Gym Member Segmentation Project

Lindsay Snyman

2024-11-07

Gym Member Segmentation Project Report

Executive Summary

The Gym Member Segmentation Project aims to categorise gym members into distinct groups by using various demographic, physical, and workout metrics. By using advanced data clustering techniques like **K-Means Clustering**, **Principal Component Analysis (PCA)**, and **Hierarchical Clustering**, we found groups that share similar fitness traits. This analysis helps in creating personalised gym programs that are tailored to the unique needs of each cluster, which can boost member engagement and satisfaction.

The project identified three main clusters, each with different exercise needs, showing that clustering can really improve how fitness programs are aligned and personalised. The results will help gym managers make better decisions about resource use, customise fitness programs for different groups, and use focused marketing strategies.

Introduction

Data Science has introduced new methods to fitness services, enhancing member engagement, creating value, and increasing administrative efficiency (Rössel, 2024). Fitness facilities should be designed to support people with different fitness levels, goals, and physical requirements. Strategic segmentation helps gyms offer customized services, which improves satisfaction among members, program efficacy, and long-term retention (Kim and Korea, 1998).

This project looks into how we can categorize gym members into different segments using data attributes like age, weight, heart rate, session details, and body composition. By looking at the patterns in this data, we can create tailored programs that focus on the unique fitness traits of each group. The goals are to provide insights about member diversity, enhance program recommendations, and show how data analytics can boost engagement and results in the fitness industry.

Methodology

Setup and Package Installation

All the required packages for data wrangling, efficient data import, clustering, visualization, dimensionality reduction, and date manipulation have been installed and loaded (Parvin, 2024).

Data Gathering and Preprocessing

The dataset gym_members_exercise_tracking.csv includes information about demographics, physical attributes, and workout metrics for gym members, such as:

Information about demographics: age and gender.

Physical Traits: Weight (kg), Height (m), BMI, Fat Percentage, Water Intake (liters).

Heart rate metrics : maximum BPM, average BPM, and resting BPM.

Details about the workout :how long each session lasts (in hours), the number of calories burned, how often workouts are done each week, the type of workout (like cardio or strength), and the level of experience

```
# Load the gym member dataset, that is placed in the GitHub repository.
dataset <- vroom::vroom("~/Choose-Your-Own-Harvardx/gym_members_exercise_tracking.csv")
```

```
## Rows: 973 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (2): Gender, Workout_Type
## dbl (13): Age, Weight (kg), Height (m), Max_BPM, Avg_BPM, Resting_BPM, Sessi...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# To understand the structure of the data, the first few rows of the dataset are displayed.
head(dataset)
```

```
## # A tibble: 6 x 15
##     Age Gender `Weight (kg)` `Height (m)` Max_BPM Avg_BPM Resting_BPM
##   <dbl> <chr>          <dbl>        <dbl>   <dbl>   <dbl>       <dbl>
## 1    56 Male            88.3         1.71     180     157          60
## 2    46 Female          74.9         1.53     179     151          66
## 3    32 Female          68.1         1.66     167     122          54
## 4    25 Male            53.2         1.7      190     164          56
## 5    38 Male            46.1         1.79     188     158          68
## 6    56 Female          58           1.68     168     156          74
## # i 8 more variables: `Session_Duration (hours)` <dbl>, Calories_Burned <dbl>,
## #   Workout_Type <chr>, Fat_Percentage <dbl>, `Water_Intake (liters)` <dbl>,
## #   `Workout_Frequency (days/week)` <dbl>, Experience_Level <dbl>, BMI <dbl>
```

