Introduction

Nutrition is linked to both physical and mental well-being, the USDA National Nutrient data set (Kelly, 2020) offers the opportunity to conduct statistical analysis and build a model for real-world implications based on suggested daily nutritional intake (Benton & Young, 2018). Extreme diets are experiencing a surge in popularity, whether driven by new social norms or heightened marketing efforts emphasising their potential health and sustainability benefits, which should be well-researched (Clarys, P. et al., 2014). It would be highly beneficial to gain an understanding of a potentially healthy diet that does not necessitate the use of extra supplements. The goal of this analysis is to investigate the potential for differentiating food groups based on their nutritional values. To achieve the aims, Principal Component Analysis is employed, followed by the K-Means Clustering method. The results could provide a useful framework for identifying which foods should be prioritised in one's diet.

The subsequent out-of-scope objective is to compare two extreme diets: carnivore and vegan. The analysis would estimate the outcomes by comparing the daily caloric intake required to meet the recommended intake of each nutrient as per the U.S. Recommended Daily Allowance (US RDA). Vegan diet composition will approximately be described as 40% vegetables and fruit, 25% plant protein, 10% healthy fats, and 25% whole grains and starchy vegetables (Roaming Vegans, 2020). Carnivore: 50% red meat, 35% poultry, fish and eggs, 15% dairy products (People's Choice Beef Jerky, 2021). It is important to recognise that this comparison would disregard the unique metabolic effects of food, such as animal or plant-based food might have different effects on the human body beyond their mineral and vitamin content.

Methodology

Analysis was completed using R Studio IDE version 2022.02 (R Core Team, 2022). Libraries used: class, cluster, colorspace, corrplot, dendextend, kableExtra, factoextra, GGally, ggcorrplot, mclust, readr, scatterplot3d, tidyverse, viridis, wesanderson, cowplot, gtsummary, cellranger.

Data and Data Wrangling

The USA National Nutrient Data Base (Release 27) is a comprehensive dataset that encompasses information about various foods and their nutritional values (Kelly, 2016). The US RDA for Iron varies based on age and sex– thus, it was omitted for simplicity.

The variables containing short descriptions, descriptions, common names, manufacturer's titles, ID, and scientific names are omitted as only the nutrient density will be used for the analysis. The character variable that describes the food group is converted to a factor variable, it contains 25 categories, f.e. Dairy and Egg Products, Vegetable and Vegetable Products, Nut and Seed Products, etc. The other 38 variables are numerical and represent nutritional measurements for 100g of food, including kilocalories, macronutrients (carbs, fat, protein, sugar, and fiber), and micronutrients (vitamin A, B6, B12, C, and folate) in micrograms or milligrams, as well as some of the nutrient percentage of the US RDA. The 15 US RDA values are removed as they describe similar information. Food group labels are saved separately as y. The resulting matrix is $X_{original}$ with size of $23x8618$.

Exploratory Analysis

The frequencies of entries across food groups are not consistent, with vegetable, beef and baked goods having the most data; spices, meals, restaurant foods and nuts having the least. This may cause issues as most categories have some extreme anomalies. However, these anomalies were not treated as error inputs, as the data has been documented extremely well. The outliers are more likely to have been caused by the nature of the data and overlaps of foods expected in groups (f.e. restaurant foods include meats, baked goods, and side dishes). There is an abundance of positively correlated continuous numerical variables, including vitamin values, fat and kilocalories, and sugar and carbohydrates, there is also a strong negative correlation between carbohydrates and protein; these results call for dimensionality reduction and feature extraction.

Principal Component Analysis

Principal Component Analysis (PCA) was employed to reduce the dimensionality and complexity of the data while retaining most of the information, it is most useful to explore and visualise data, find patterns, and reduce the risk of overfitting. First, the $X_{original}$ variables are centred and scaled using standardisation($X_{scaled} = \frac{X-\text{mean}}{\rho}$). The PCA analysis was conducted on X_{scaled} and revealed that 14 *components* were needed to explain at least 90% (Dmitriy, 2019), in this case, 90.152%, of the variance. The resulting PCA dataset (X_{pca}) indicates that the data has a complex underlying structure, which also makes visualisation challenging.

K-Means Clustering, Unsupervised Learning Approach

As PCA has suggested the overlap of foods among food groups, K-Means clustering is further employed to discover underlying patterns. The Clustering was applied to the X_{scaled} data set to help visualise the clusters and identify whether the nutrition composition is informative enough to identify and summarise food groups. In the application of K-means clustering, certain considerations must be made, including the use of scaled data and the assumption of spherical clusters with consistent within-cluster variation across all variables. However, during exploratory analysis, anomalies were identified in the data that may have affected the positioning of centroids. Due to the data being well-documented, outliers have been retained.

The clustering process was started by choosing the number of clusters, in this case, multiple K's were tested, including 25 (corresponding to the number of food groups) and others 2 to 24, with the optimal number of clusters identified as 9 (see ***Figure 1***). The elbow method was used to check how many K clusters are advised by calculating the Within-Cluster-Sum of Squared Errors $WCSS$ for values of K, with optimum K being for which WSS becomes first start to diminish (ODSC - Open Data Science, 2018), suggested K was ambiguous (Mahendru, 2019). Using the silhouette method, i.e. interpretation and validation of consistency within clusters of data, the optimal number is identified to be 9. The cluster silhouette plot is subpar, width variation and negative values indicate a relatively poor fit. The negative values suggest that some points might have been assigned to the "wrong" cluster. After K selection, the clustering method repeatedly updates the labels based on the within-cluster sum of squares ($WCSS$) criterion until the optimal clustering is achieved.

