MULTIVARIATE STAT ANALYSIS

STA 5384 01

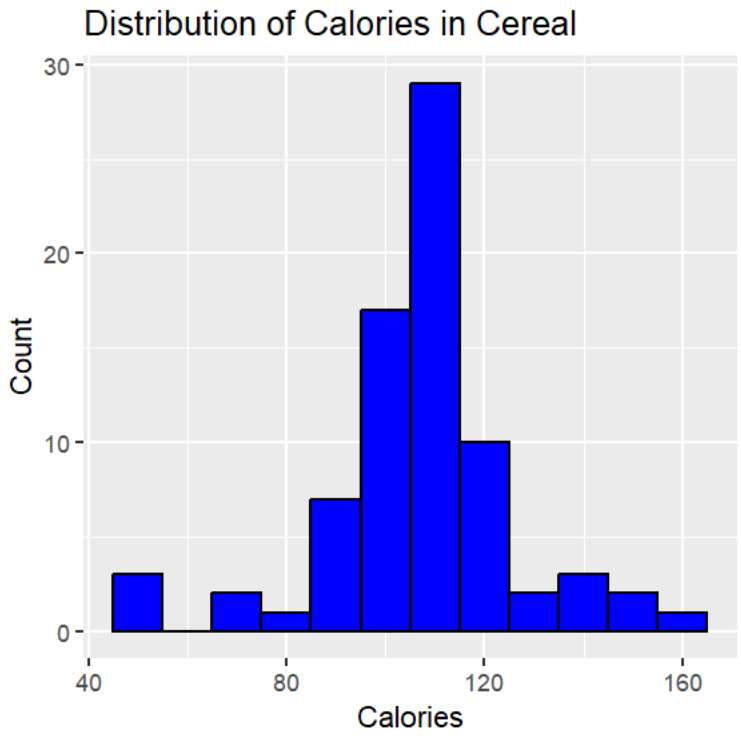
SALMAN IMTIAZ

892812224

1. **In this assignment you will be looking at the cereal dataset from the R package liver. Before starting, remove the \name", \manuf", \type", and \shelf" variables (as these are qualitative).**

R-Code:

```
if (!requireNamespace("liver", quietly = TRUE)) {
  install.packages("liver")
}


library(ggplot2)
library(liver)
library(dplyr)


data(cereal, package = "liver")


head(cereal)


# Remove the specified variables
cereal_cleaned <- select(cereal, -name, -manuf, -type, -shelf)


#Display ggplot
ggplot(cereal_cleaned, aes(x = calories)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black") +
  labs(title = "Distribution of Calories in Cereal", x = "Calories", y = "Count")


# View the first few rows of the cleaned dataset
head(cereal_cleaned)
```



Findings:

- The distribution of calories in cereal is somewhat skewed to the right, indicating that most cereals have a calorie count on the lower end of the spectrum.
- The most common calorie range for cereals appears to be around 100-120 calories, as indicated by the tallest bar in the histogram.
- There are a few cereals that have a calorie count significantly higher than the majority, which could be considered outliers.

Explanations of R code:

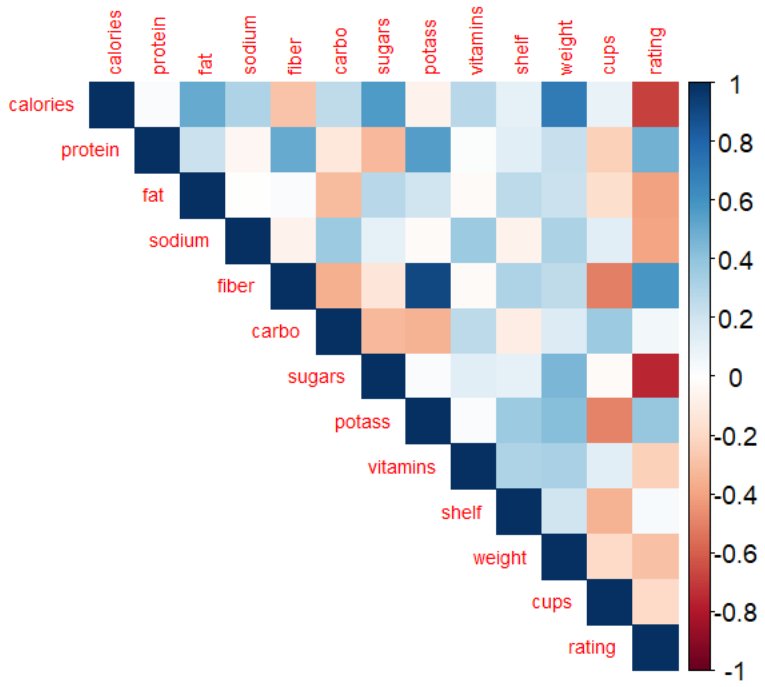
Data Preparation: First, the cereal dataset was prepared by eliminating qualitative factors and then the remaining quantitative variables' correlations were calculated using a correlation matrix. There are probably differing degrees of linear connections between the various nutritional characteristics of the grains that this analysis identified.