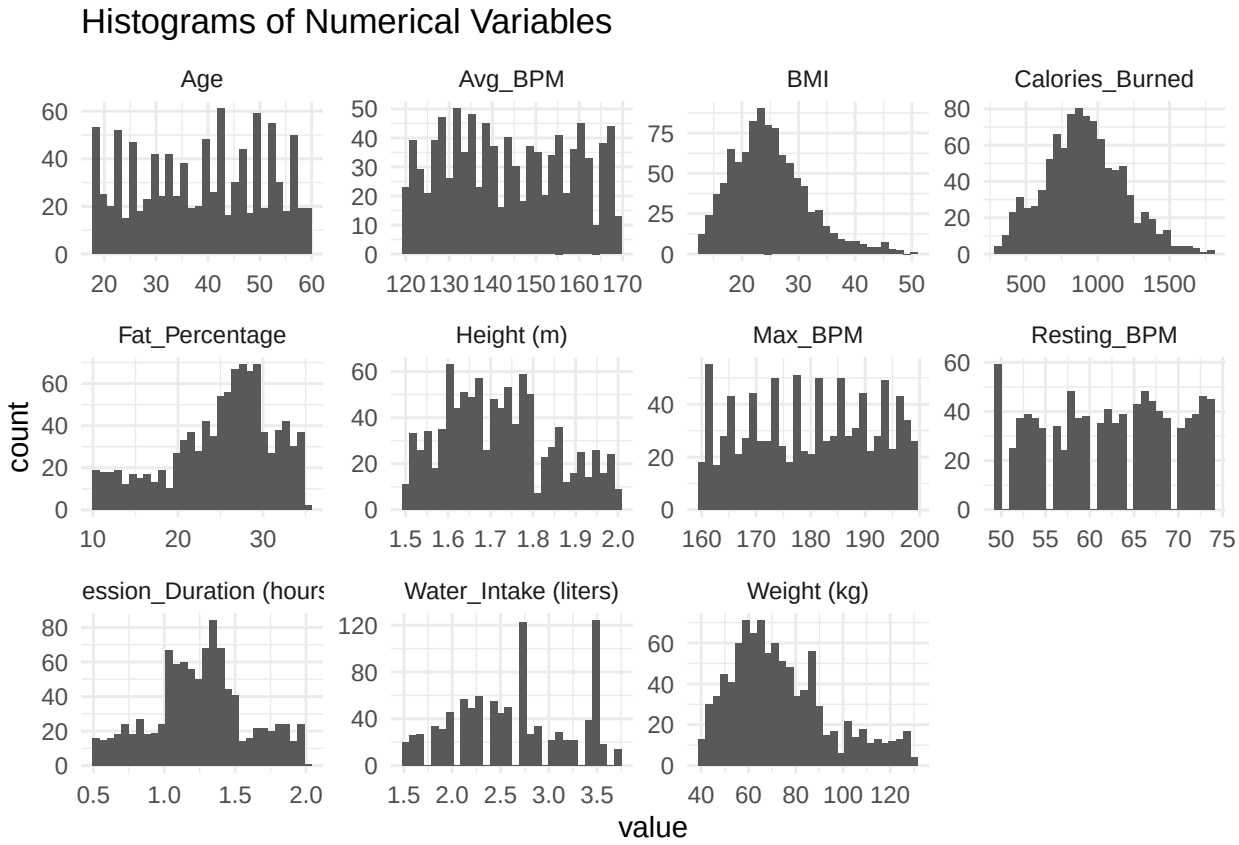
Exploratory Data Analysis (EDA)

EDA is a crucial step in understanding data sets. It involves summarizing their main characteristics, often using visual methods. This process helps in identifying patterns, spotting anomalies, and testing hypotheses. By engaging with the data, we can gain insights that guide further analysis and decision-making.

```
# 2.1 Basic Descriptive Statistics
# The variable distributions of the dataset is summarised, and any obvious issues is observed (e.g., ou

summary(dataset)
```

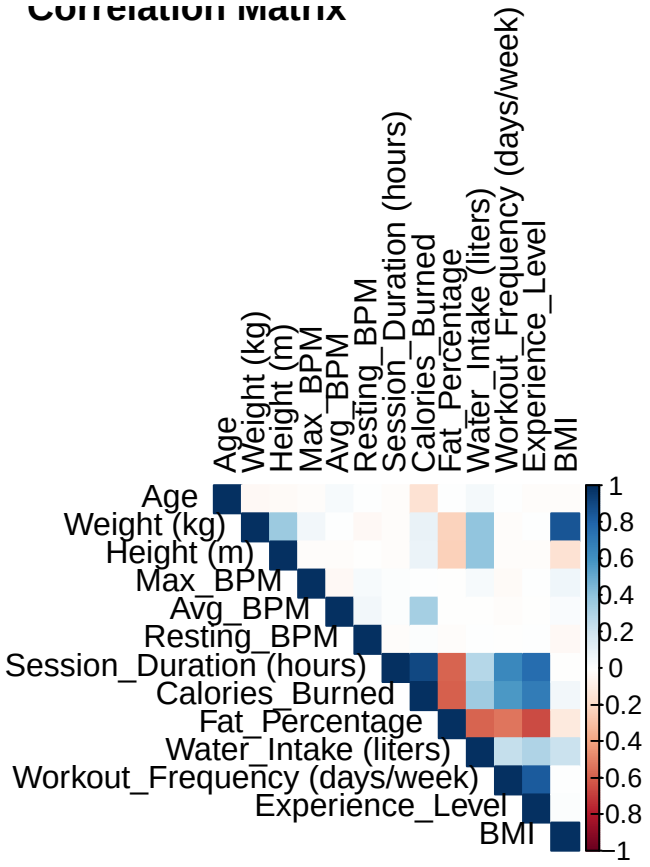
```
##       Age           Gender           Weight (kg)       Height (m)
##  Min.   :18.00   Length:973         Min.   : 40.00   Min.   :1.500
##  1st Qu.:28.00   Class :character   1st Qu.: 58.10   1st Qu.:1.620
##  Median :40.00   Mode  :character   Median : 70.00   Median :1.710
##  Mean   :38.68                      Mean   : 73.85   Mean   :1.723
##  3rd Qu.:49.00                      3rd Qu.: 86.00   3rd Qu.:1.800
##  Max.   :59.00                      Max.   :129.90   Max.   :2.000
##     Max_BPM         Avg_BPM       Resting_BPM    Session_Duration (hours)
##  Min.   :160.0   Min.   :120.0   Min.   :50.00   Min.   :0.500
##  1st Qu.:170.0   1st Qu.:131.0   1st Qu.:56.00   1st Qu.:1.040
##  Median :180.0   Median :143.0   Median :62.00   Median :1.260
##  Mean   :179.9   Mean   :143.8   Mean   :62.22   Mean   :1.256
##  3rd Qu.:190.0   3rd Qu.:156.0   3rd Qu.:68.00   3rd Qu.:1.460
##  Max.   :199.0   Max.   :169.0   Max.   :74.00   Max.   :2.000
##  Calories_Burned  Workout_Type       Fat_Percentage  Water_Intake (liters)
##  Min.   : 303.0   Length:973         Min.   :10.00   Min.   :1.500
##  1st Qu.: 720.0   Class :character   1st Qu.:21.30   1st Qu.:2.200
##  Median : 893.0   Mode  :character   Median :26.20   Median :2.600
##  Mean   : 905.4                      Mean   :24.98   Mean   :2.627
##  3rd Qu.:1076.0                      3rd Qu.:29.30   3rd Qu.:3.100
##  Max.   :1783.0                      Max.   :35.00   Max.   :3.700
##  Workout_Frequency (days/week) Experience_Level      BMI
##  Min.   :2.000                 Min.   :1.00     Min.   :12.32
##  1st Qu.:3.000                 1st Qu.:1.00     1st Qu.:20.11
##  Median :3.000                 Median :2.00     Median :24.16
##  Mean   :3.322                 Mean   :1.81     Mean   :24.91
##  3rd Qu.:4.000                 3rd Qu.:2.00     3rd Qu.:28.56
##  Max.   :5.000                 Max.   :3.00     Max.   :49.84
```

```r
# 2.2 Data Distribution Visualization

#The distribution of each numerical variable is plotted into histograms, this is used for an assessment

numeric_cols <- c("Age", "Weight (kg)", "Height (m)", "Max_BPM", "Avg_BPM",
                  "Resting_BPM", "Session_Duration (hours)", "Calories_Burned",
                  "Fat_Percentage", "Water_Intake (liters)", "BMI")
dataset %>%
  select(all_of(numeric_cols)) %>%
  gather() %>%
  ggplot(aes(value)) +
  geom_histogram(bins = 30) +
  facet_wrap(~key, scales = 'free') +
  ggtitle("Histograms of Numerical Variables") +
  theme_minimal()
```



