DATA 2010: Group Project

Full Written Report

Arjun Sarkar, Sachin Bhatt, Sahil Sharma, Gustavo Nunez

April 4th 2025

1. Introduction

As a group we collectively decided to focus on the **Travel Reviews** dataset from the UCI Machine Learning Repository for our term project.

2. Tentative Analysis Questions

Here are the following questions we will be answering in our analysis.

- Is it possible to determine distinct traveler preference groups by looking at how they rate various travel categories?
- Are there various traveler groups who share similar preferences?
- Which traveler preferences are most common across all travel categories?

3. Dataset Selection

It essentially consists of many reviews of East Asian places in 10 categories which are in this collection. Our goal is to derive valuable insights from the dataset by analyzing it using statistical methods and data visualization.

Dataset Source: UCI Machine Learning Repository. (2018). Uci.edu. https://archive.ics.uci.edu/dataset/484/travel+reviews

3.1 Dataset Context

```
# loading dataset
travel.data <- read.csv("tripadvisor_review.csv")
head(travel.data,1)
```

```
##   User.ID Category.1 Category.2 Category.3 Category.4 Category.5 Category.6
## 1  User 1       0.93        1.8       2.29       0.62        0.8       2.42
##   Category.7 Category.8 Category.9 Category.10
## 1       3.19       2.79       1.82        2.42
```

The dataset contains 10 Features(excluding **User ID**) and 980 instances. Also, it is important to note that this dataset supports classification and clustering tasks.

The 10 different types of travel categories the travelers gave ratings on are **Art Galleries, Dance Clubs, Juice Bars, Restaurants, Museums, Resorts, Parks/Picnic Spots, Beaches, Theaters, and Religious Institutions**

The following is a mapping of each traveler rating:

- Excellent (4), Very Good (3), Average (2), Poor (1), Terrible (0)

4. Cleaning Up Dataset

In this section, we will check for missing values in the dataset. Missing values can lead to incorrect analysis results, so it's crucial to address them early on.

4.1 Checking for Missing Values

We'll use the `is.na()` function to check for missing values, and `colSums()` to sum up the number of missing values in each column.

```r
# checking for missing values in each column (if any)
# sum of missing values per column
colSums(is.na(travel.data))
```

```
##     User.ID  Category.1  Category.2  Category.3  Category.4  Category.5
##           0           0           0           0           0           0
##  Category.6  Category.7  Category.8  Category.9 Category.10
##           0           0           0           0           0
```

4.2 Duplicate Rows

Since `User.ID` is unique for each row, we don't need to worry about duplicates. Therefore, checking for duplicate rows might not be a high priority but it is still a good idea to verify. However, we do bring this up as it can distort results, especially when clustering or other statistical tests.

```r
# checking how many duplicate rows there are
sum(duplicated(travel.data))
```

```
## [1] 0
```

4.3 Replacing Column Names

We figured to introduce more meaningful column names representing each location other than having it as `Category 1, 2 ...` giving it a more clean and polished look.

```r
# destination names
new_column_names <- c("Art Galleries", "Dance Clubs", "Juice Bars",
                      "Restaurants","Museums", "Resorts", "Parks/Picnic Spots",
                      "Beaches", "Theaters", "Religious Institutions")

# reserving first column as it's `User.ID`
# replace the rest of column names with the destination names
colnames(travel.data)[-1] <- new_column_names

# storing it in another variable
travelUpdate.data <- travel.data
head(travelUpdate.data,1)
```

```
##   User.ID Art Galleries Dance Clubs Juice Bars Restaurants Museums Resorts
## 1  User 1          0.93         1.8       2.29        0.62     0.8    2.42
##   Parks/Picnic Spots Beaches Theaters Religious Institutions
## 1               3.19    2.79     1.82                   2.42
```