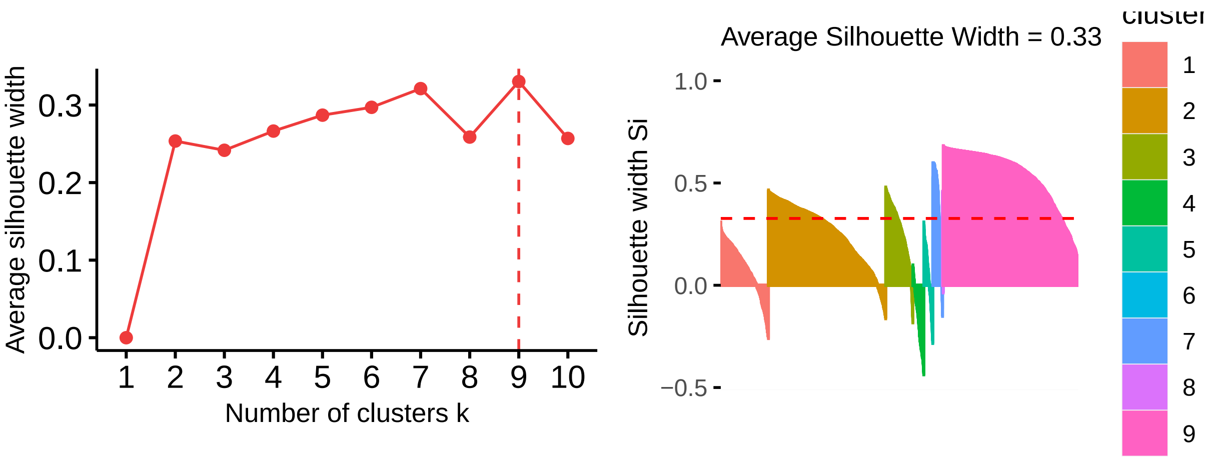

Figure 1: Left - Optimal Number of Clusters for K-Means determined by Silhouette Analysis; Right - Silhouette Plot of K-means Clustering (9 Clusters) Results with Silhouette Coefficient of 0.33

Results

PCA

Although principal component analysis has not been effective in reducing dimensionality to 2 or 3, it has indicated a complex underlying food group structure, signalling that there is an overlay of foods in different food groups, as well as an overlap of nutritional values within the same food group. This may be due to the presence of multiple variables that are highly correlated with each other or due to the existence of non-linear relationships among the variables (Firmin, 2019).

K-MEANS

The K-Means clustering has not been able to distinguish 25 separate food labels, however, other intricate results are visible as 9 clusters have been identified. The effectiveness of K-Means cluster visualisation in describing the results is limited due to the reduction of the data to the first two dimensions, namely calories and protein, which only account for 35.1% of the overall variation (see ***Figure 2*** Left). Furthermore, the visualisation is hindered by the presence of numerous clusters, resulting in unclear visual representation.

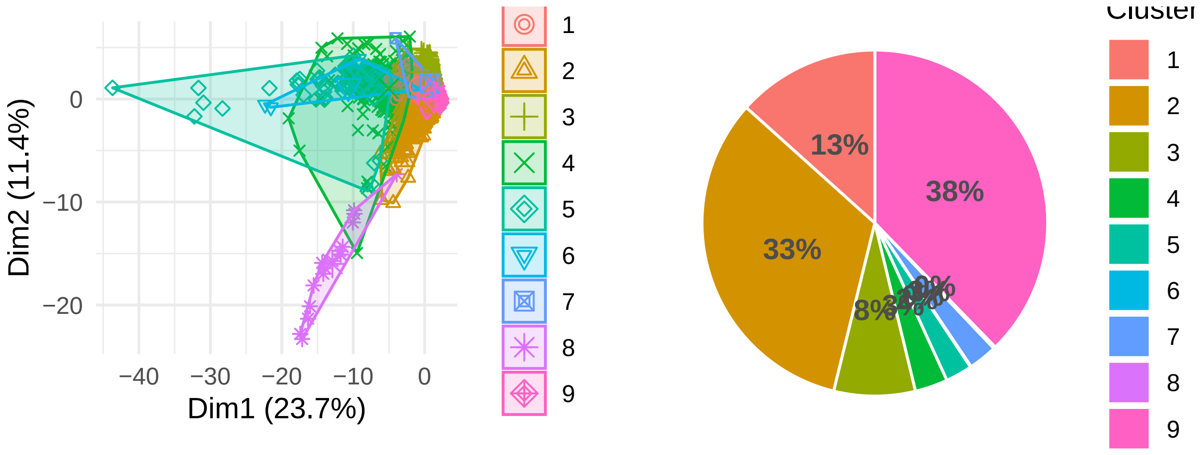

Figure 2: Left - K-Means Clusters Results Marked by Convex Polygons, Axes Represent First Two Dimensions (X - Calories, Y - Protein); Right - Resulting Percentage of Foods Assigned to Clusters

The results are well described in ***Table 1***, the continuation of it ***Table 2***, and ***Figures 3-5*** that represent the percentage of food group elements in respective clusters. Cluster 7 comprises beverages, fruit juices, and vegetables in equal proportions, whereas cluster 3 includes beef, poultry, lamb and veal, and fats and oils, indicating comparable nutritional values, including vitamin content, clustering in these cases seems reasonable. However, the presence of almost all food groups in cluster 8 suggests that this cluster may have been composed of outliers. The proportion division across the clusters is uneven, with 3 and 7 containing less than 1% of data (see ***Figure 2*** Right). The original frequency of data was uneven amongst food groups, but not as severely. Parallel coordinates describe the macronutrient composition of the clusters (see **Figure 6**), f.e., cluster 6 mainly contains sweets, baked goods, and baby food which is high in carbohydrates and sugar; cluster 7 - fibre and sugar, again, these results are justified considering the types of foods that have been clustered.