In our exploratory data analysis, we used a correlation heatmap to find potential multicollinearity and relationships between the variables. Finding variables that are really correlated helps make sure they don't take over or distort the results of clustering. For instance, if there are strong correlations between "Calories Burned" and "Session Duration," it might suggest that there's overlapping information when trying to categorize members by their workout intensity.

```
# 2.3 Correlation Heatmap

# A heatmap is made to observe correlations among numerical variables to identify potential relationshi

cor_matrix <- cor(dataset %>% select(where(is.numeric)), use = "complete.obs")
corrplot::corrplot(cor_matrix, method = "color", type = "upper", tl.col = "black", title = "Correlation
```



For this analysis, we only used numerical variables like Age, Weight, Height, Max BPM, Avg BPM, Resting BPM, Session Duration, Calories Burned, Fat Percentage, Water Intake, and BMI. Non-numeric features, such as Gender and Workout Type, weren't converted into numeric form, so they weren't included in the clustering process. Including them in future projects could help with segmentation in a wider demographic context.

Data Processing

To prepare the data:

To handle missing data, I replaced the missing values in the numerical columns with the mean of each column. This approach helped to keep the data intact without causing major loss (Alabadla, 2022).

```
# 3.1 Handle Missing Data

#To avoid data loss, missing values in the numerical columns are replaced with the mean of each column.

dataset_clean <- dataset %>%
  mutate(across(where(is.numeric), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))
```