5. Data Summarization

5.1 Loading Libraries

Prior to conducting our analysis efficiently, we load essential R libraries for data manipulation, visualization, correlation analysis, and clustering.

```r
suppressPackageStartupMessages(library(dplyr))      # Data manipulation
library(tidyr)                                      # Data transformation
library(ggplot2)                                    # Data visualization
suppressPackageStartupMessages(library(corrplot))   # Correlation analysis
suppressPackageStartupMessages(library(factoextra)) # Clustering analysis
library(cluster)                                    # Clustering algorithms
suppressPackageStartupMessages(library(dendextend)) # Hierarchical Analysis (Dendograms)
```

5.2 Summary Statistics

We calculate summary statistics, such as mean, median, standard deviation, and the five-number summary (minimum, Q1, median, Q3, maximum) for every travel category in order to obtain a preliminary comprehension of the dataset.

```r
# summary statistics for all locations
summary.stats <- summary(travelUpdate.data[, -1])
summary.stats
```

```
##  Art Galleries     Dance Clubs      Juice Bars     Restaurants
##  Min.   :0.3400   Min.   :0.000   Min.   :0.130   Min.   :0.1500
##  1st Qu.:0.6700   1st Qu.:1.080   1st Qu.:0.270   1st Qu.:0.4100
##  Median :0.8300   Median :1.280   Median :0.820   Median :0.5000
##  Mean   :0.8932   Mean   :1.353   Mean   :1.013   Mean   :0.5325
##  3rd Qu.:1.0200   3rd Qu.:1.560   3rd Qu.:1.573   3rd Qu.:0.5800
##  Max.   :3.2200   Max.   :3.640   Max.   :3.620   Max.   :3.4400
```

```
##     Museums          Resorts      Parks/Picnic Spots    Beaches
##  Min.   :0.0600   Min.   :0.140   Min.   :3.160      Min.   :2.420
##  1st Qu.:0.6400   1st Qu.:1.460   1st Qu.:3.180      1st Qu.:2.740
##  Median :0.9000   Median :1.800   Median :3.180      Median :2.820
##  Mean   :0.9397   Mean   :1.843   Mean   :3.181      Mean   :2.835
##  3rd Qu.:1.2000   3rd Qu.:2.200   3rd Qu.:3.180      3rd Qu.:2.910
##  Max.   :3.3000   Max.   :3.760   Max.   :3.210      Max.   :3.390
##     Theaters     Religious Institutions
##  Min.   :0.740   Min.   :2.140
##  1st Qu.:1.310   1st Qu.:2.540
##  Median :1.540   Median :2.780
##  Mean   :1.569   Mean   :2.799
##  3rd Qu.:1.760   3rd Qu.:3.040
##  Max.   :3.170   Max.   :3.660
```

Let's present the standard deviations.

```r
# standard deviations for all categories
sapply(travelUpdate.data[, -1], sd)
```

```
##          Art Galleries            Dance Clubs             Juice Bars
##            0.326912231            0.478280151            0.788606876
##            Restaurants                Museums                Resorts
##            0.279731330            0.437429966            0.539538040
##     Parks/Picnic Spots                Beaches               Theaters
##            0.007824448            0.137505488            0.364629454
## Religious Institutions
##            0.321379831
```

Highly Rated Categories

- `Parks/Picnic Spots` have the highest average rating (3.18) with low variation (SD = 0.0078), indicating consistent positive traveler satisfaction. `Beaches` (Mean = 2.83) and `Religious Institutions` (Mean = 2.80) also receive high ratings, suggesting positive traveler experiences

Moderate Ratings & Mixed Opinions

- `Resorts` (Mean = 1.84, SD = 0.54) and `Theaters` (Mean = 1.57, SD = 0.36) show moderate ratings with some variation, indicating diverse traveler preferences. `Dance Clubs`(Mean = 1.35, SD = 0.48) display significant spread, hinting at conflicting experiences

Lower-Rated Categories

- `Restaurants` have the lowest average rating (0.53) and low variation, suggesting overall dissatisfaction among travelers. `Juice Bars` (Mean = 1.01, SD = 0.79) show high variability, meaning some travelers enjoyed them while others had poor experiences

Category-Specific Trends

- `Art Galleries` and `Museums` have similar low ratings ($\approx 0.89 - 0.94$) with slightly lower variability, implying generally unfavorable experiences. `Dance Clubs` have a broad spread (SD = 0.48), possibly due to differences in expectations or quality between locations.