Table 1: Cluster Composition. Proportion of Each Food Group in a Cluster

Clusters	Baby	Baked	Beef	Cereals	Dairy, Eggs	Drinks	Fast Food	Fats, Oils	Fish	Fruit, Juice	Grains	Lamb, Veal.
1	1.04	39.01	-	10.95	0.17	0.96	8.86	0.17	-	0.7	10.34	-
2	0.07	-	31.9	-	3.82	-	4.95	-	8.42	-	0.04	14.36
3	7.47	44.82	-	1.52	1.98	4.27	0.46	0.61	-	4.12	-	-
4	0.38	1.52	-	5.68	3.79	6.06	-	-	1.89	-	2.65	-
5	5.58	0.47	-	83.72	-	0.47	-	-	-	-	-	-
6	-	-	-	-	-	33.33	-	-	-	33.33	-	-
7	-	-	5.63	-	1.73	-	0.43	68.4	-	-	-	6.93
8	-	-	31.25	-	-	-	-	6.25	-	-	-	50
9	8.8	1.51	0.8	0.98	3.91	7.91	3.84	1.66	0.74	9.5	1.72	0.25

Table 2: Cluster Composition. Proportion of Each Food Group in a Cluster (Continued)

Clusters	Legumes	Meals	Native Am	Nuts, Seeds	Pork	Poultry	Restaurant	Sausages	Snacks	Soups	Spices	Sweets	Vegetables
1	2.95	2.17	1.3	2.09	-	-	1.3	-	11.21	1.82	0.78	2.87	1.3
2	1.94	0.14	2.09	-	11.17	13.54	1.13	6.01	0.14	0.18	-	0.11	-
3	0.15	-	-	0.76	-	-	-	-	3.96	0.91	-	28.81	0.15
4	23.11	-	1.89	28.41	-	-	-	-	0.76	-	14.77	4.17	4.92
5	1.86	-	0.47	-	0.47	0.93	-	-	4.65	-	-	-	1.4
6	-	-	-	-	-	-	-	-	-	-	-	-	33.33
7	-	-	4.33	5.63	6.49	0.43	-	-	-	-	-	-	-
8	-	-	-	-	-	12.5	-	-	-	-	-	-	-
9	7.2	2.58	2.31	0.49	0.34	0.06	1.88	2.28	-	12.92	0.49	3.41	24.42

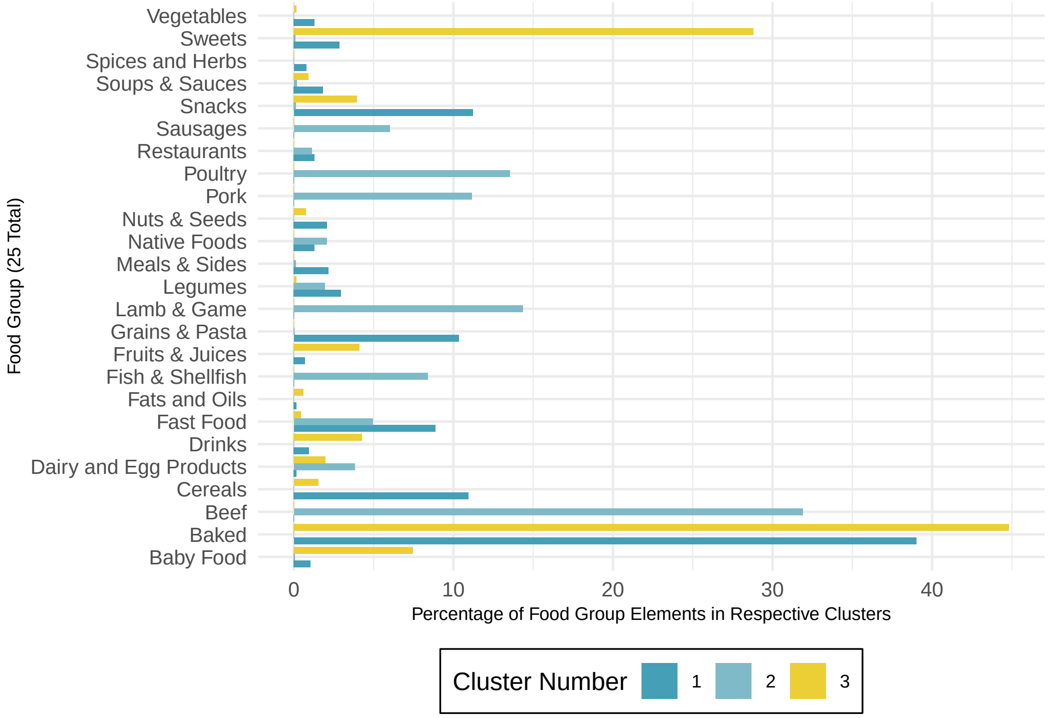

Figure 3: Composition of Three Clusters. Plot Presents Percentage of Food Group Elements in Respective Clusters