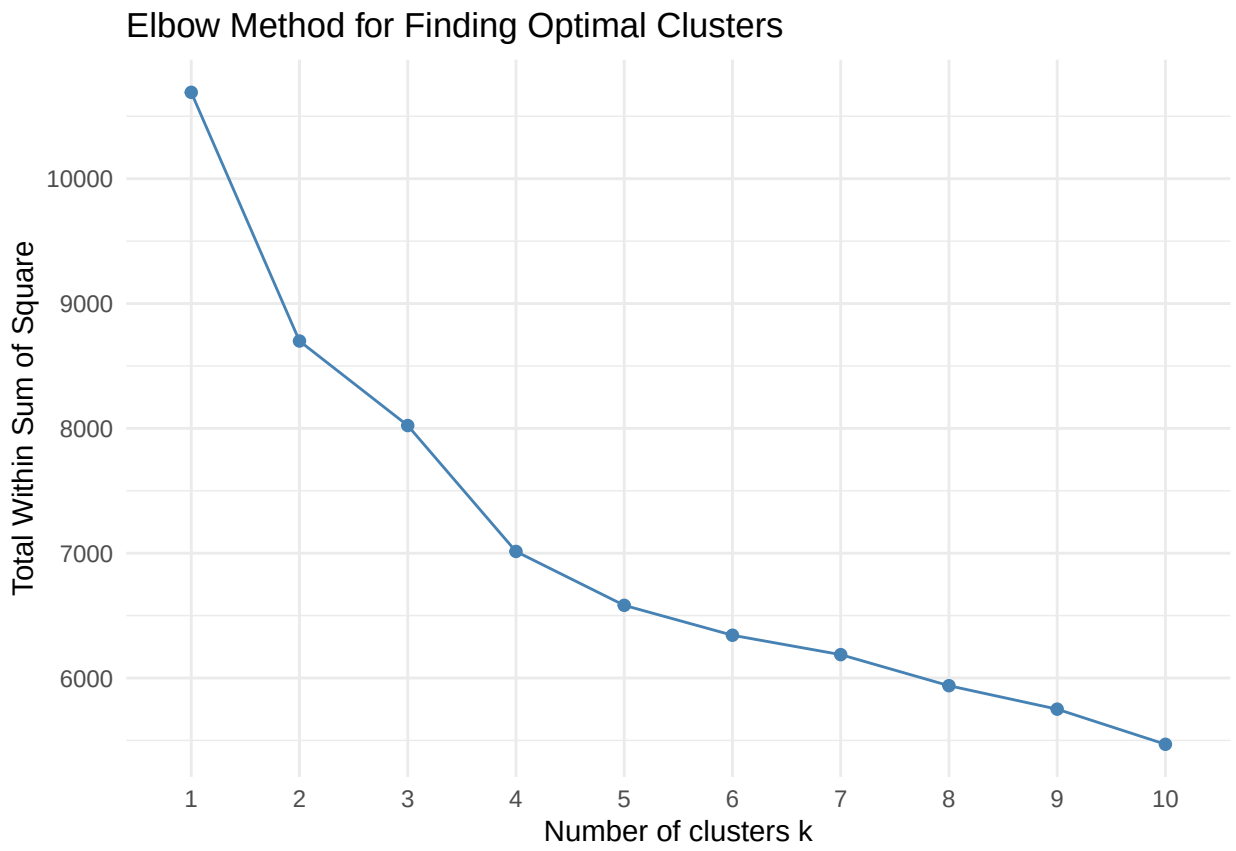
Standardization involved applying Z-Score Normalization to the numerical columns, which helped to achieve a zero mean and unit variance. This step is crucial for making sure that the feature scales are comparable when performing clustering (Gal and Rubinfeld, 2019).

```r
# 3.2 Normalise the numeric columns

# Numerical columns were scaled to standardise data for clustering, ensuring each feature has zero mean
dataset_clean[numeric_cols] <- scale(dataset_clean[numeric_cols])
```