Summary

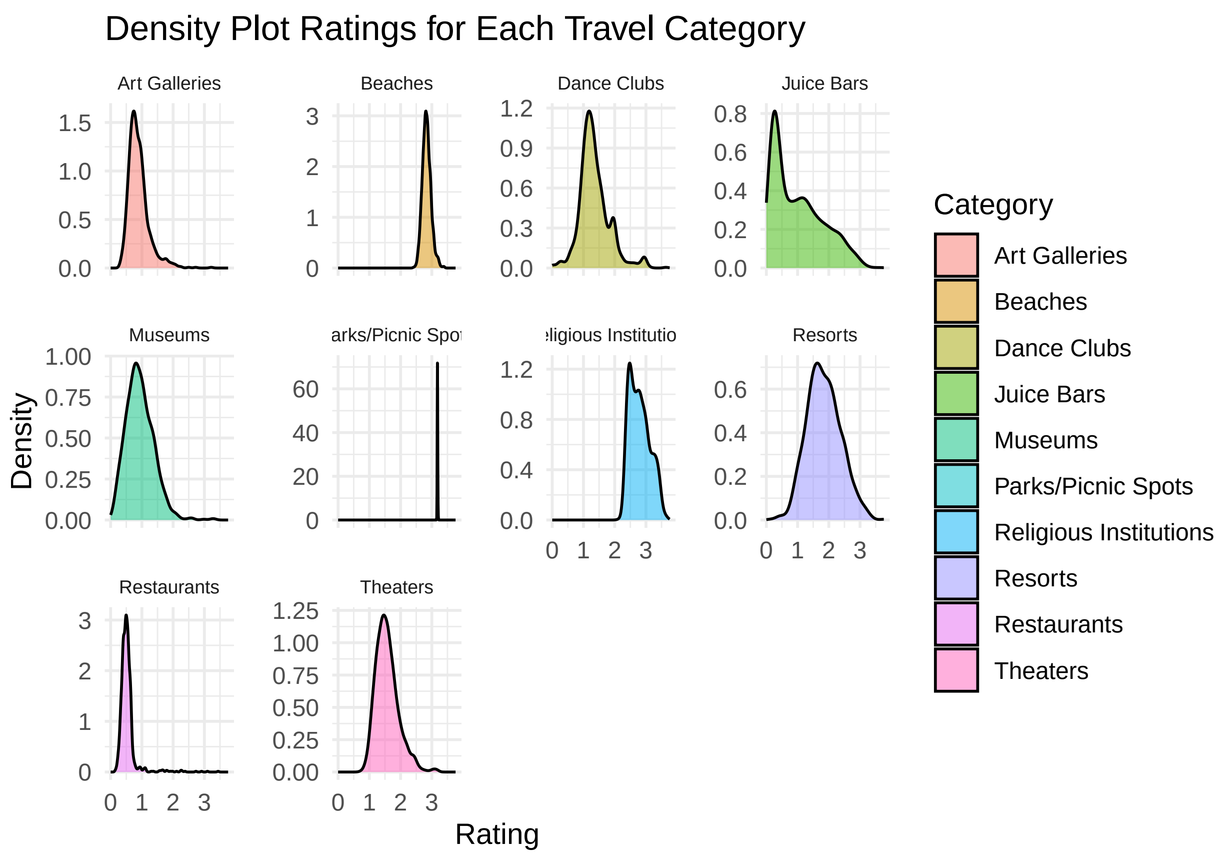
- The data suggests that outdoor locations (`Parks`, `Beaches`, `Religious Institutions`) receive higher and more consistent ratings, while urban entertainment spots (`Dance Clubs`, `Theaters`, `Resorts`) show mixed traveler opinions. Restaurants and Juice Bars appear to be less favored, with notable dissatisfaction among travelers.

