Class 8: Customer Segmentation Using Unsupervised Learning for M&S

Dr. Wei Miao

UCL School of Management

October 23, 2024

Section 1

(Case Study) Customer Segmentation for M&S

Customer Segmentation

Segmentation is a key step in the marketing strategy (STP) process, where customers are divided into meaningful groups based on characteristics relevant to designing and executing your marketing strategy.

It assumes that different customer groups provide varying levels of value to the company and/or require distinct marketing programs to succeed (e.g., based on differing goals and needs).

Conventional Segmentation

- **Customer value segmentation** is for targeting decisions based on customers' potential long-term financial and strategic value to your company.
- **Demographic segmentation** uses variables such as age, gender, income, family life cycle, educational qualification, socio-economic status, religion, company size and income, etc. These serve as proxies for goals, preferences or psychographics, as well as to characterize segments for marketing mix decisions.
- **Psychographic segmentation** is for positioning and marketing mix design based on the psychology of the customer and consumer, including attitudes, identity, lifestyle, personality, etc.

Conventional segmentation methods often require **subjective** judgments. A more objective approach is to 'let the data speak' by utilizing data analytics tools.

Syntax of `kmeans()`

```
kmeans(x, centers, iter.max = 10, nstart = 1,
       algorithm = c("Hartigan-Wong", "Lloyd", "Forgy",
                     "MacQueen"), trace=FALSE)
```

- `x`: data with selected variables to apply K-means
- `centers`: number of clusters
- `iter.max`: the maximum number of iterations allowed
- `nstart`: how many random sets should be chosen
- `algorithm`: which algorithm to choose; default often works
- `trace`: do you want to trace intermediate steps?

Section 2

Data Pre-processing

Data Loading

- Let's first try customer segmentation based on *total spending* and *Income*.
- **Exercise**: load data_full, create `total_spending`, and select `total_spending` and `Income` as the clustering variables into a new data frame `data_kmeans`.

Data Pre-processing

- To perform a cluster analysis in R, generally, the data should be prepared as follows:
  - Rows are observations (individuals) and columns are variables of interest for clustering.
  - Any missing value in the data must be removed or imputed.
  - The data must be standardized (i.e., scaled) to make variables comparable. Standardization consists of transforming the variables such that they have mean zero and standard deviation one.[1]

[1] Another common method is to normalize the data, which consists of transforming the variables such that they have a minimum of zero and a maximum of one.

Data Pre-processing: Missing Values

- Check if there are any missing values in the data.
- Use mean imputation to fill in missing values.

Data Pre-processing: Standardization

- Need to re-scale the clustering variables using `scale()`, because the variables can be of very different scales.
  - **Exercise**: Scale the variables and create a new data frame `data_kmeans_scaled`.
  - **This is extremely important!**

```
# method 1
data_kmeans_scaled <- data_kmeans %>%
    select(total_spending, Income) %>%
    mutate(
        total_spending = scale(total_spending),
        Income = scale(Income)
    )

# method 2: using across when there are many variables with the same tr
data_kmeans_scaled <- data_kmeans %>%
    select(total_spending, Income) %>%
    mutate(across(everything(), scale))
```

Visualization of the Data

- Let's visualize the data to see if there are any natural clusters.
- **Exercise**: Create a scatter plot of `total_spending` and `Income` using `ggplot2`.
- Refer to the ggplot2 cheat sheet for more information on data visualization in R.

ggplot2

ggplot2 is a system for declaratively creating graphics, based on The Grammar of Graphics. You provide the data, tell ggplot2 how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details. Go to docs...