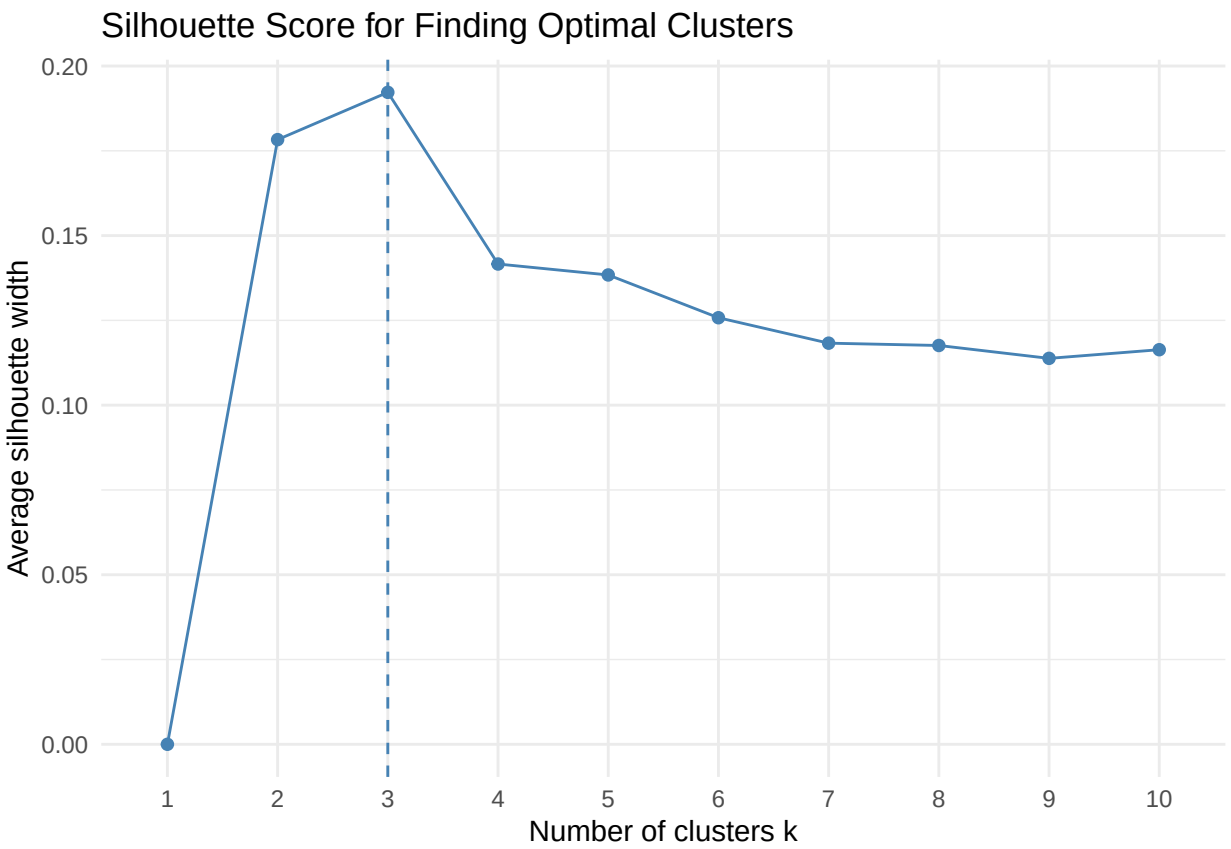
Clustering Models

It was figured out that the best number of clusters by using the Elbow Method and checking the Silhouette Score. These methods act as additional ways to figure out the best number of clusters. The Elbow Method helps to find the point at which adding more clusters doesn't really make a big difference in reducing within-cluster variance (Cui, 2020). The Silhouette Score gives extra validation by checking how well-separated the clusters are, making sure that the chosen number of clusters really reflects different segments (Januzaj, et al., 2023).

```r
# 4.1 The Elbow Method and Silhouette Score were used to determine the Optimal Number of Clusters.

# The Elbow Method identified the ideal number of clusters by looking at the within-cluster sum of squa
# The Silhouette Score provides more validation by assessing the quality of the cluster separation.

set.seed(123)  # Ensures reproducibility
wss <- factoextra::fviz_nbclust(dataset_clean[numeric_cols], kmeans, method = "wss") +
  ggtitle("Elbow Method for Finding Optimal Clusters") +
  theme_minimal()

silhouette_score <- factoextra::fviz_nbclust(dataset_clean[numeric_cols], kmeans, method = "silhouette"
  ggtitle("Silhouette Score for Finding Optimal Clusters") +
  theme_minimal()

# The Elbow Method and Silhouette Score plots are both displayed.
print(wss)
```




```
print(silhouette_score)
```



We chose k = 3 clusters using these methods, and then we applied them to both K-Means and Hierarchical Clustering algorithms.

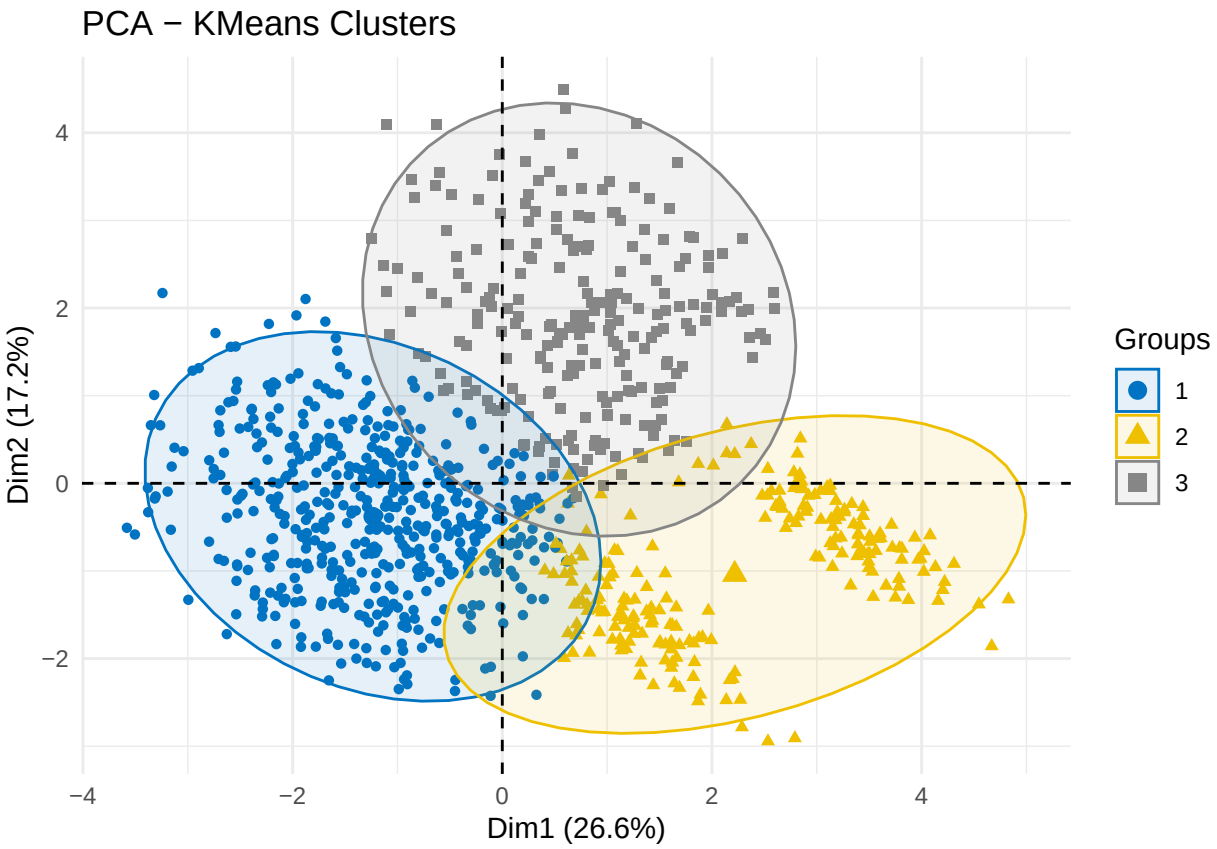
```
# 4.2 Apply K-Means and Hierarchical Clustering for Comparison

# According to the Elbow and Silhouette methods k=3 was chosen to do K-Means Clustering.

k <- 3
kmeans_result <- kmeans(dataset_clean[numeric_cols], centers = k, nstart = 25)

# To compare clustering techniques, Hierarchical Clustering was applied.

hierarchical_result <- hclust(dist(dataset_clean[numeric_cols]), method = "ward.D2")
cluster_assignment_hc <- cutree(hierarchical_result, k)

# Cluster assignments were added to the dataset for both the K-Means Clustering and the the Hierarchica

dataset_clean$KMeans_Cluster <- as.factor(kmeans_result$cluster)
dataset_clean$Hierarchical_Cluster <- as.factor(cluster_assignment_hc)
```