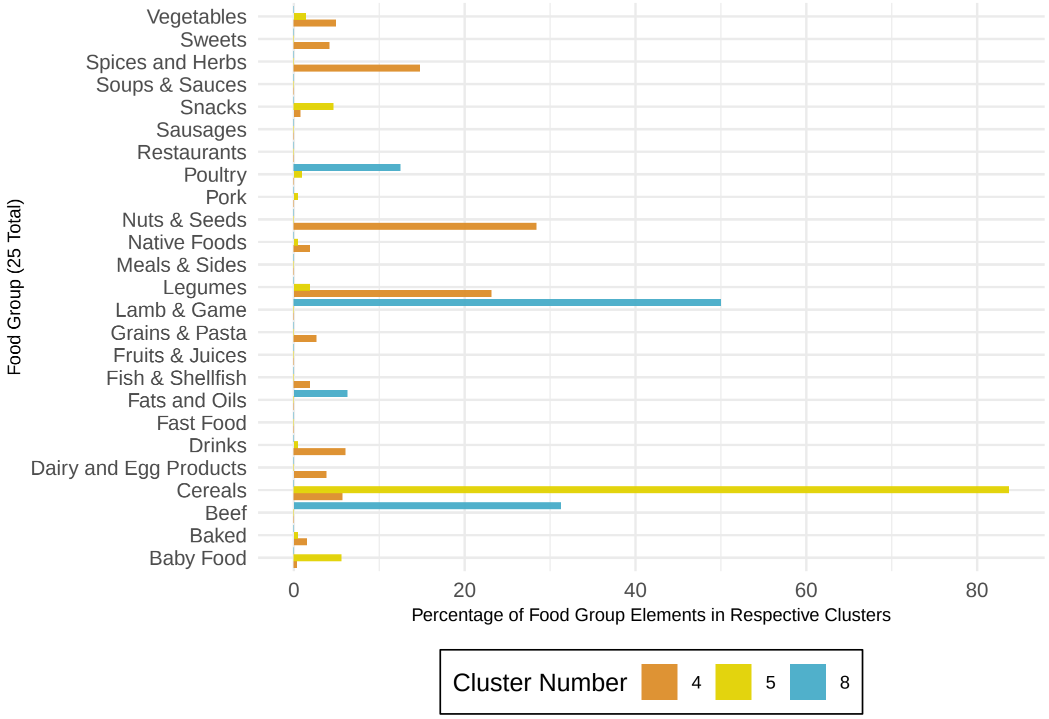

Figure 4: Composition of Three Clusters. Plot Presents Percentage of Food Group Elements in Respective Clusters

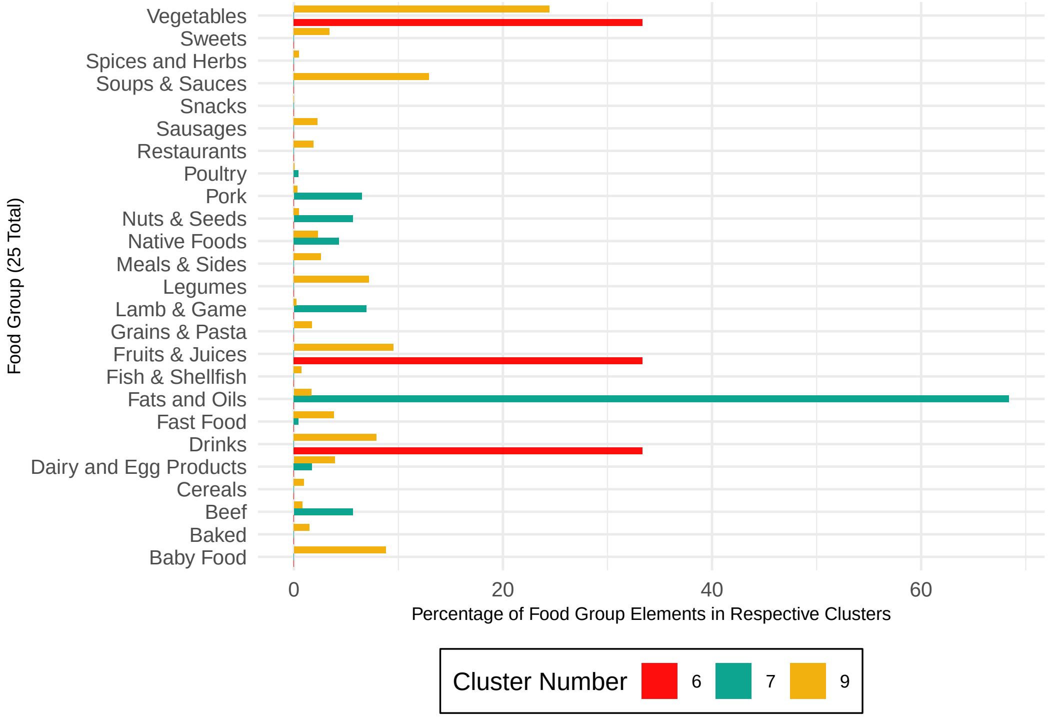

Figure 5: Composition of Three Clusters. Plot Presents Percentage of Food Group Elements in Respective Clusters

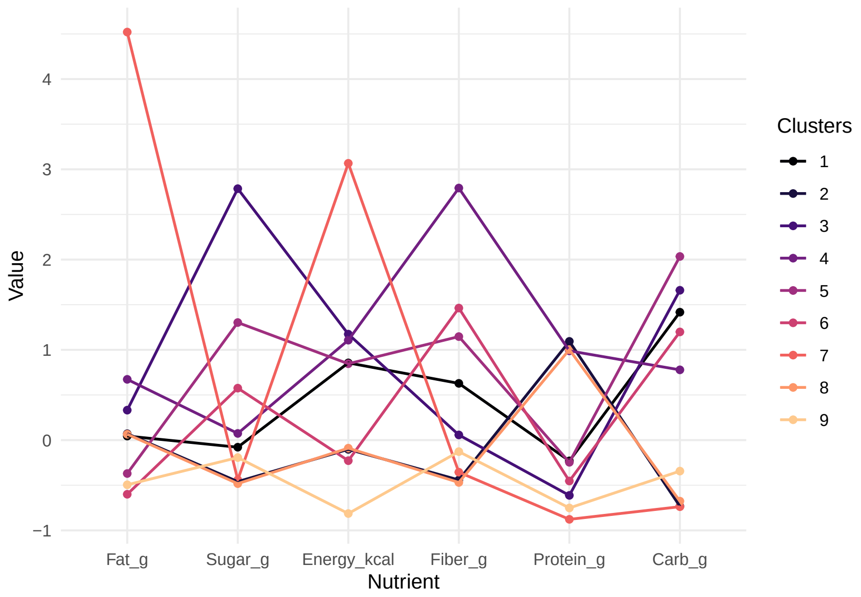

Figure 6: Parallel Coordinates Describing Cluster Composition by Part of the Macronutrients

Conclusions

Both Principal Component Analysis and K-Means Clustering results are signalling that some foods have similar nutrition values across food groups. Considering the types of foods included and the labels that have been imputed in the original data, an overlap of similarities is reasonable.

