x = "Mois",
       y = "Chiffre d'affaires") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1") +
  scale_y_continuous(labels = scales::comma)
```



3.1.3 Répartition des ventes par jour et heure

```
# Création de données pour la heatmap ventes par jour/heure
hourly_sales <- online_retail_clean %>%
  mutate(Hour = as.numeric(Hour)) %>%
  group_by(WeekDay, Hour) %>%
  summarise(
    TransactionCount = n_distinct(InvoiceNo),
    TotalSales = sum(TotalPrice)
  )
```

```r
# Création de la heatmap des transactions par jour/heure
ggplot(hourly_sales, aes(x = Hour, y = WeekDay, fill = TransactionCount)) +
  geom_tile() +
  scale_fill_viridis_c(option = "plasma") +
  labs(title = "Nombre de transactions par heure et jour de la semaine",
       x = "Heure",
       y = "Jour de la semaine",
       fill = "Nombre de\ntransactions") +
  theme_minimal() +
  scale_x_continuous(breaks = seq(0, 23, 2))
```



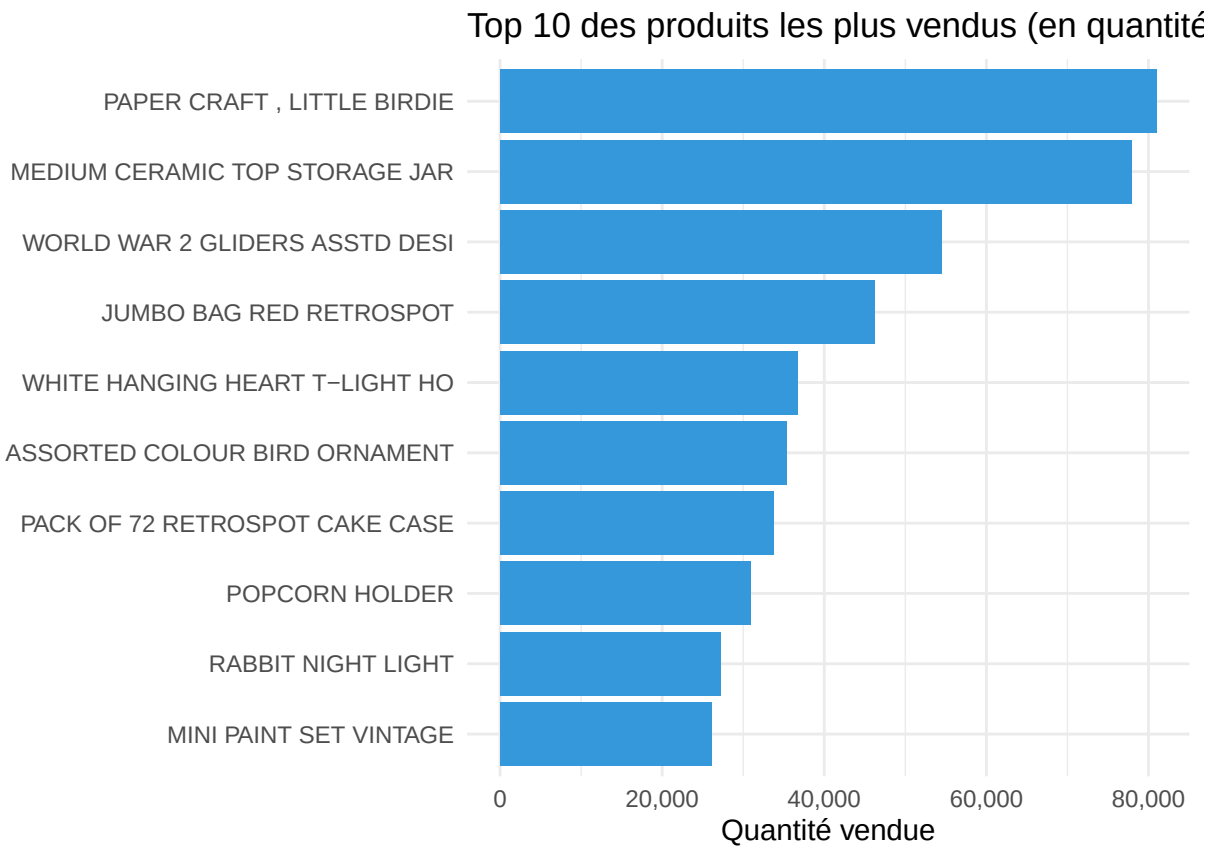
3.2 Analyse des produits

```r
# Préparation des données pour l'analyse des produits
# Top 10 des articles les plus vendus (quantité)
top_items_quantity <- online_retail_clean %>%
  group_by(StockCode, Description) %>%
  summarise(TotalQuantity = sum(Quantity)) %>%
  arrange(desc(TotalQuantity)) %>%
  head(10)

# Top 10 des articles les plus vendus (valeur)
top_items_value <- online_retail_clean %>%
```

```r
  group_by(StockCode, Description) %>%
  summarise(TotalValue = sum(TotalPrice)) %>%
  arrange(desc(TotalValue)) %>%
  head(10)

# Articles les plus achetés (fréquence d'achat)
top_items_frequency <- online_retail_clean %>%
  group_by(StockCode, Description) %>%
  summarise(PurchaseCount = n_distinct(InvoiceNo)) %>%
  arrange(desc(PurchaseCount)) %>%
  head(10)
```

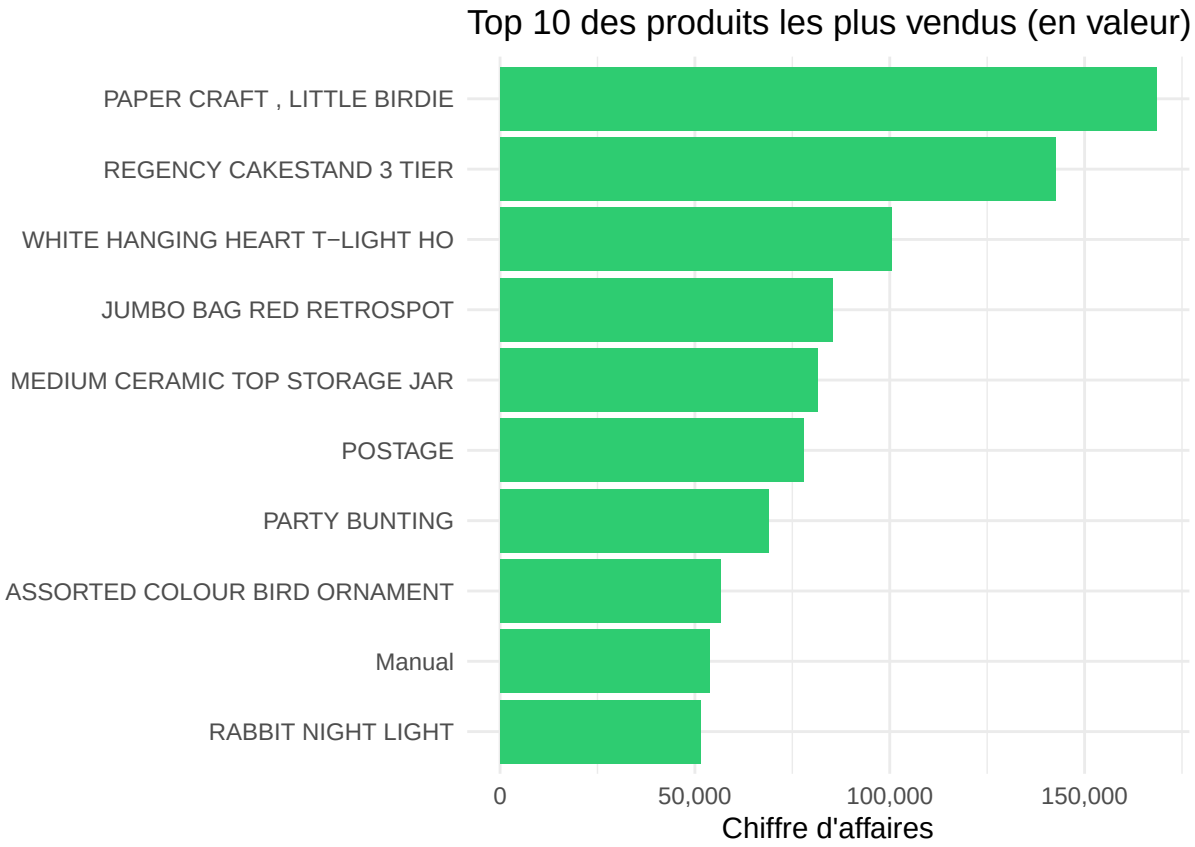
3.2.1 Top 10 des produits les plus vendus (en quantité)