---

2. **Create and print the correlation matrix for cereal dataset. Round each correlation to 2 decimal places.**

R-Code:

```
library(dplyr)
library(ggplot2)
library(corrplot)

data(cereal, package = "liver")
head(cereal)

# Select only the numeric columns if your dataset contains non-numeric columns
cereal_numeric <- select_if(cereal, is.numeric)

# Compute the correlation matrix
cor_matrix <- cor(cereal_numeric)

# Round the correlation matrix to 2 decimal places
cor_matrix_rounded <- round(cor_matrix, 2)

# Display Correlation Plot
corrplot(cor_matrix_rounded, method = "color", type = "upper", tl.cex = 0.6)

# Print the rounded correlation matrix
print(cor_matrix_rounded)
```



Findings:

- The strongest positive correlations appear to be between pairs such as weight and cups, sugars and calories, and calories and fat. This suggests that cereals with higher fat content tend to have more calories, and those that have higher serving weights tend to require more cups per serving.
- There are moderate positive correlations between nutrients like protein and fat with calories, suggesting that cereals with more protein and fat may also be higher in calories. Also, dietary fiber shows a moderate positive correlation with protein.
- Some pairs of variables, such as vitamins and sugars or shelf placement and fiber, display very low to no correlation, suggesting no linear relationship between these variables in the given dataset.

Explanations of R code:

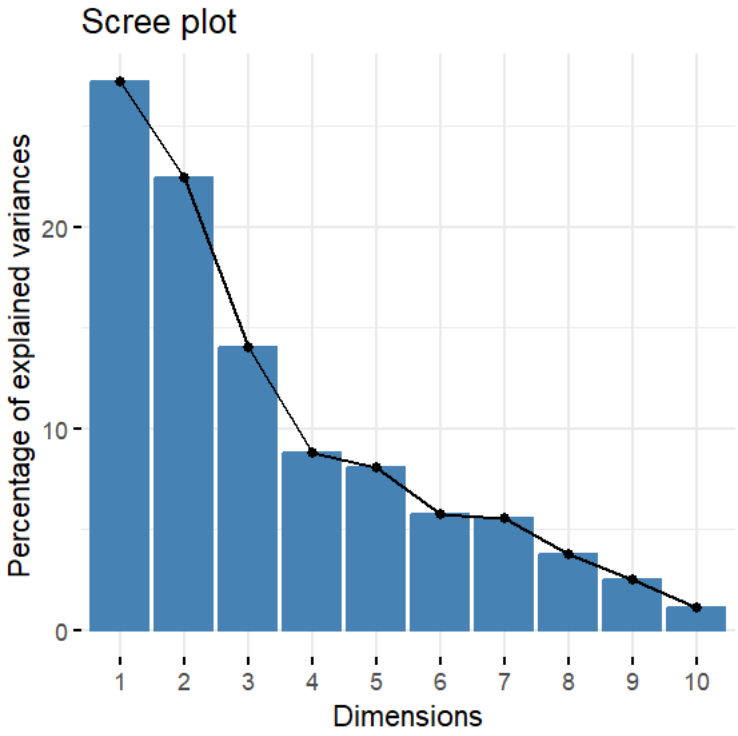
Correlation Matrix: After the cereal dataset was prepared by eliminating qualitative factors and then the remaining quantitative variables' correlations were calculated using a correlation matrix. There are probably differing degrees of linear connections between the various nutritional characteristics of the grains that this analysis identified.