Section 3

Apply K-Means

Apply K-Means Clustering with 2 Clusters

- `set.seed()` is to allow replication of results.
- `kmeans()` is the function to perform K-means clustering.
- `centers` is the number of clusters to form.
- `nstart` is the number of sets to be chosen.

```
set.seed(888)
result_kmeans <- kmeans(data_kmeans_scaled,
    centers = 2,
    nstart = 10
)
```

Examine the returned object, `result_kmeans`

```
tidy(result_kmeans)
```

```
# A tibble: 2 x 5
  total_spending Income  size withinss cluster
           <dbl>  <dbl> <int>    <dbl> <fct>
1          1.02   0.950   822     726. 1
2         -0.713 -0.663  1178     553. 2
```

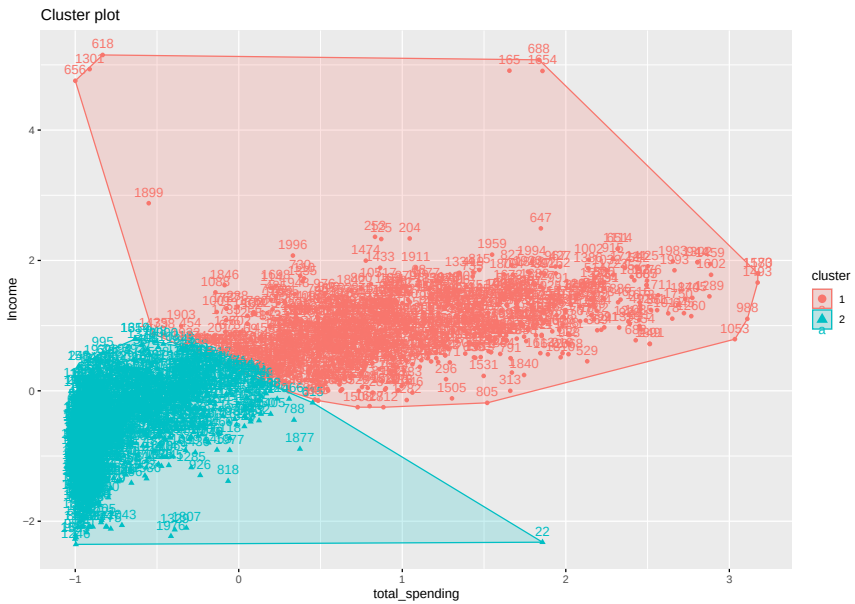
- `size`: The number of points in each cluster.
- `cluster`: **A vector of integers (from 1:k) indicating the cluster to which each point is allocated.**
- `withinss`: Vector of within-cluster sum of squares, one component per cluster.

Visualize the clusters

- We need 2 packages `cluster` and `factoextra`
- Use function `fviz_cluster()` to generate visualizations

```
pacman::p_load(cluster, factoextra)

fviz_cluster(result_kmeans,
     data = data_kmeans_scaled
)
```



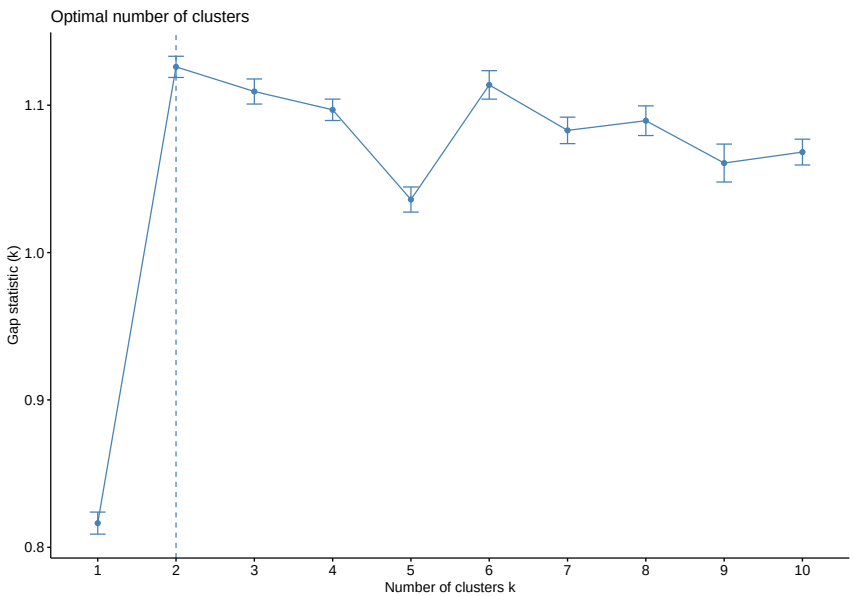
Section 4

Determine the K Value

Determine the optimal number of clusters: GAP Method

- The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. The estimate of the optimal clusters will be the value that maximizes the gap statistic.