```r
# Graphique des produits les plus vendus (quantité)
ggplot(top_items_quantity, aes(x = reorder(substr(Description, 1, 30), TotalQuantity), y = TotalQuantity
  geom_bar(stat = "identity", fill = "#3498db") +
  coord_flip() +
  labs(title = "Top 10 des produits les plus vendus (en quantité)",
       x = "",
       y = "Quantité vendue") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)
```



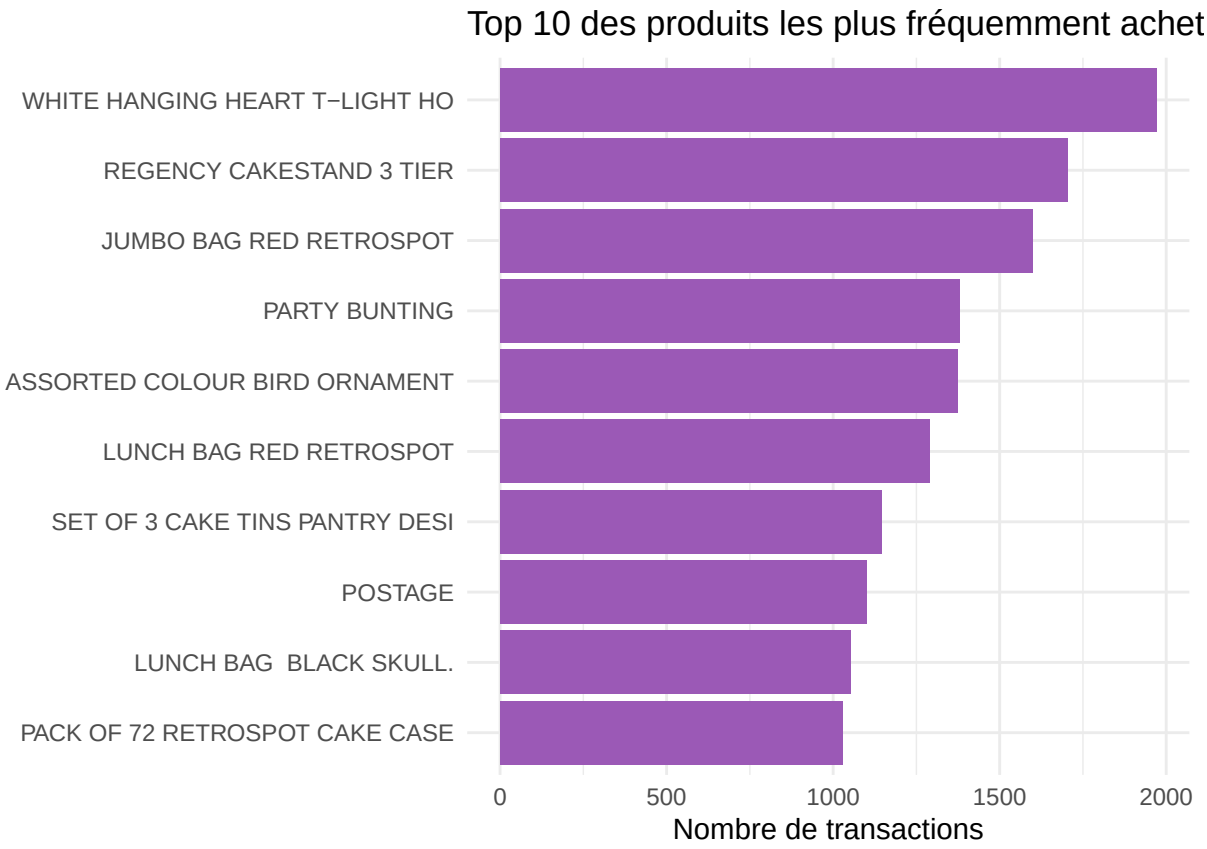
3.2.2 Top 10 des produits les plus vendus (en valeur)

```
# Graphique des produits les plus vendus (valeur)
ggplot(top_items_value, aes(x = reorder(substr(Description, 1, 30), TotalValue), y = TotalValue)) +
  geom_bar(stat = "identity", fill = "#2ecc71") +
  coord_flip() +
  labs(title = "Top 10 des produits les plus vendus (en valeur)",
       x = "",
       y = "Chiffre d'affaires") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)
```



3.2.3 Top 10 des produits les plus fréquemment achetés