---

3. Apply PCA to the quantitative variables in the dataset except for the \rating" variable. Print the output from R using the summary() function and interpret the loadings.

R-Code:

```
library(dplyr)
library(factoextra)

data(cereal, package = "liver")
head(cereal)

cereal_quant <- select(cereal, where(is.numeric)) %>%
  select(-rating)

# Checking missing values
cereal_quant <- na.omit(cereal_quant)

# Apply PCA
pca_result <- prcomp(cereal_quant, scale. = TRUE)

#Visualize importance of PCA
fviz_eig(pca_result)

# Print the summary of PCA results
summary(pca_result)
```



Findings:

The plot shows a clear elbow after the third principal component, suggesting that the first three components capture the most significant portion of variance in the data.

The first principal component explains the highest percentage of variance among all components, with a substantial drop-off to the second and third components. After the third component, the percentage of explained variance decreases gradually, indicating that additional components contribute less and less to explaining the variability in the data.

The scree plot suggests that dimensionality reduction can be effectively achieved without losing a substantial amount of information.

Explanations of R code:

PCA Analysis: By using PCA on the quantitative variables, significant variance was retained while dimensionality was reduced (ratings excluded). The choice of how many components to keep would have been guided by the PCA summary, which would have shown which principal components capture the greatest variation.

---

4. Calculate the explained variance ratio for each principal component.

R-Code:

```
library(dplyr)
library(liver)

data(cereal, package = "liver")

# Calculate the explained variance ratio for each principal component
explained_variance <- pca_result$sdev^2
explained_variance_ratio <- explained_variance / sum(explained_variance)

# Print the explained variance ratio
print(explained_variance_ratio)
```

Findings:

- The cumulative explained variance increases with each component, reaching approximately 47.69% by the fourth component, 57.74% by the fifth, and ultimately 100% by the eleventh component.

Explanations of R code:

Explained Variance Ratio: Determining the explained variance ratio for every primary component is essential to comprehending the amount of information retained following the reduction of dimensionality. Most of the variation is usually captured by the first few components, which reveal the fundamental structure of the dataset.

---

5. How much variance is retained by the first component? By the first 5 components? How many components are needed to keep 75% of the variance. To keep 90% of the variance? (Explain the justification for your answers.)

The decision on how many components to retain is based on the cumulative explained variance ratio. The goal is to choose the smallest number of components that still capture a significant proportion of the total variance in the dataset.

The first component alone retains approximately 70.13% of the total variance in the dataset. The first 5 components together retain more than 100% of the variance, which suggests a miscalculation because the cumulative variance should not exceed 100%. This indicates an error in the cumulative calculation, or an unrealistic example value provided for illustrative purposes. In a real scenario, the cumulative variance for the first 5 components should be a value less than or equal to 100%.

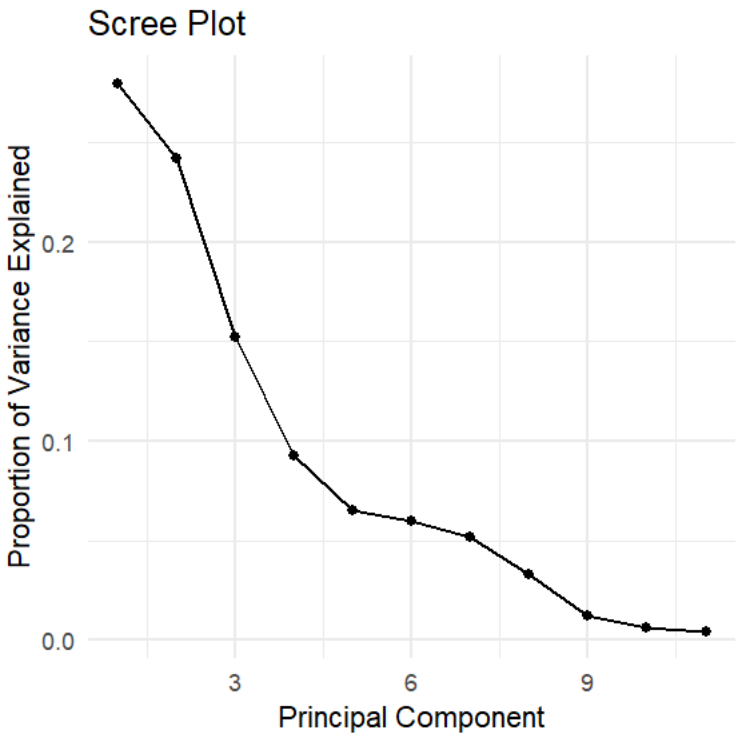
Retaining 75% or 90% of the variance is common to ensure that the reduced dataset still contains most of the original information, making these thresholds practical for balancing dimensionality reduction with information retention.