The strange difference in food labels assigned to the K-Means clusters indicates a sub-optimal model performance or/and that the given food groups are difficult to distinguish. As the K-Means method is sensitive to outliers, there may be other methods that are more reasonable to use for this specific data. Future studies could include Fuzzy clustering exploration, and the results might improve, however, due to the nature of the food types, it is unlikely to be able to identify 25 groups.

Finally, regarding the out-of-scope analysis mentioned earlier, a comparison of the macronutrient composition between carnivore and vegan diets should be conducted to determine if they are statistically different. This comparison can potentially be performed using the Mann-Whitney U or Wilcoxon rank sum test.

References

1. Roaming Vegans. (2020). What does it mean to be vegan? Roaming Vegans. Retrieved March 12, 2023, from https://roamingvegans.com/why-go-vegan/

2. People's Choice Beef Jerky. (2021, January 5). The Ultimate Carnivore Diet Food List, Meal Plan, And Shopping List. People's Choice Beef Jerky. Retrieved March 12, 2023, from https://peopleschoicebeefjerky.com/blogs/news/the-ultimate-carnivore-diet-food-list

3. Mahendru, K. (2019, June 17). How to Determine the Optimal K for K-Means? Medium. Retrieved March 12, 2023, from https://medium.com/analytics-vidhya/how-to-determine-the-optimal-k-for-k-means-708505d204eb

4. Benton, D., & Young, H. A. (2017). Reducing Calorie Intake May Not Help You Lose Body Weight, 12(5). https://doi.org/https://doi.org/10.1177/1745691617690878

5. Kelly, C. (2020). USDA National Nutrient DB. data.world. Retrieved March 12, 2023, from https://data.world/craigkelly/usda-national-nutrient-db

6. Dmitriy. (2019, February 21). Principal Component Analysis and k-means Clustering to Visualize a High Dimensional Dataset. medium. Retrieved March 12, 2023, from https://medium.com/more-python-less-problems/principal-component-analysis-and-k-means-clustering-to-visualize-a-high-dimensional-dataset-577b2a7a5fe2

7. ODSC - Open Data Science. (2018, December 17). Unsupervised Learning: Evaluating Clusters. medium. Retrieved March 12, 2023, from https://odsc.medium.com/unsupervised-learning-evaluating-clusters-bd47eed175ce

8. Firmin, S. (2019, July 29). Tidying up with PCA: An Introduction to Principal Components Analysis. Medium. Retrieved April 1, 2023, from https://towardsdatascience.com/tidying-up-with-pca-an-introduction-to-principal-components-analysis-f876599af383