```
# Graphique des produits les plus fréquemment achetés
ggplot(top_items_frequency, aes(x = reorder(substr(Description, 1, 30), PurchaseCount), y = PurchaseCou
  geom_bar(stat = "identity", fill = "#9b59b6") +
  coord_flip() +
  labs(title = "Top 10 des produits les plus fréquemment achetés",
       x = "",
       y = "Nombre de transactions") +
  theme_minimal()
```



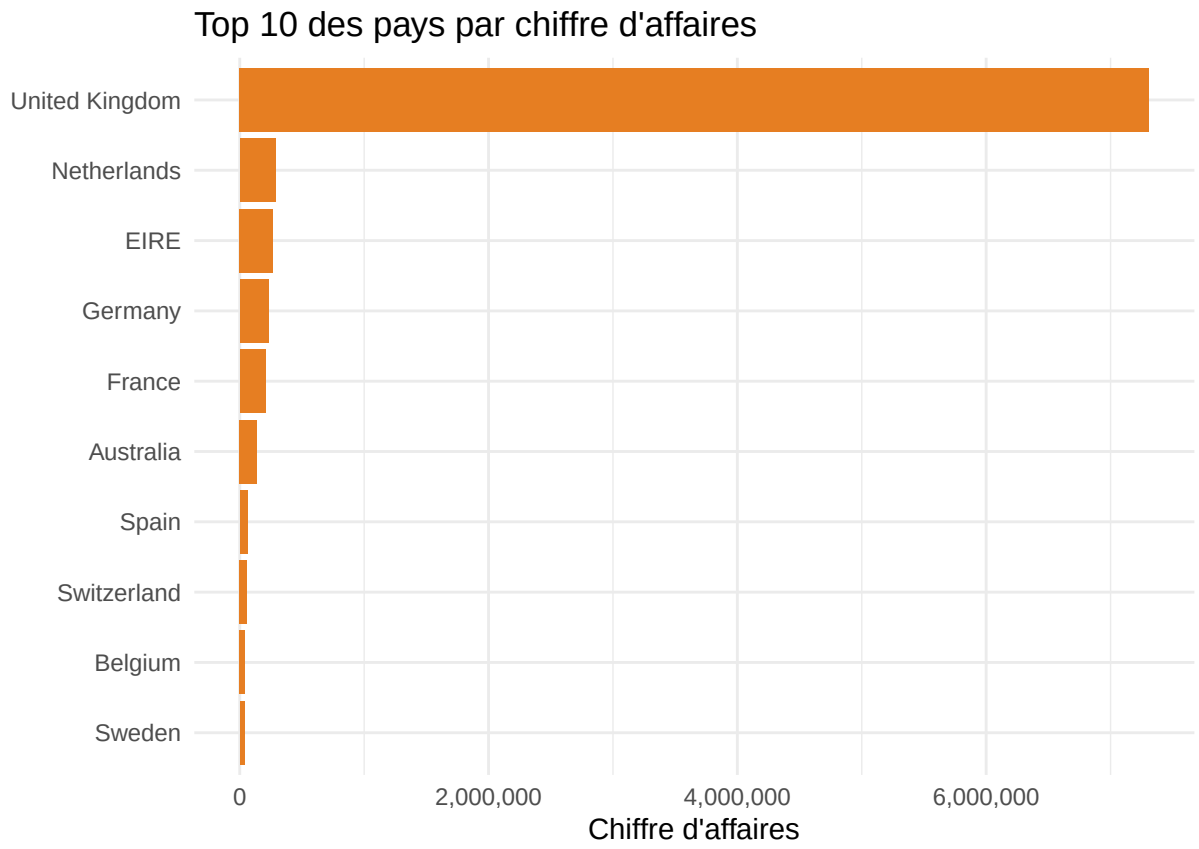
3.3 Analyse géographique

```r
# Préparation des données pour l'analyse géographique
# Top pays en termes de ventes
top_countries <- online_retail_clean %>%
  group_by(Country) %>%
  summarise(
    TotalSales = sum(TotalPrice),
    OrderCount = n_distinct(InvoiceNo),
    CustomerCount = n_distinct(CustomerID)
  ) %>%
  arrange(desc(TotalSales))

# Top 10 des pays
top_10_countries <- top_countries %>% head(10)

# Nombre de clients et ventes moyennes par pays
country_metrics <- top_countries %>%
  mutate(
    AverageSalePerCustomer = TotalSales / CustomerCount,
    AverageSalePerOrder = TotalSales / OrderCount
  ) %>%
  arrange(desc(AverageSalePerCustomer)) %>%
  head(10)
```

```r
# Produit le plus vendu par pays
best_product_by_country <- online_retail_clean %>%
  group_by(Country, StockCode, Description) %>%
  summarise(TotalSales = sum(TotalPrice)) %>%
  arrange(Country, desc(TotalSales)) %>%
  group_by(Country) %>%
  slice(1) %>%
  ungroup() %>%
  arrange(desc(TotalSales))
```

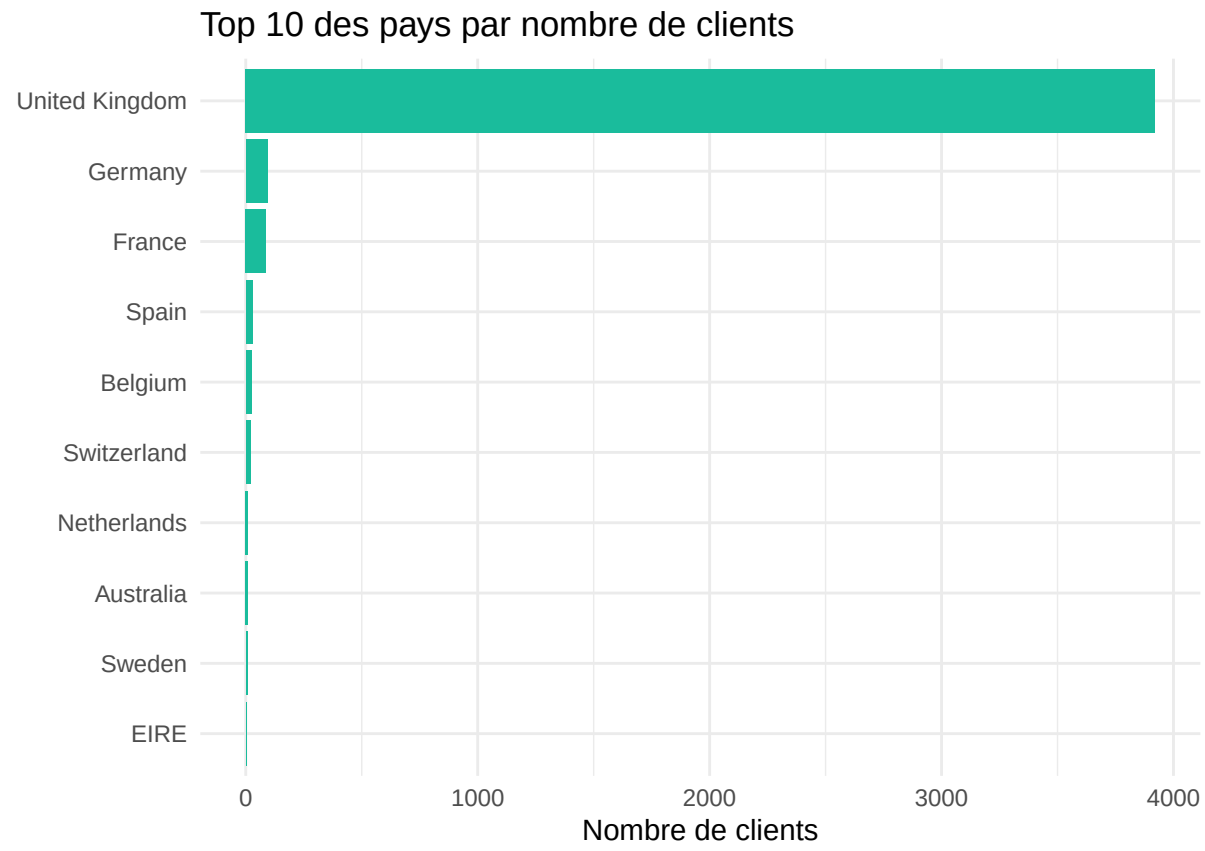
3.3.1 Répartition des ventes par pays (Top 10)