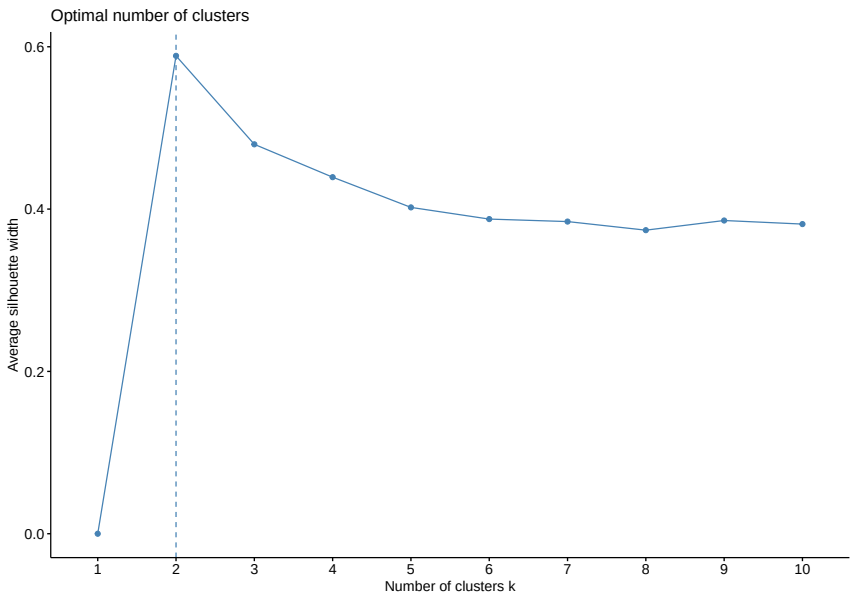
```
set.seed(888)
data_kmeans_scaled %>%
    clusGap(FUN = kmeans, K.max = 10, B = 50) %>%
    fviz_gap_stat()
```



Determine the optimal number of clusters: Silhouette Method

- The silhouette value measures how similar an object is to its cluster (cohesion) compared to other clusters (separation). The silhouette ranges from -1 to 1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters.

```
set.seed(888)
data_kmeans_scaled %>%
    fviz_nbclust(kmeans, method = "silhouette")
```



Next Steps After Segmentation

- Compare the CLV in different segments, and decide which segments to serve.
- Develop marketing strategies for each segment. For example, for the high-value segment, you may want to increase the frequency of purchase by offering discounts or promotions.
- Develop a customer journey map for each segment.

Term 3 Project Scopes

- Smartphones contain sensors, from which we can apply machine learning models to understand the context of the user, whether it be relaxing on the sofa, jogging in a park, or working indoors in an office. The task is to consume this real life data and produce visualisations, and to produce an anomaly detection engine. The project may be extended to clustering users according to their behavioural patterns in an unsupervised fashion.
- The project will explore **fraud detection approaches** using **unsupervised ML** including models such as isolation forests. The candidate will develop an understanding of the business problem and our data, formulating hypotheses and testing them. They will build, evaluate, and interpret their ML models.

After-Class Readings

- After-class Exercise: Try total spending, Frequency, and Recency as clustering covariates. Why these three variables? Then, find the optimal number of clusters. Visualize the clusters.
  - Because they are the most important variables for customer segmentation, i.e., RFM (Recency, Frequency, Monetary) analysis.
- Useful source: K-means Cluster Analysis