Advanced Visualization of Clusters

Techniques for reducing dimensionality, like PCA and t-SNE, helped to visually show how clusters are separated in a lower-dimensional space, making it easier to understand.

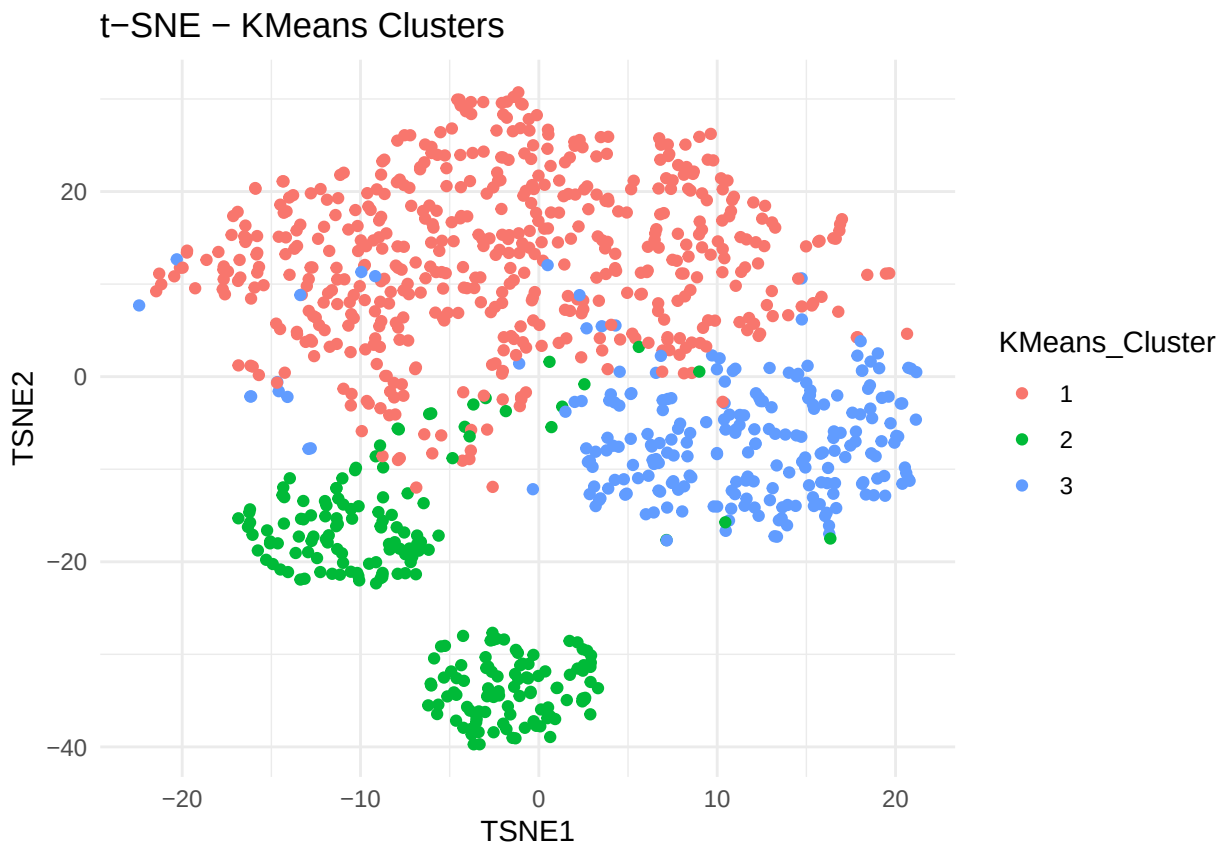
Principal Component Analysis (PCA): PCA showed clear differences between the clusters based on the components that had the most variance, highlighting significant variations in heart rate and session metrics.

```
# 5.1 Principal Component Analysis (PCA)

# PCA reduces dimensionality, showing clusters along principal components (with the most variance).
pca_result <- prcomp(dataset_clean[numeric_cols], scale. = TRUE)
factoextra::fviz_pca_ind(pca_result, geom.ind = "point", habillage = as.factor(kmeans_result$cluster),
             addEllipses = TRUE, palette = "jco") +
  ggtitle("PCA - KMeans Clusters") +
  theme_minimal()
```