```r
# Graphique des ventes par pays
ggplot(top_10_countries, aes(x = reorder(Country, TotalSales), y = TotalSales)) +
  geom_bar(stat = "identity", fill = "#e67e22") +
  coord_flip() +
  labs(title = "Top 10 des pays par chiffre d'affaires",
       x = "",
       y = "Chiffre d'affaires") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)
```



3.3.2 Nombre de clients par pays (Top 10)

```
# Graphique du nombre de clients par pays
ggplot(top_10_countries, aes(x = reorder(Country, CustomerCount), y = CustomerCount)) +
  geom_bar(stat = "identity", fill = "#1abc9c") +
  coord_flip() +
  labs(title = "Top 10 des pays par nombre de clients",
       x = "",
       y = "Nombre de clients") +
  theme_minimal()
```

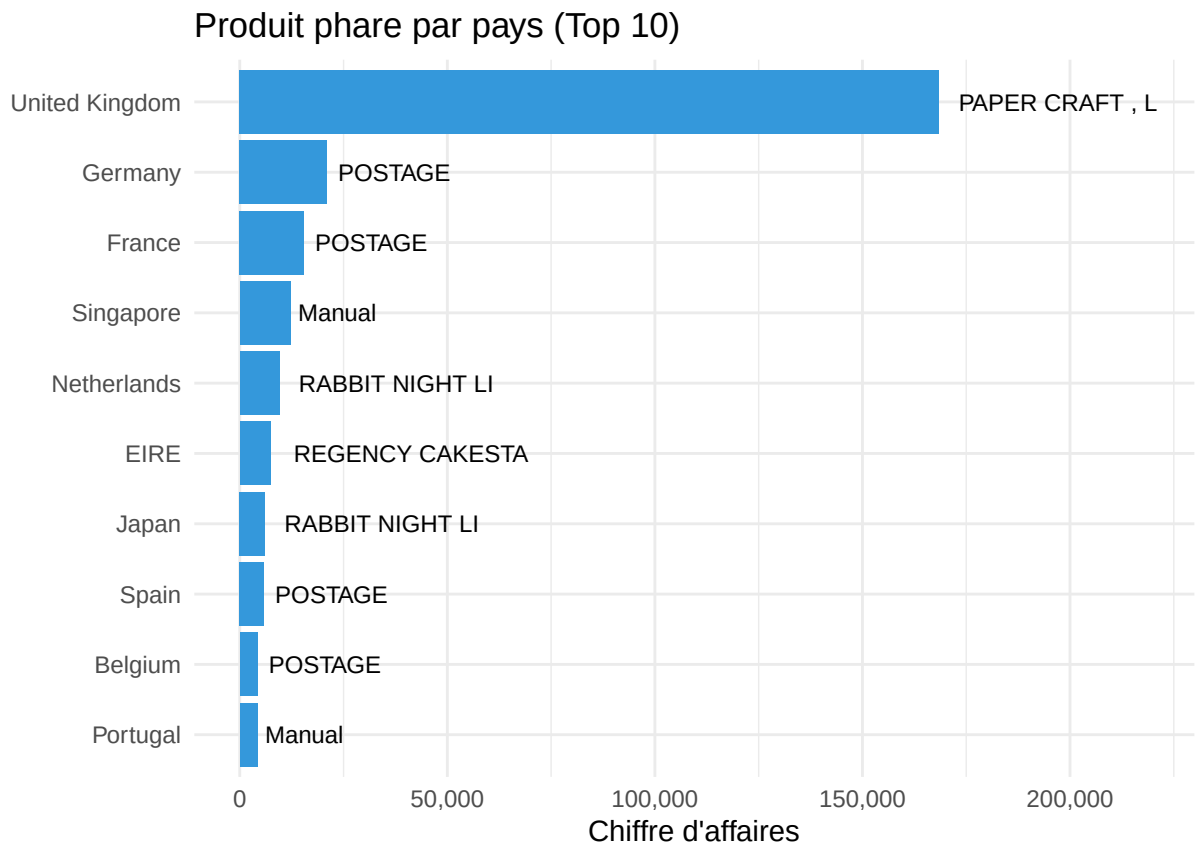


3.3.3 Panier moyen par pays

```
# Graphique du panier moyen par pays
ggplot(country_metrics, aes(x = reorder(Country, AverageSalePerCustomer), y = AverageSalePerCustomer)) +
  geom_bar(stat = "identity", fill = "#f1c40f") +
  coord_flip() +
  labs(title = "Top 10 des pays par panier moyen par client",
       x = "",
       y = "Panier moyen (€)") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma)
```

3.3.4 Produit phare par pays

```
# Graphique des meilleurs produits par pays
top_country_products <- best_product_by_country %>% head(10)

ggplot(top_country_products, aes(x = reorder(Country, TotalSales), y = TotalSales)) +
  geom_bar(stat = "identity", fill = "#3498db") +
  geom_text(aes(label = substr(Description, 1, 15)), hjust = -0.1, size = 3) +
  coord_flip() +
  labs(title = "Produit phare par pays (Top 10)",
       x = "",
       y = "Chiffre d'affaires") +
  theme_minimal() +
  scale_y_continuous(labels = scales::comma, limits = c(0, max(top_country_products$TotalSales) * 1.3))
```



3.3.5 Création de caractéristiques RFM (Recency, Frequency, Monetary)

Le modèle RFM est une méthode d'analyse qui identifie les clients les plus précieux en évaluant: - Récence (R): quand a eu lieu le dernier achat - Fréquence (F): combien d'achats ont été réalisés pendant la période - Montant (M): combien d'argent le client a dépensé pendant la période

```
# Convertir la date en format Date
online_retail_clean$InvoiceDate <- as.Date(online_retail_clean$InvoiceDate)

# Créer une variable pour le montant total
online_retail_clean$TotalPrice <- online_retail_clean$Quantity * online_retail_clean$UnitPrice

# Définir la date de référence (un jour après la dernière date dans les données)
max_date <- max(online_retail_clean$InvoiceDate) + 1

# Calculer les métriques RFM par client
rfm_data <- online_retail_clean %>%
  group_by(CustomerID) %>%
  summarise(
    Recency = as.numeric(max_date - max(InvoiceDate)),
    Frequency = n_distinct(InvoiceNo),
    Monetary = sum(TotalPrice)
  )