9. R Core Team. (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing. https://www.R-project.org/.

Appendix

```r
##### SET UP #####

# Libraries required
library_list <- c("knitr", "class", "cluster", "colorspace", "corrplot", "dendextend", "kableExt

# Load the libraries
for (i in library_list) {
  library(i, character.only = TRUE)
}

# Set working directory
setwd("~/Documents/MT5758 Multivariate Analysis/Assignments/Nutrition_data/Nutrition")

# Load data
data <- read_csv("data.csv")






##### DATA WRANGLING #####

# Check for missing values in each column
missing <- colSums(is.na(data))

# Not numeric types
non_numeric_cols <- sapply(data, function(x) !is.numeric(x))
non_numeric_col_names <- names(data)[non_numeric_cols]

# Find number of categories of food groups
n <- unique(data$FoodGroup) %>% length()

# Convert the food groups variable to factors and save it as a separate vector
labels <- data$FoodGroup

# Delete the unwanted columns with character variables, delete USRDA variables
#   as they are basically duplicates of the nutrition values
data <- data %>% dplyr::select(-c(1:7, 31:45))

# Convert all the other variables to numeric
data <- lapply(data, function(x) if(is.numeric(x)) x else as.numeric(as.character(x)))

# Convert to data frame
data <- as.data.frame(data)
```

```r
#### FREQUENCY PLOT

# Save the data frame
data_full <- cbind(data, labels)

# Find number of entries in each category
group_freq <- table(data_full$labels)

# Sorted food labels
fg <- c("Dairy, Eggs", "Spices", "Baby Foods", "Fats, Oils", "Poultry",
        "Soups, Sauces", "Sausages, Luncheon", "Br Cereals",
        "Snacks", "Fruit, Fr Juice", "Pork", "Vegetables",
        "Nuts, Seeds", "Beef", "Beverages", "Finfish, Shellf.",
        "Legumes", "Lamb, Veal, Game", "Baked Products",
        "Sweets", "Cereal Gr, Pasta", "Fast Foods", "Meals, Entr, Side",
        "American Native", "Restaurant F") %>%
  sort()

# Set colours to be different for each cluster
my_col1 <- wesanderson::wes_palette(n = 1, "Zissou1")

# Convert the table to a data frame and abbreviate food group names
group_freq <- data.frame(Food_Group = fg,
                         Frequency = as.numeric(group_freq))

# Plot the frequencies
p1 <- ggplot(group_freq, aes(x = Food_Group, y = Frequency)) +
  geom_bar(stat = "identity", color = "white", alpha = 0.9,
           fill = my_col1, alpha = 0.8) +
  labs(x = "Food Group", y = "Frequency") +
  ylim(c = 0, 1000) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        panel.background = element_rect(fill = "#F5F5F5", color = NA)) +
  guides(fill = FALSE) +
  geom_rect(aes(xmin = as.numeric(Food_Group) - 0.5,
                xmax = as.numeric(Food_Group) + 0.5,
                ymin = 0, ymax = Frequency),
            fill = NA, color = "black")




##### CORRELATION #####

# Reset the grid
par(mfrow = c(1, 1))
```

```r
# Get correlation matrix and plot correlogram
cor_mat <- cor(data)

# Plot the matrix
corrplot.mixed(cor_mat, upper = "circle", addgrid = c("upper"),
               order = "hclust", tl.col = "black", tl.pos = "lt", diag = "l",
               number.font = 0.2, tl.cex = 0.5, number.cex = 0.55)






##### DATA SCALING AND CENTRE-ING #####

# Standardize the data to prepare for Principal Component Analysis
scaled_data <- scale(data, center = TRUE, scale = TRUE)

# Add the IDs and the food groups back to the scaled data set
foodGroup <- data.frame(foodGroup = labels)
dataf <- cbind(foodGroup, scaled_data)






##### PCA SELECTION #####

# Perform PCA on the scaled data
pca <- prcomp(scaled_data)

# Extract the principal component scores
PCA_components <- pca$x %>%
  data.frame()

### PCA variance
# Compute scores and get eigenvalues
eig <- eigen(cov(scaled_data))
Z <- scaled_data %*% eig$vectors

# Compute cumulative proportion of variance explained
pc_cumul_prop_var <- cumsum(eig$values/sum(eig$values))


### Plot both plots

# Set up the plot grid
```

```
par(mfrow = c(1, 2))


# Plot the PCA
plot(pca, type = "l",
xlab = "Principal Component Index",
ylab = "Proportion of Variance Explained",
     col = "brown2", lwd = 2)


# Plot cumulative proportion of variance explained
plot(pc_cumul_prop_var, xlab = "Principal Component Index",
     ylab = "Cum. Prop. of Var. Explained",
     ylim = c(0, 1),
     col = "brown2",
     lwd = 2)
abline(h = 0.9, lwd = 1.5, col = "azure4", lty = 2)
abline(v = 14, lwd = 1.5, col = "azure4", lty = 2)
text(14, 0.75, "14 PCA C.", col = "gray60", adj = c(0, -.1))

# Extract the first 14 principal components
pca_components <- pca$x[, 1:14]

# Reset the plot grid to default
par(mfrow = c(1, 1))

y <- foodGroup %>% as.vector()

# Plot biplot of the first two PCA components
fviz_pca_biplot(pca,
                label = "var",
                habillage = y)





### K-MEANS CLUSTERING

# Perform K-means clustering on the scaled data
kmeans_fit <- kmeans(scaled_data, centers = 9, nstart = 50, iter.max = 50)

# Get the cluster assignments for each observation
clustersAssigned <- kmeans_fit$cluster

# Add the cluster assignments to the data frame WITH PCA COMPONENTS, overwrite
# the scaled data
dataf <- cbind(foodGroup, scaled_data, clustersAssigned)
```

```
### EVALUATE CLUSTERS

# Calculate ARI and FMI
ari <- mclust::adjustedRandIndex(dataf$foodGroup, dataf$clustersAssigned)
fmi <- FM_index_R(dataf$foodGroup, dataf$clustersAssigned,  assume_sorted_vectors = FALSE)

# Print the results
cat("Adjusted Rand Index (ARI):", ari, "\n")
cat("Fowlkes-Mallows Index (FMI):", fmi, "\n")


### PLOT K MEANS

# Colour palette
pal <- c("red", "yellow")

# Scatterplot matrices coloured by clusters
pairs(pca_components[, 11:14], pch = 20, cex = 0.8, col = pal[clustersAssigned])


# Silhouette
silkmeans_p <- fviz_nbclust(scaled_data, kmeans, linecolor = "brown2", method = "s")

silkmeans_p2 <- silkmeans_p +
  ggtitle(" ") +
  theme(plot.title = element_text(size = 10),
        axis.title.x = element_text(size = 10),
        axis.title.y = element_text(size = 10))

# Total within sum of squares
wsskmeans <- fviz_nbclust(scaled_data, kmeans, linecolor = "brown2", method = "wss")

# Visualise silhouhette information
sil_p <- fviz_silhouette(silhouette(kmeans_fit$cluster, dist(scaled_data)),
                         print.summary = FALSE)

sil_p2 <- sil_p +
  ggtitle("Average Silhouette Width = 0.33") +
  theme(plot.title = element_text(size = 10),
        axis.title.x = element_text(size = 10),
        axis.title.y = element_text(size = 10))

# Set up the plot grid
par(mfrow = c(1, 1))
# Suggests optimal k = 9
# Suggests k = +- 4

# Together plots
```

```r
cowplot::plot_grid(silkmeans_p2, sil_p2, ncol = 2, height = 150)


# Visualise kmeans clustering
kmeans_p <- fviz_cluster(kmeans_fit, scaled_data, outlier.color = "black", geom = "point") +
  theme_minimal() +
  ggtitle(NULL)