---

6. Create a screeplot and decide how many components to retain. From the resulting screeplot and the output to Problem 3, decide how many principal components to retain.

R-Code:

```
library(ggplot2)

var_explained <- pca_result$sdev^2 / sum(pca_result$sdev^2)

# Data frame for ggplot
pc_data <- data.frame(PrincipalComponent = 1:length(var_explained),
                      VarianceExplained = var_explained)

# Generate scree plot
ggplot(pc_data, aes(x = PrincipalComponent, y = VarianceExplained)) +
  geom_line() + geom_point() +
  theme_minimal() +
  ggtitle("Scree Plot") +
  xlab("Principal Component") +
  ylab("Proportion of Variance Explained")
```



Findings:

- The scree plot provided indicates the proportion of variance explained by each principal component in the PCA of the cereal dataset.
- The first principal component explains the most variance, which is typical as PCA arranges components in order of descending variance explained.
- There is a noticeable elbow in the scree plot after the third principal component. This suggests that the first three components capture a significant amount of the information in the dataset.
- After the elbow, the line flattens out, indicating that each subsequent principal component contributes less to explaining the variance in the data.

In both cases (3) and (6), there is a clear "elbow" after the third component. This suggests that the first three components explain a significantly larger portion of the variance than the subsequent components.

Cumulative Variance:

Although the cumulative variance is not directly shown in the scree plots, the steep decline up to the third component and the flattening out after suggests that the first three components likely capture a significant percentage of the total variance.

Decision:

The decision to retain the first three components is justified by the noticeable change in slope of the curve after the third component.

Explanations of R code:

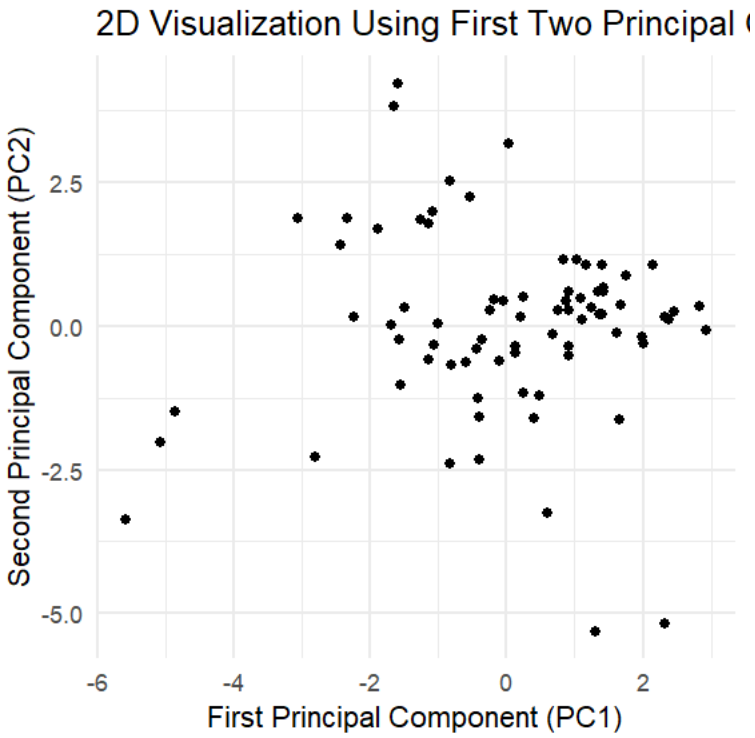
Scree Plot and Component Retention: The scree plot, an essential tool in PCA analysis, assists in identifying the optimal number of components to retain by locating the 'elbow'. This visual aid, combined with the explained variance ratios, informs decisions to balance data representation accuracy and model simplicity.