# Aperçu des données RFM
```

```r
head(rfm_data)
```

```
## # A tibble: 6 x 4
##   CustomerID Recency Frequency Monetary
##        <dbl>   <dbl>     <int>    <dbl>
## 1      12346     326         1   77184.
## 2      12347       3         7    4310
## 3      12348      76         4    1797.
## 4      12349      19         1    1758.
## 5      12350     311         1     334.
## 6      12352      37         8    2506.
```

```r
summary(rfm_data)
```

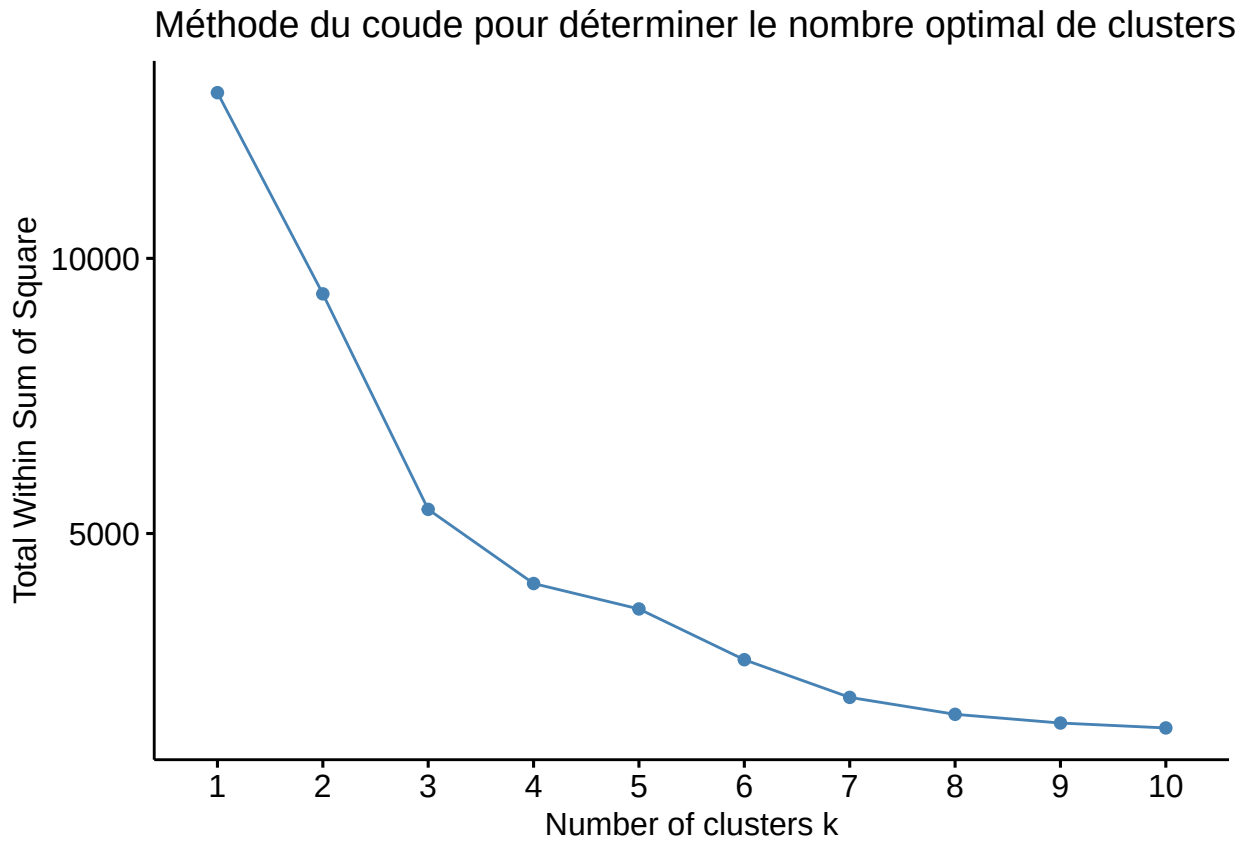
```
##    CustomerID       Recency         Frequency          Monetary
##  Min.   :12346   Min.   :  1.00   Min.   :  1.000   Min.   :     3.75
##  1st Qu.:13813   1st Qu.: 18.00   1st Qu.:  1.000   1st Qu.:   307.42
##  Median :15300   Median : 51.00   Median :  2.000   Median :   674.48
##  Mean   :15300   Mean   : 93.06   Mean   :  4.272   Mean   :  2054.27
##  3rd Qu.:16779   3rd Qu.:142.75   3rd Qu.:  5.000   3rd Qu.:  1661.74
##  Max.   :18287   Max.   :374.00   Max.   :209.000   Max.   :280206.02
```

```r
# Normalisation des variables RFM pour le clustering
rfm_normalized <- rfm_data %>%
  mutate(
    RecencyNorm = scale(Recency),
    FrequencyNorm = scale(Frequency),
    MonetaryNorm = scale(Monetary)
  )
```

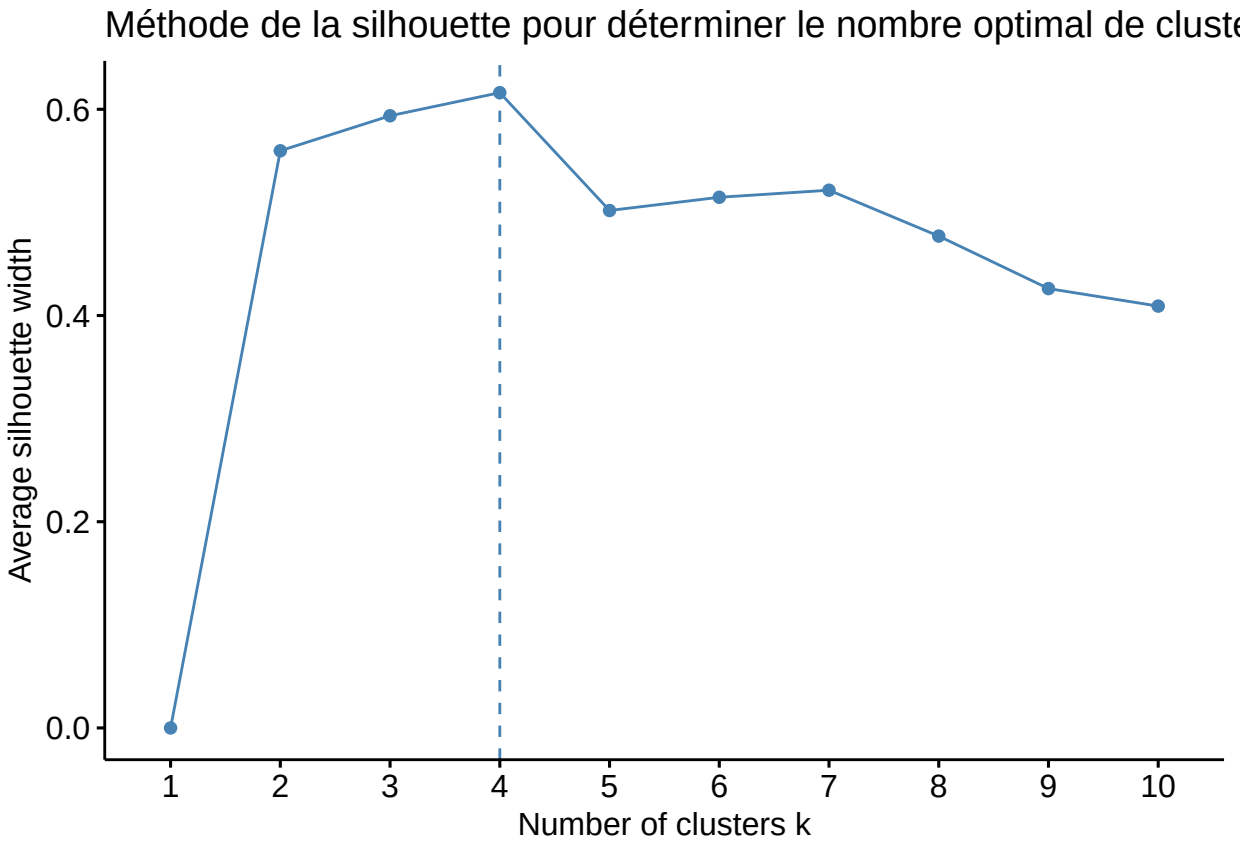
3.4 Segmentation client avec K-means

3.4.1 Détermination du nombre optimal de clusters