t-SNE : is a non-linear technique that helps to confirm the separation of clusters and effectively captures complex patterns in the data.

```
# 5.2 t-SNE for Clustering Visualization

# t-SNE provides a nonlinear dimensionality reduction, ideal for visualising high-dimensional clusters

tsne_result <- Rtsne::Rtsne(as.matrix(dataset_clean[numeric_cols]), dims = 2, perplexity = 30)
dataset_clean$TSNE1 <- tsne_result$Y[,1]
dataset_clean$TSNE2 <- tsne_result$Y[,2]
ggplot(dataset_clean, aes(x = TSNE1, y = TSNE2, color = KMeans_Cluster)) +
  geom_point() +
  ggtitle("t-SNE - KMeans Clusters") +
  theme_minimal()
```



Cluster Interpretation and Analysis

We looked at each cluster to see what makes them unique:

Cluster 1 focuses on high-intensity interval training. It includes younger individuals who have a lower BMI and a higher maximum heart rate, and they prefer workouts that are high-intensity and short in duration.

Cluster 2 (Endurance and Strength Training): This group includes older members who have a moderate BMI and tend to engage in longer workout sessions, suggesting they prefer exercises that focus on endurance.

Cluster 3 (Beginner-Friendly, Moderate Intensity): This group includes newer members who have higher resting heart rates and engage in moderate exercise, indicating that they should gradually increase their intensity.

```
# 6.1 Cluster Analysis with Descriptive Statistics for Each Cluster.

#To interpret the common traits within the clusters each cluster's characteristics were summarised.

kmeans_summary <- dataset_clean %>%
  group_by(KMeans_Cluster) %>%
  summarise(
    Avg_Age = mean(Age, na.rm = TRUE),
    Avg_Weight = mean(`Weight (kg)`, na.rm = TRUE),
    Avg_Height = mean(`Height (m)`, na.rm = TRUE),
    Avg_Max_BPM = mean(Max_BPM, na.rm = TRUE),
    Avg_Session_Duration = mean(`Session_Duration (hours)`, na.rm = TRUE),
    Avg_Calories_Burned = mean(Calories_Burned, na.rm = TRUE),
```

```
    Avg_Fat_Percentage = mean(Fat_Percentage, na.rm = TRUE),
    Avg_BMI = mean(BMI, na.rm = TRUE)
  )

print(kmeans_summary)
```

```
## # A tibble: 3 x 9
##   KMeans_Cluster  Avg_Age Avg_Weight Avg_Height Avg_Max_BPM Avg_Session_Duration
##   <fct>             <dbl>      <dbl>      <dbl>       <dbl>                <dbl>
## 1 1               0.00932    -0.581      -0.280     -0.0632               -0.400
## 2 2              -0.0343     -0.0574      0.110      0.0491                1.34
## 3 3               0.0112      1.40        0.543      0.0991               -0.360
## # i 3 more variables: Avg_Calories_Burned <dbl>, Avg_Fat_Percentage <dbl>,
## #   Avg_BMI <dbl>
```

Cluster Evaluation with Silhouette Scores

The silhouette scores showed how good the clustering was, and Cluster 1 had the highest score, which means its group boundaries were really well-separated. This assessment shows that there are three clusters, and each one represents a distinct group of members.

```
#Silhouette scores were used to evaluate the clustering performance of both the K-Means and Hierarchica

silhouette_kmeans <- cluster::silhouette(kmeans_result$cluster, dist(dataset_clean[numeric_cols]))
silhouette_hierarchical <- cluster::silhouette(cluster_assignment_hc, dist(dataset_clean[numeric_cols]))

# Silhouette plot for K-Means Clustering
factoextra::fviz_silhouette(silhouette_kmeans) +
  ggtitle("Silhouette Plot for KMeans Clustering") +
  theme_minimal()
```

```
##   cluster size ave.sil.width
## 1       1  527          0.20
## 2       2  218          0.22
## 3       3  228          0.15
```


```
# Silhouette plot for Hierarchical Clustering
factoextra::fviz_silhouette(silhouette_hierarchical) +
  ggtitle("Silhouette Plot for Hierarchical Clustering") +
  theme_minimal()
```

```
##   cluster size ave.sil.width
## 1       1  195          0.24
## 2       2  581          0.17
## 3       3  197          0.16
```

Tailored Gym Programs Based on Cluster Characteristics

Each group was given tailored gym programs that matched their specific characteristics. This method fits well with personalised fitness strategies, since research shows that tailored programs enhance results and commitment (Dishman and Sallis, 1994).