5.3 Data Distribution Analysis

5.31 Density Plots for Individual Travel Categories

For each category, we create a distinct density plot to show the rating distribution. This method avoids clutter and makes insights clearer.

```
# Converting `travelUpdate.data` into long data format for ggplot visualization
# excluding the `User.ID` column
long_data <- pivot_longer(travelUpdate.data, cols = c(-User.ID),
                          names_to = "Category", values_to = "Rating")

# Density plot for each category (separate plots)
ggplot(long_data, aes(x = Rating, fill = Category)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot Ratings for Each Travel Category",
       x = "Rating", y = "Density") + theme_minimal() +
  facet_wrap(~ Category, scales = "free_y") + # Creates separate density plots
  theme(strip.text = element_text(size = 7),  # Increase facet label font size
        panel.spacing = unit(1, "lines"))     # Increase space between facets
```



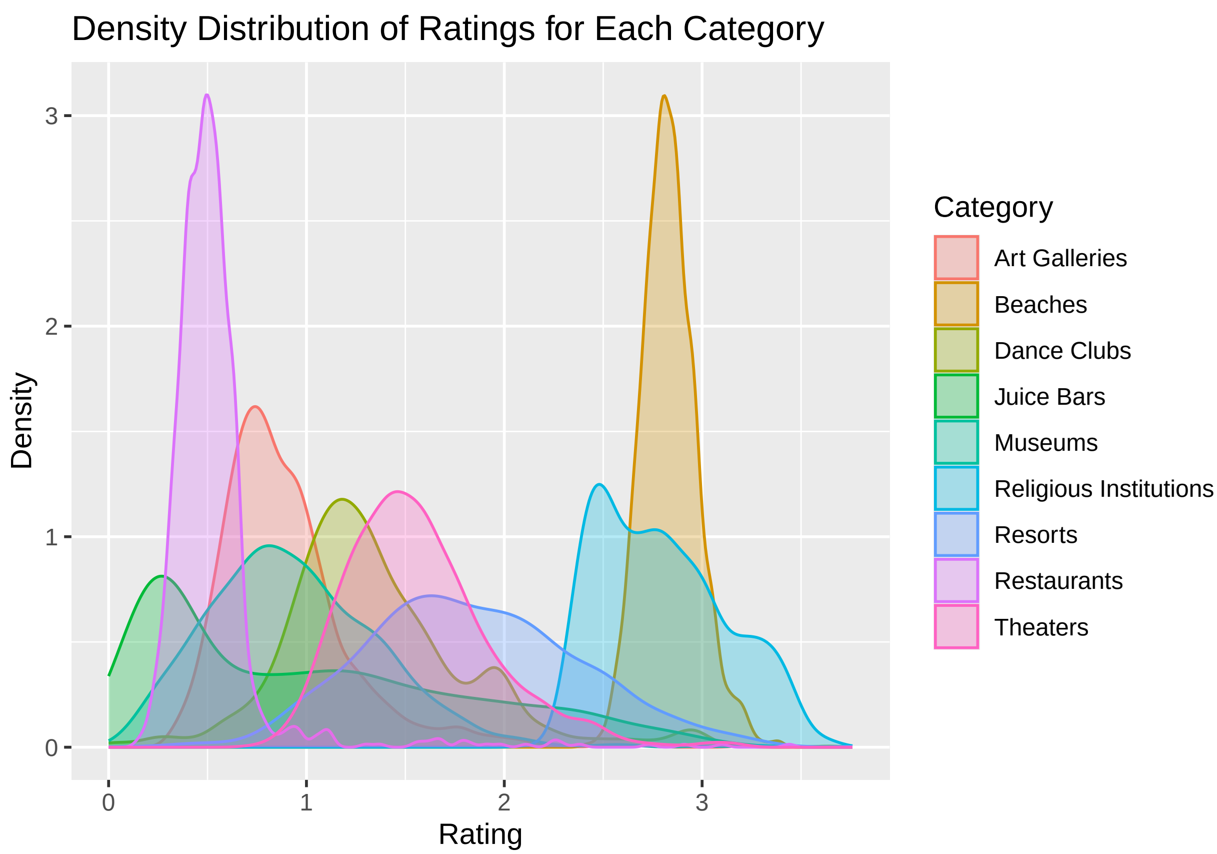
Insights

- Some locations have wider curves that reflect a range of traveler perspectives, others have narrow peaks that indicate consistent ratings. `Dance Clubs` have shown a little bit of bimodal distribution, suggesting conflicting opinions where some travelers loved them, while others had negative experiences.

5.32 Combined Density Plot

A combined density plot allows us to compare rating distributions across different categories all in one place. However, `Parks/Picnic Spots` had an extreme peak, so we exclude it for better visualization.

```r
# Density Plot for each category (all together excluding `Parks/Picnic Spots`)
long_data.temp <- pivot_longer(travelUpdate.data,
                               cols = c(-User.ID,-"Parks/Picnic Spots"),
                               names_to = "Category", values_to = "Rating")

ggplot(long_data.temp, aes(x = Rating, fill = Category, color = Category)) +
  geom_density(alpha = 0.3) +
  labs(title = "Density Distribution of Ratings for Each Category",
       x = "Rating", y = "Density")
```