# Count the number of entries in each cluster
cluster_counts <- table(kmeans_fit$cluster)

# Create a data frame with the cluster counts and percentages
pie_data <- data.frame(cluster = names(cluster_counts),
                       count = as.numeric(cluster_counts),
                       percent = round(as.numeric(cluster_counts) /
                                         sum(cluster_counts) * 100, 1))

# Plot the pie chart
kmeans_pie_p <- ggplot(pie_data, aes(x = "", y = count, fill = cluster)) +
  geom_bar(stat = "identity", width = 1) +
  geom_bar(stat = "identity", width = 1,
           color = "white", alpha = 0, lwd = 0.5) +
  coord_polar(theta = "y") +
  theme_void() +
  ggtitle("Cluster Counts") +
  ggtitle(NULL) +
  labs(fill = "Cluster") +
  geom_text(aes(label = paste0(round(count/sum(count)*100), "%")),
            position = position_stack(vjust = 0.5),
            color = "grey30", fontface = "bold")

# Plot together
cowplot::plot_grid(kmeans_p, kmeans_pie_p, ncol = 2, height = 150)


### PLOT THE COMPOSITION OF CLUSTERS

# Recode foodGroup factor levels with shorter labels
dataf$foodGroup_short <- dplyr::recode(dataf$foodGroup,
                                       "American Indian/Alaska Native Foods" = "Native Foods",
                                       "Baby Foods" = "Baby Food",
                                       "Baked Products" = "Baked",
                                       "Beef Products" = "Beef",
                                       "Beverages" = "Drinks",
                                       "Breakfast Cereals" = "Cereals",
                                       "Cereal Grains and Pasta" = "Grains & Pasta",
                                       "Dairy, Eggs" = "Dairy & Eggs",
                                       "Fast Foods" = "Fast Food",
                                       "Fats, Oils" = "Fats & Oils",
                                       "Finfish and Shellfish Products" = "Fish & Shellfish",
                                       "Fruits and Fruit Juices" = "Fruits & Juices",
                                       "Lamb, Veal, and Game Products" = "Lamb & Game",
                                       "Legumes and Legume Products" = "Legumes",
                                       "Meals, Entrees, and Side Dishes" = "Meals & Sides",
                                       "Nut and Seed Products" = "Nuts & Seeds",
                                       "Pork Products" = "Pork",
                                       "Poultry Products" = "Poultry",
                                       "Restaurant Foods" = "Restaurants",
                                       "Sausages and Luncheon Meats" = "Sausages",
                                       "Snacks" = "Snacks",
                                       "Soups, Sauces, and Gravies" = "Soups & Sauces",
                                       "Spices" = "Spices",
                                       "Sweets" = "Sweets",
                                       "Vegetables and Vegetable Products" = "Vegetables")

# Summarise the percentages of each food group in clusters
summaryTable <- dataf %>%
  group_by(clustersAssigned, foodGroup) %>%
  summarize(n = n()) %>%
  ungroup() %>%
  group_by(clustersAssigned) %>%
  # Shorten the titles of food groups to they fit in the table
  mutate(foodGroup_short = dplyr::recode(foodGroup,
                                         "clustersAssigned" = "Cluster",
                                         "American Indian/Alaska Native Foods" = "Native Am",
                                         "Baby Foods" = "Baby",
                                         "Baked Products" = "Baked",
                                         "Beef Products" = "Beef",
                                         "Beverages" = "Drinks",
                                         "Breakfast Cereals" = "Cereals",
                                         "Cereal Grains and Pasta" = "Grains",
                                         "Dairy and Egg Products" = "Dairy, Eggs ",
                                         "Fast Foods" = "Fast Food",
                                         "Fats and Oils" = "Fats, Oils",
                                         "Finfish and Shellfish Products" = "Fish",
                                         "Fruits and Fruit Juices" = "Fruit, Juice ",
                                         "Lamb, Veal, and Game Products" = "Lamb, Veal. ",
                                         "Legumes and Legume Products" = "Legumes",
                                         "Meals, Entrees, and Side Dishes" = "Meals",
                                         "Nut and Seed Products" = "Nuts, Seeds",
                                         "Pork Products" = "Pork",
                                         "Poultry Products" = "Poultry",
                                         "Restaurant Foods" = "Restaurant",
                                         "Sausages and Luncheon Meats" = "Sausages",
                                         "Snacks" = "Snacks",
                                         "Soups, Sauces, and Gravies" = "Soups",
                                         "Spices and Herbs" = "Spices",
                                         "Sweets" = "Sweets",
                                         "Vegetables and Vegetable Products" = "Vegetables")) %>
  mutate(prop = round(n * 100 / sum(n), 2)) %>%
  dplyr::select(-n, -foodGroup) %>%
  spread(foodGroup_short, prop) %>%
  rename(Clusters = clustersAssigned) %>%
  # Replace "NA" with "-"X
  mutate_all(~ifelse(. == "NA", "-", as.numeric(.))) %>%
  # Convert summary table from list to data frame
  as.data.frame()

# Get rid of NA's and add "-"
summaryTable[is.na(summaryTable)] <- "-"

# Check if the proportions have been calculated well
sum(summaryTable[2, 2:26], na.rm = TRUE)

cluster_majority <- character(nrow(summaryTable))
for (i in 1:nrow(summaryTable)) {
  cluster_majority[i] <- colnames(summaryTable)[which.max(summaryTable[i, ])]
}

# Change colours of the values for the summary table
for (i in i:ncol(summaryTable)) {
  summaryTable[, i] <- cell_spec(summaryTable[, i], color = ifelse(summaryTable[, i] > 20, "toma
}


# Cut the tables into two as they are too wide
summary_t1 <- summaryTable[, 1:13]
summary_t2 <- summaryTable[, c(1, 14:26)]

# Print the tables
kable(summary_t1, align = "c", caption = "Cluster Composition. Proportion of Each Food Group in
  row_spec(0, angle = 90) %>%
  kableExtra::kable_styling(
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!7",
    font_size = 8,
    full_width = TRUE) %>%
  kable_styling(latex_options = "HOLD_position")

kable(summary_t2, align = "c", caption = "Cluster Composition. Proportion of Each Food Group in
  row_spec(0, angle = 90)  %>%
  kableExtra::kable_styling(