```r
# Préparation des données pour le clustering
rfm_for_clustering <- rfm_normalized %>%
  select(RecencyNorm, FrequencyNorm, MonetaryNorm)

# Méthode du coude (Elbow method)
set.seed(123)
wss <- sapply(1:10, function(k) {
  kmeans(rfm_for_clustering, centers = k, nstart = 25)$tot.withinss
})

# Visualisation de la méthode du coude
elbow_plot <- fviz_nbclust(rfm_for_clustering, kmeans, method = "wss") +
  labs(title = "Méthode du coude pour déterminer le nombre optimal de clusters")
elbow_plot
```




```r
# Méthode de la silhouette
silhouette_plot <- fviz_nbclust(rfm_for_clustering, kmeans, method = "silhouette") +
  labs(title = "Méthode de la silhouette pour déterminer le nombre optimal de clusters")
silhouette_plot
```



3.4.2 Application de l'algorithme K-means

Supposons que l'analyse précédente nous a conduit à choisir 4 clusters:

```
# Appliquer K-means avec k=4
set.seed(123)
k <- 4
kmeans_result <- kmeans(rfm_for_clustering, centers = k, nstart = 25)

# Ajouter les clusters aux données RFM
rfm_with_clusters <- rfm_normalized %>%
  mutate(Cluster = as.factor(kmeans_result$cluster))

# Résumé des clusters
cluster_summary <- rfm_with_clusters %>%
  group_by(Cluster) %>%
  summarise(
    N = n(),
    Recency_Mean = mean(Recency),
    Frequency_Mean = mean(Frequency),
    Monetary_Mean = mean(Monetary)
  ) %>%
  arrange(desc(Monetary_Mean))

# Affichage du résumé
```

```r
kable(cluster_summary, caption = "Résumé des clusters") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

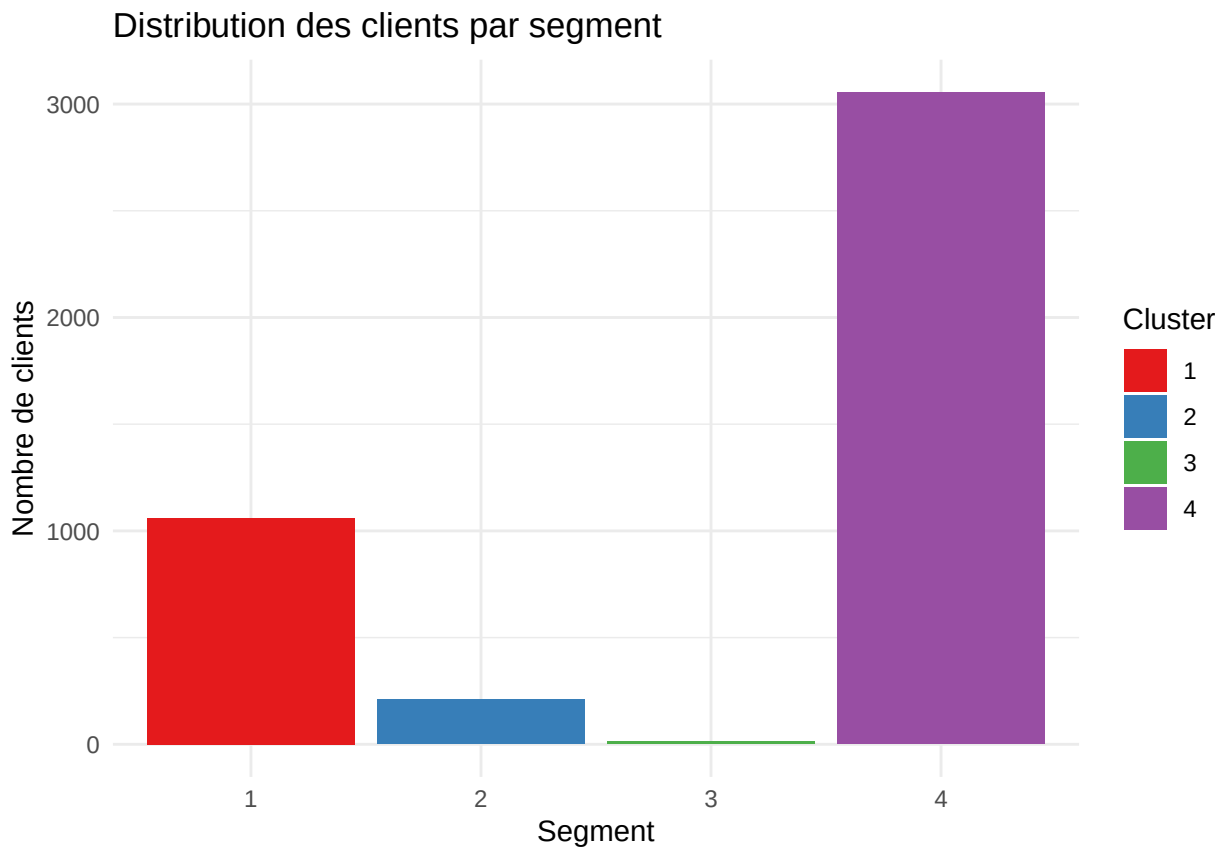
Table 1: Résumé des clusters

Cluster	N	Recency_Mean	Frequency_Mean	Monetary_Mean
3	13	7.615385	82.538461	127338.3138
2	209	16.014354	22.133971	12509.7558
4	3055	44.475614	3.661866	1353.2263
1	1061	249.173421	1.551367	478.1947