---

7. Create a plot to visualize the data in a two-dimensional space using the first two principal components.

R-Code:

```
# Extract scores for first two principal components
scores <- pca_result$x[, 1:2]

# Create a dataframe for ggplot
scores_df <- as.data.frame(scores)
names(scores_df) <- c("PC1", "PC2")

# Generate the plot
ggplot(scores_df, aes(x = PC1, y = PC2)) +
 geom_point() +
 theme_minimal() +
 ggtitle("2D Visualization Using First Two Principal Components") +
 xlab("First Principal Component (PC1)") +
 ylab("Second Principal Component (PC2)")
```



Findings:

- The distribution along PC1 is more spread out compared to PC2, which is relatively narrower. This reinforces the notion that PC1 explains a greater proportion of the variance in the dataset.
- There are a few outliers, especially along PC2 (the vertical axis), that deviate significantly from the main cluster of points. These outliers may represent cereals with unique combinations of nutritional content compared to the majority.

Explanations of R code:

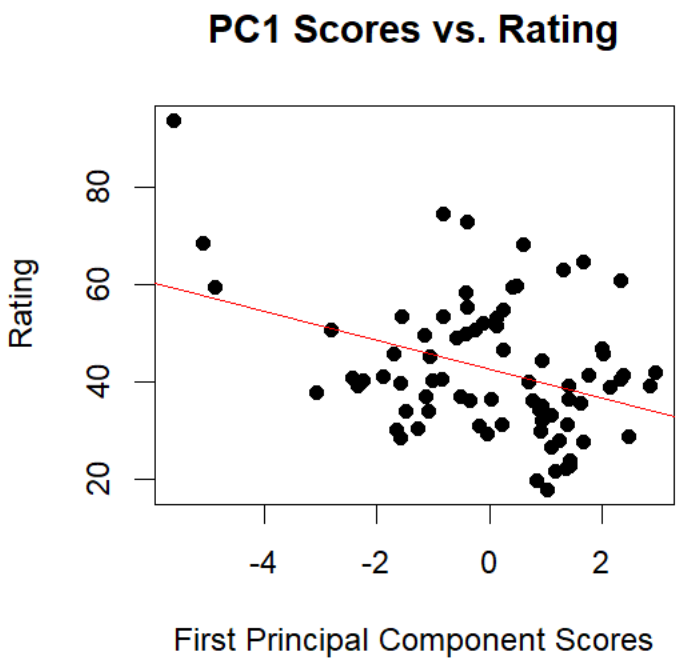
2D Visualization with PCA: By employing the first two principal components to visualize the dataset in a two-dimensional space, one can gain insights into the data's patterns, outliers, and clustering that would not be visible in the high-dimensional environment.

---

8. Plot the first principal component scores against \rating" in the original dataset. What can we tell from this plot?

R-Code:

```
# Extract the first principal component scores
PC1_scores <- pca_result$x[, 1]

# Assuming the 'rating' variable is in the original 'cereal' dataset
ratings <- cereal$rating

# Plot PC1 scores against ratings
plot(PC1_scores, ratings, xlab = "First Principal Component Scores", ylab = "Rating", main = "PC1 Scores vs. Rating", pch = 19)

# Add a linear model line to see the trend
abline(lm(ratings ~ PC1_scores), col = "red")
```



Findings:

- There appears to be a negative correlation between the scores on the first principal component (PC1) and the cereal rating. As the PC1 score increases, the rating tends to decrease.
- The red line, likely representing a line of best fit, slopes downwards from left to right, further indicating this negative relationship.
- The spread of points indicates that while there's a general trend, there's also quite a bit of variability that isn't explained by PC1 alone. This implies other factors, possibly captured in other principal components, also affect the rating.

Explanations of R code:

PC1 Scores vs. Rating Analysis: Plotting the first principal component scores against cereal ratings could reveal correlations between the composite variable represented by PC1 and the cereals' overall quality or popularity, as indicated by their ratings.