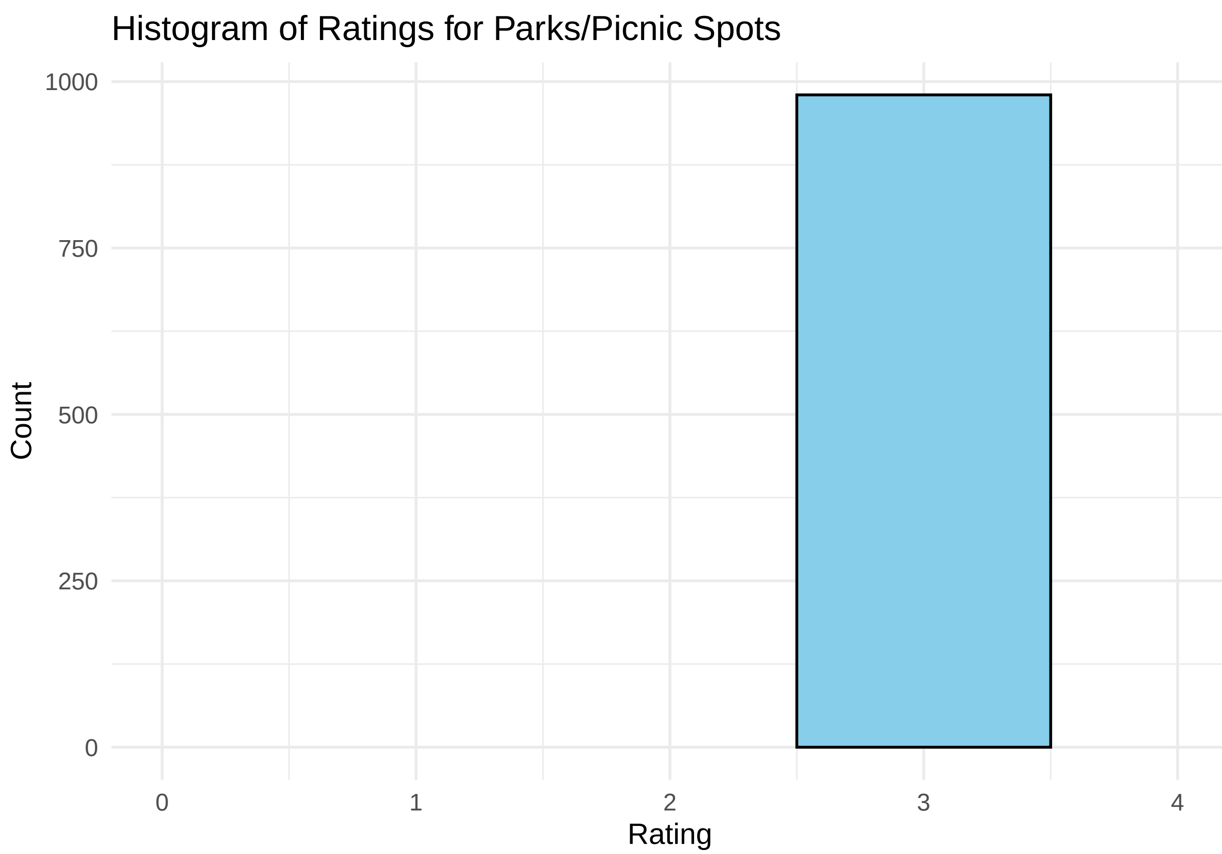
Insights

- `Restaurants` show low ratings, indicating potential dissatisfaction. `Beaches` have higher peaks around rating 3, suggesting positive traveler experiences. `Religious Institutions`, `Resorts`, and `Museums` display wider curves, reflecting diverse traveler opinions. As stated earlier, `Dance Clubs` present a bimodal trend, clarifying the inconsistent user experience.

5.33 Investigating Parks/Picnic Spots

Since `Parks/Picnic Spots` displayed an extreme peak, we analyze its rating distribution separately using a histogram.

```
ggplot(long_data %>%
         filter(Category == "Parks/Picnic Spots", !is.na(Rating)),
       aes(x = Rating)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
  labs(title = "Histogram of Ratings for Parks/Picnic Spots",
       x = "Rating", y = "Count") +
  coord_cartesian(xlim = c(0, 4)) +
  theme_minimal()
```