4 Visualisations

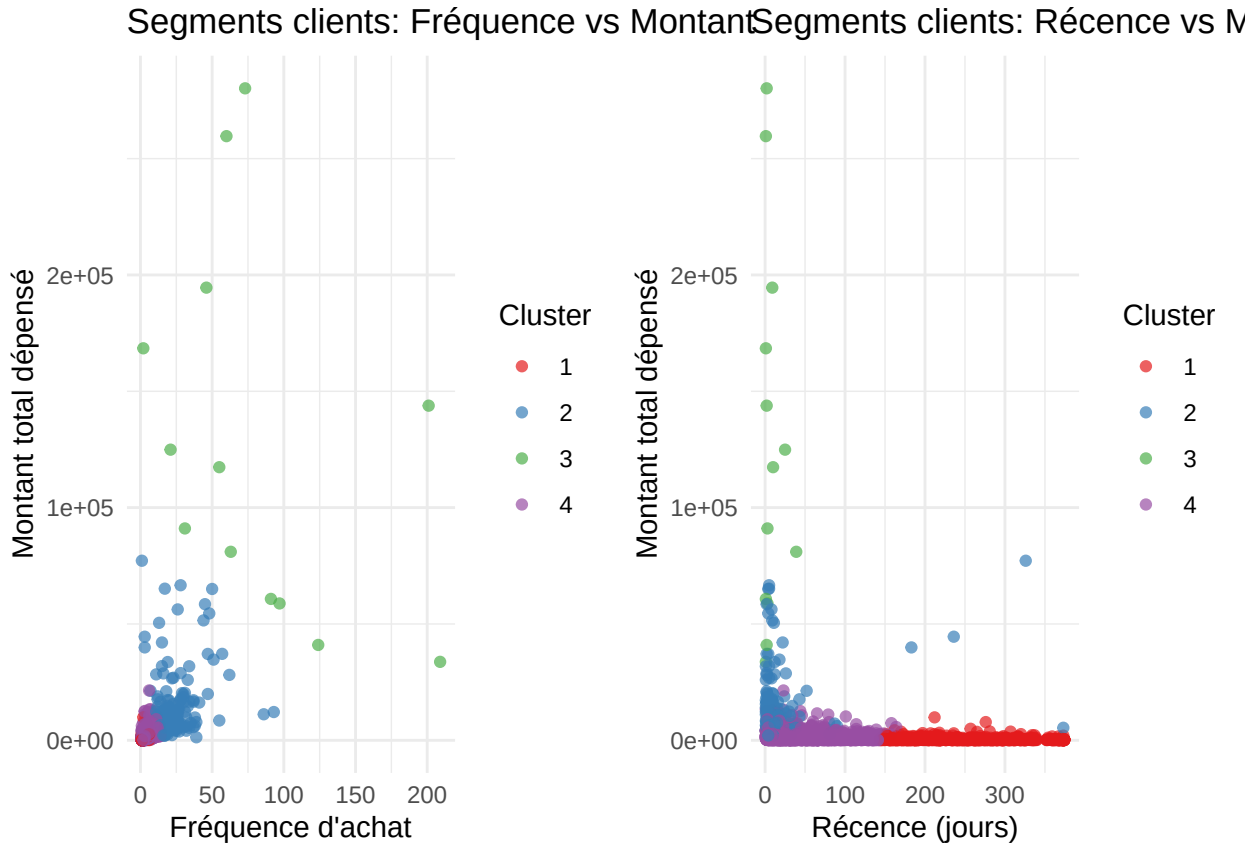
4.1 Distribution des segments clients

```r
# Visualisation de la distribution des clients par cluster
ggplot(rfm_with_clusters, aes(x = Cluster, fill = Cluster)) +
  geom_bar() +
  labs(title = "Distribution des clients par segment",
       x = "Segment",
       y = "Nombre de clients") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")
```



4.2 Représentation des clusters dans l'espace RFM

```r
# Visualisation en 2D (F vs M)
plot_fm <- ggplot(rfm_with_clusters, aes(x = Frequency, y = Monetary, color = Cluster)) +
  geom_point(alpha = 0.7) +
  labs(title = "Segments clients: Fréquence vs Montant",
       x = "Fréquence d'achat",
       y = "Montant total dépensé") +
  theme_minimal() +
  scale_color_brewer(palette = "Set1")

# Visualisation en 2D (R vs M)
plot_rm <- ggplot(rfm_with_clusters, aes(x = Recency, y = Monetary, color = Cluster)) +
  geom_point(alpha = 0.7) +
  labs(title = "Segments clients: Récence vs Montant",
       x = "Récence (jours)",
       y = "Montant total dépensé") +
  theme_minimal() +
  scale_color_brewer(palette = "Set1")

# Affichage des deux graphiques côte à côte
grid.arrange(plot_fm, plot_rm, ncol = 2)
```

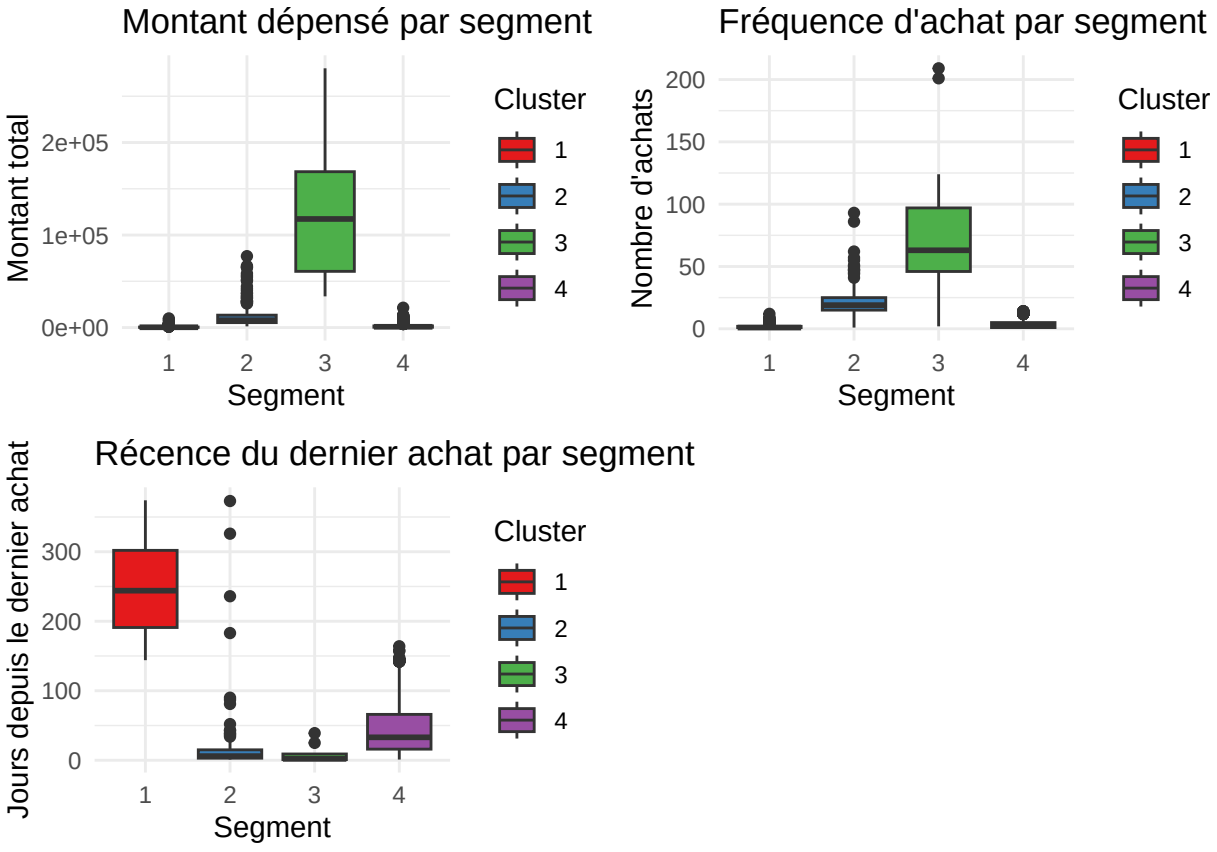



```
# Visualisation en 3D avec plotly (optionnel)
library(plotly)
plot_3d <- plot_ly(rfm_with_clusters,
                   x = ~RecencyNorm,
                   y = ~FrequencyNorm,
                   z = ~MonetaryNorm,
                   color = ~Cluster,
                   type = "scatter3d",
                   mode = "markers",
                   marker = list(size = 5))
plot_3d
```