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!7",
    font_size = 8,
    full_width = TRUE)  %>%
  kable_styling(latex_options = "HOLD_position")


kable(summary_t2, align = "c", caption = "Summary Table") %>%
  row_spec(0, angle = 90)  %>%
  kableExtra::kable_styling(
      latex_options = c("striped", "repeat_header"),
      stripe_color = "gray!7",
      font_size = 8,
      full_width = TRUE)





##### RESULTS OF THE CLUSTERING #####

# Group the data by cluster and foodGroup, count the frequency of each group, and summarize by t
df_summary <- dataf %>%
  group_by(clustersAssigned, foodGroup) %>%
  summarize(count = n()) %>%
  mutate(total_count = sum(count)) %>%
  mutate(percentage = count / total_count * 100) %>%
  dplyr::select(clustersAssigned, foodGroup, percentage) %>%
  pivot_wider(names_from = clustersAssigned,
              values_from = percentage,
              values_fill = 0) %>%
  arrange(foodGroup) %>%
  pivot_longer(cols = -foodGroup, names_to = "Cluster", values_to = "Percentage")


### PLOT RESULTS

# Set colours to be different for each cluster
my_col1 <- wesanderson::wes_palette(n = 3, "Zissou1")
my_col2 <- wesanderson::wes_palette(n = 3, "FantasticFox1")
my_col3 <- wesanderson::wes_palette(n = 3, "Darjeeling1")

# Recode the names so they fit in the table
df_summary$foodGroup <- dplyr::recode(df_summary$foodGroup,
                                      "American Indian/Alaska Native Foods" = "Native Foods",
                                      "Baby Foods" = "Baby Food",
                                      "Baked Products" = "Baked",
                                      "Beef Products" = "Beef",
                                      "Beverages" = "Drinks",
                                      "Breakfast Cereals" = "Cereals",
                                      "Cereal Grains and Pasta" = "Grains & Pasta",
                                      "Dairy, Eggs" = "Dairy & Eggs",
                                      "Fast Foods" = "Fast Food",
                                      "Fats, Oils" = "Fats & Oils",
                                      "Finfish and Shellfish Products" = "Fish & Shellfish",
                                      "Fruits and Fruit Juices" = "Fruits & Juices",
                                      "Lamb, Veal, and Game Products" = "Lamb & Game",
                                      "Legumes and Legume Products" = "Legumes",
                                      "Meals, Entrees, and Side Dishes" = "Meals & Sides",
                                      "Nut and Seed Products" = "Nuts & Seeds",
                                      "Pork Products" = "Pork",
                                      "Poultry Products" = "Poultry",
                                      "Restaurant Foods" = "Restaurants",
                                      "Sausages and Luncheon Meats" = "Sausages",
                                      "Snacks" = "Snacks",
                                      "Soups, Sauces, and Gravies" = "Soups & Sauces",
                                      "Spices" = "Spices",
                                      "Sweets" = "Sweets",
                                      "Vegetables and Vegetable Products" = "Vegetables")

# Filter the data frame to include only clusters 1, 2, and 3
df1 <- df_summary %>%
  filter(Cluster %in% c("1", "2", "3"))

# Create the side-by-side bar chart # 1-3 clusters
ggplot(df1, aes(x = Percentage, y = foodGroup, fill = Cluster)) +
  geom_bar(position = position_dodge(width = 0.9), alpha = 0.95, stat = "identity") +
  # set the color palette
  scale_fill_manual(values = my_col1) +
  labs(x = "Percentage of Food Group Elements in Respective Clusters",
       y = "Food Group (25 Total)", fill = "Cluster Number") +
  theme_minimal() +
  #  move legend to bottom
  theme(legend.position = "bottom",
        # add border to legend box
        legend.box.background = element_rect(color = "black"),
        # increase size of legend text
        legend.text = element_text(size = 8),
        # increase size of axis titles
        axis.title = element_text(size = 8))

# Filter the data frame to include only clusters 1, 2, and 3
df2 <- df_summary %>%
  filter(Cluster %in% c("4", "5", "8"))

# Create the side-by-side bar chart # 4-6 clusters
ggplot(df2, aes(x = Percentage, y = foodGroup, fill = Cluster)) +
  geom_bar(position = position_dodge(width = 0.9), alpha = 0.95, stat = "identity") +
  scale_fill_manual(values = my_col2) +
  labs(x = "Percentage of Food Group Elements in Respective Clusters",
       y = "Food Group (25 Total)", fill = "Cluster Number") +
  theme_minimal() +
  theme(legend.position = "bottom",
        legend.box.background = element_rect(color = "black"),
        legend.text = element_text(size = 8),
        axis.title = element_text(size = 8))



# Filter the data frame to include only clusters 1, 2, and 3
df3 <- df_summary %>%
  filter(Cluster %in% c("6", "7", "9"))

# Create the side-by-side bar chart # 7-9 clusters
ggplot(df3, aes(x = Percentage, y = foodGroup, fill = Cluster)) +
  geom_bar(position = position_dodge(width = 0.9), alpha = 0.95, stat = "identity") +
  scale_fill_manual(values = my_col3) +
  labs(x = "Percentage of Food Group Elements in Respective Clusters",
       y = "Food Group (25 Total)", fill = "Cluster Number") +
  theme_minimal() +
  theme(legend.position = "bottom",
        legend.box.background = element_rect(color = "black"),
        legend.text = element_text(size = 8),
        axis.title = element_text(size = 8))






##### CLUSTER EXPLORATION #####

# Create a data frame for the main nutrition components to be visualised
cluster_nutr <- dataf[, c(2:7, 25)]

# Create an empty list to store the cluster means
cluster_means <- list()
# Loop through the clusters and calculate the mean values of each variable
for (i in 1:9){
  cluster_data <- cluster_nutr[dataf$clustersAssigned == i, ]
  cluster_mean <- colMeans(cluster_data)
  cluster_means[[i]] <- cluster_mean
}

# Combine the cluster means into a data frame
cluster_means_df <- do.call(rbind, cluster_means) %>% data.frame()
```

```r
# Cluster column must be a factor
cluster_means_df$clustersAssigned <- as.factor(cluster_means_df$clustersAssigned)

# Create a parallel coordinates plot for the selected variables,
# with lines colored by cluster

# Pick colours
my_col4 <- viridis_pal(option = "magma")(10)

# Plot the parallel coordinates
ggparcoord(cluster_means_df, columns = 1:6, groupColumn = 7, order = "anyClass",
           showPoints = TRUE, scale = "globalminmax") +
  scale_color_manual(values = my_col4) +
  ylab("Value") +
  xlab("Nutrient") +
  scale_fill_discrete(name = "Clusters") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),
        fill = "Clusters") +
  geom_line(size = 0.7) +
  theme_minimal() +
  labs(color = "Clusters")
```