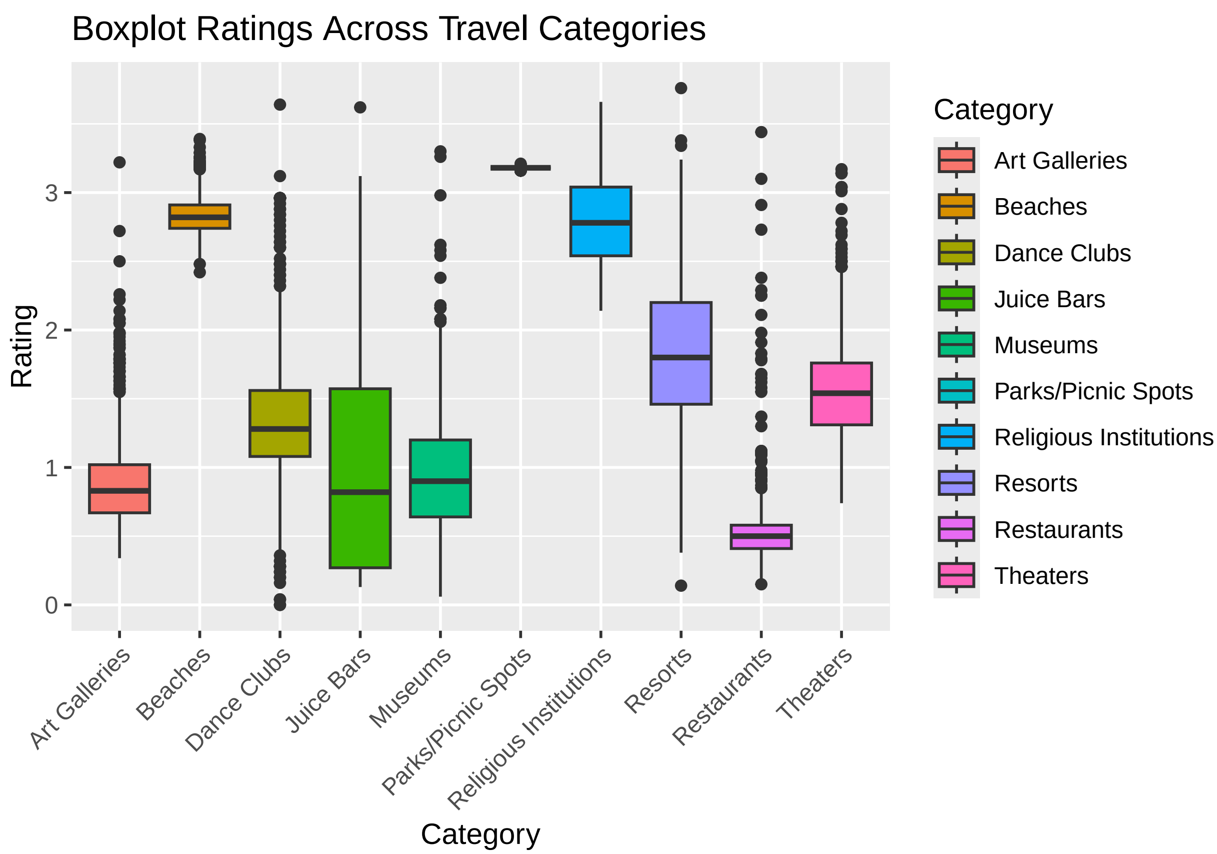
Insights

- The histogram shows that almost all 980 ratings fall between 2.5 and 3.5, forming a single, dominant peak. This suggests that travelers consistently rate `Parks/Picnic Spots` favorably, with low variation in opinion based on results from Summary Statistics section.

5.34 Boxplot Analysis - Comparing Ratings Across Categories

Hopefully, the boxplots provide a comparative visualization of ratings, highlighting central tendencies, spread, and outliers for our needs.

```
ggplot(long_data, aes(x = Category, y = Rating, fill = Category)) +
  theme(axis.text.x= element_text(angle = 45, hjust = 1)) + geom_boxplot() +
  labs(title = "Boxplot Ratings Across Travel Categories")
```



Insights

Central Tendencies

- `Parks/Picnic Spots` have the highest median rating (~3.0), followed by `Beaches` and `Religious Institutions`.`Resorts` and `Theaters` have lower median ratings (~2.0), indicating lower traveler preference. `Restaurants` have the lowest median, suggesting they are the least favored destination.

Spread of Ratings