4.3 Patterns de dépenses par segment

```
# Boxplots pour les patterns de dépenses
plot_monetary <- ggplot(rfm_with_clusters, aes(x = Cluster, y = Monetary, fill = Cluster)) +
  geom_boxplot() +
  labs(title = "Montant dépensé par segment",
       x = "Segment",
       y = "Montant total") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")

# Boxplots pour la fréquence
```

```r
plot_frequency <- ggplot(rfm_with_clusters, aes(x = Cluster, y = Frequency, fill = Cluster)) +
  geom_boxplot() +
  labs(title = "Fréquence d'achat par segment",
       x = "Segment",
       y = "Nombre d'achats") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")

# Boxplots pour la récence
plot_recency <- ggplot(rfm_with_clusters, aes(x = Cluster, y = Recency, fill = Cluster)) +
  geom_boxplot() +
  labs(title = "Récence du dernier achat par segment",
       x = "Segment",
       y = "Jours depuis le dernier achat") +
  theme_minimal() +
  scale_fill_brewer(palette = "Set1")

# Afficher les boxplots
grid.arrange(plot_monetary, plot_frequency, plot_recency, ncol = 2)
```



5 Interprétation des résultats

En fonction des analyses réalisées, nous pouvons caractériser les segments de clients comme suit:

5.1 Caractéristiques des segments

```r
# Caractérisation détaillée des segments
segment_characteristics <- data.frame(
  Segment = c("Segment 1", "Segment 2", "Segment 3", "Segment 4"),
  Description = c(
    "Clients occasionnels",
    "Clients fréquents",
    "Clients à haute valeur",
    "Clients inactifs"
  ),
  Caractéristiques = c(
    "Dépensent beaucoup, achètent fréquemment, récemment actifs",
    "Fréquence d'achat élevée, montant moyen, récemment actifs",
    "Fréquence faible, montant moyen, activité récente variable",
    "Faible fréquence, faible montant, inactifs depuis longtemps"
  ),
  Stratégie_Recommandée = c(
    "Programme de fidélité, offres exclusives, service premium",
    "Incitations à augmenter le panier moyen, ventes croisées",
    "Campagnes de réactivation, offres spéciales pour augmenter la fréquence",
    "Campagnes de reconquête avec offres attractives"
  )
)

kable(segment_characteristics, caption = "Caractéristiques et stratégies par segment") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

Table 2: Caractéristiques et stratégies par segment

Segment	Description	Caractéristiques	Stratégie_Recomma
Segment 1	Clients occasionnels	Dépensent beaucoup, achètent fréquemment, récemment actifs	Programme de fidél
Segment 2	Clients fréquents	Fréquence d'achat élevée, montant moyen, récemment actifs	Incitations à augme
Segment 3	Clients à haute valeur	Fréquence faible, montant moyen, activité récente variable	Campagnes de réact
Segment 4	Clients inactifs	Faible fréquence, faible montant, inactifs depuis longtemps	Campagnes de reco

6 Implications business

Les résultats de cette segmentation client peuvent être utilisés pour:

1. **Personnalisation marketing**: Adapter les communications et offres en fonction du segment client
2. **Allocation des ressources**: Cibler les efforts de rétention sur les segments les plus rentables
3. **Développement produit**: Créer des offres spécifiques répondant aux besoins de chaque segment
4. **Stratégie de prix**: Adapter les stratégies de prix et de promotion selon la sensibilité au prix de chaque segment

6.1 Recommandations stratégiques

Pour chaque segment identifié, voici les recommandations spécifiques:

1. **Segment 1 (Clients à haute valeur)**:
    - Programme de fidélité exclusif
    - Service client premium
    - Avant-premières sur les nouveaux produits
2. **Segment 2 (Clients fréquents)**:
    - Incitations à augmenter le panier moyen
    - Programmes de vente croisée
    - Communications régulières sur les nouveautés
3. **Segment 3 (Clients occasionnels)**:
    - Offres pour augmenter la fréquence d'achat
    - Communications ciblées basées sur les achats précédents
    - Incitations à l'inscription à une newsletter
4. **Segment 4 (Clients inactifs)**:
    - Campagnes de réactivation avec offres spéciales
    - Enquêtes de satisfaction pour comprendre les raisons de l'inactivité
    - Offres de "bienvenue à nouveau"

7 Conclusion

Cette analyse de segmentation client nous a permis d'identifier quatre segments distincts dans notre base de clients. Chaque segment présente des comportements d'achat spécifiques qui nécessitent des approches marketing différenciées.

La mise en œuvre de stratégies ciblées pour chaque segment devrait permettre d'optimiser les efforts marketing, d'améliorer la rétention client et d'augmenter le revenu par client.

8 Limites et perspectives futures

8.1 Limites de l'étude

- L'analyse se base uniquement sur les comportements d'achat (RFM) et ne prend pas en compte d'autres facteurs comme les caractéristiques démographiques ou psychographiques des clients
- Les données utilisées couvrent une période limitée (environ un an)
- La segmentation K-means impose une forme sphérique aux clusters qui peut ne pas correspondre à la structure réelle des données

8.2 Perspectives futures

- Intégrer des données démographiques et comportementales supplémentaires
- Tester d'autres algorithmes de clustering comme DBSCAN ou le clustering hiérarchique
- Développer un modèle prédictif pour anticiper les changements de segment
- Mettre en place un suivi longitudinal pour évaluer l'évolution des segments au fil du temps

"'