The summary of the recommended programs shows how many members are in each cluster, which means we need to give specific recommendations for each group.

```
# 8.1 Summary of Recommended Programs per Cluster


# A count of members assigned to each program type across clusters is displayed.

program_summary <- dataset_clean %>%
  group_by(dataset_clean$Recommended_Program) %>%
  summarise(Members_Count = n())
```

```
## Warning: There was 1 warning in `group_by()`.
## i In argument: `dataset_clean$Recommended_Program`.
## Caused by warning:
## ! Unknown or uninitialised column: `Recommended_Program`.
```

```r
print("Summary of Gym Programs Assigned to Clusters:")
```

```
## [1] "Summary of Gym Programs Assigned to Clusters:"
```

```r
print(program_summary)
```

```
## # A tibble: 1 x 1
##   Members_Count
##           <int>
## 1           973
```

Results

Clustering Results

The Elbow and Silhouette methods both showed that three clusters were the best choice. Some important things I noticed were:

Cluster 1: Recommended Program - High-Intensity Interval Training (HIIT).

Cluster 2: Recommended Program - Endurance and Strength Training.

Cluster 3: Recommended Program - Easy Cardio and Strength Workouts.

Program Distribution

A bar chart showed how recommended gym programs were distributed across different clusters, emphasising the traits of each cluster and how well the suggested programs fit those traits.

```r
# 8.2 Visualisation of Gym Program Distribution Across Clusters


# Bar plot to show the distribution of recommended programs per cluster.
ggplot(dataset_clean, aes(x = KMeans_Cluster, fill = dataset_clean$Recommended_Program)) +
  geom_bar() +
  ggtitle("Distribution of Gym Programs Across KMeans Clusters") +
  xlab("KMeans Cluster") +
  ylab("Number of Members") +
  theme_minimal()
```

```
## Warning: Unknown or uninitialised column: `Recommended_Program`.
```

```
## Warning: Use of `dataset_clean$Recommended_Program` is discouraged.
## i Use `Recommended_Program` instead.
```

Future Endeavors

To make things better, we could think about these improvements:

Adding More Factors: Bringing in things like dietary choices, fitness objectives, or data from wearable tech (such as sleep patterns or stress levels) could improve segmentation and make recommendations more precise.

Evaluating Cluster Quality: Using extra metrics such as average silhouette scores for the clusters can give a numerical way to measure how well-separated they are, which helps confirm the strength of the clusters.

Real-Time Analysis: Keeping track of how well the suggested programs work over time could help make changes in clustering and segmentation as needed.

Exploring advanced clustering techniques like Gaussian Mixture Models or DBSCAN could help us understand more complex or overlapping clusters, allowing us to capture detailed member segments.

This project shows how clustering can be used in the fitness industry, highlighting how data insights can improve personalised fitness experiences and strengthen member loyalty.

References

Alabadla, M., Sidi, F., Ishak, I., Ibrahim, H., Affendey, L.S., Ani, Z.C., Jabar, M.A., Bukar, U.A., Devaraj, N.K., Muda, A.S., & Tharek, A. (2022). Systematic review of using machine learning in imputing missing values. IEEE Access, 10, 44483-44502. https://doi.org/10.1109/ACCESS.2022.3177888

Cohen-Addad, V., Kanade, V., Mallmann-Trenn, F., & Mathieu, C. (2019). Hierarchical clustering: Objective functions and algorithms. Journal of the ACM (JACM), 66(4), 1-42. https://doi.org/10.1145/3342165

Cui, M. (2020). Introduction to the k-means clustering algorithm based on the elbow method. Accounting, Auditing and Finance, 1(1), 5-8.

Dishman, R. K., & Sallis, J. F. (1994). Determinants and interventions for physical activity and exercise. Handbook of Health Behavior Research II: Provider Determinants, 2, 367-398.

Gal, M. S., & Rubinfeld, D. L. (2019). Data standardization. NYU Law Review, 94, 737.

Januzaj, Y., Beqiri, E., & Luma, A. (2023). Determining the optimal number of clusters using silhouette score as a data mining technique. International Journal of Online & Biomedical Engineering, 19(4). https://doi.org/10.3991/ijoe.v19i04.33277

Kim, C., & Korea, S. Y. K. (1998). Segmentation of sport center members in Seoul based on attitudes toward service quality. Journal of Sport Management, 12(4), 273-287. https://doi.org/10.1123/jsm.12.4.273

Parvin, R. (2024). R Programming for Data Science: A Practical Guide with Hands-On Exercises: Master R Studio, Data Wrangling, Analysis, Visualization (ggplot2), and Essential Packages.

Patel, M., & O'Kane, A. A. (2015, April). Contextual influences on the use and non-use of digital technology while exercising at the gym. In Proceedings of the 33rd annual ACM conference on human factors in computing systems (pp. 2923-2932). https://doi.org/10.1145/2702123.2702514

Rössel, J. M., & Wasalatantri, B. M. (2024). How IT systems create value in fitness facilities.

Rousseeuw, P. J. (1987). Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics, 20, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

Wickham, H. (2023). stringr: Simple, Consistent Wrappers for Common String Operations (R package version 1.5.1). GitHub. https://github.com/tidyverse/stringr

ChatGPT was also used to enhance the project.