- `Juice Bars`, `Resorts`, `Museums`, `Religious Institutions`, and `Dance Clubs` exhibit large interquartile ranges (IQRs), indicating high variability in traveler ratings. `Parks/Picnic Spots` and `Beaches` have small IQRs, suggesting consistent traveler experiences.

Outliers

- `Restaurants`, `Dance Clubs`, `Art Galleries`, and `Theaters` show numerous outliers, indicating that while most travelers rated them within a certain range, some gave extreme ratings. Although `Restaurants` have the lowest median, on the contrary, a moderate amount of users rate it highly as due to better experiences as displayed on the boxplot. `Dance Clubs` display many low outliers, further supporting the contrasting user experience observed in the Density Plots for Individual Travel Categories section.

5.4 Data Summarization Key Findings

Overall Rating Trends

- `Parks/Picnic Spots` have high, consistent ratings. `Beaches`, `Religious Institutions`, and `Resorts` receive moderately high ratings, but opinions vary. `Restaurants` and `Theaters` are less preferred, with lower median ratings.

Traveler Preferences & Ratings

- Locations such as `Juice Bars` and `Museums` have large variations, indicating diverse user experiences. `Dance Clubs` have bimodal ratings, suggesting mixed traveler opinions.

Unexpected Trends & Outliers

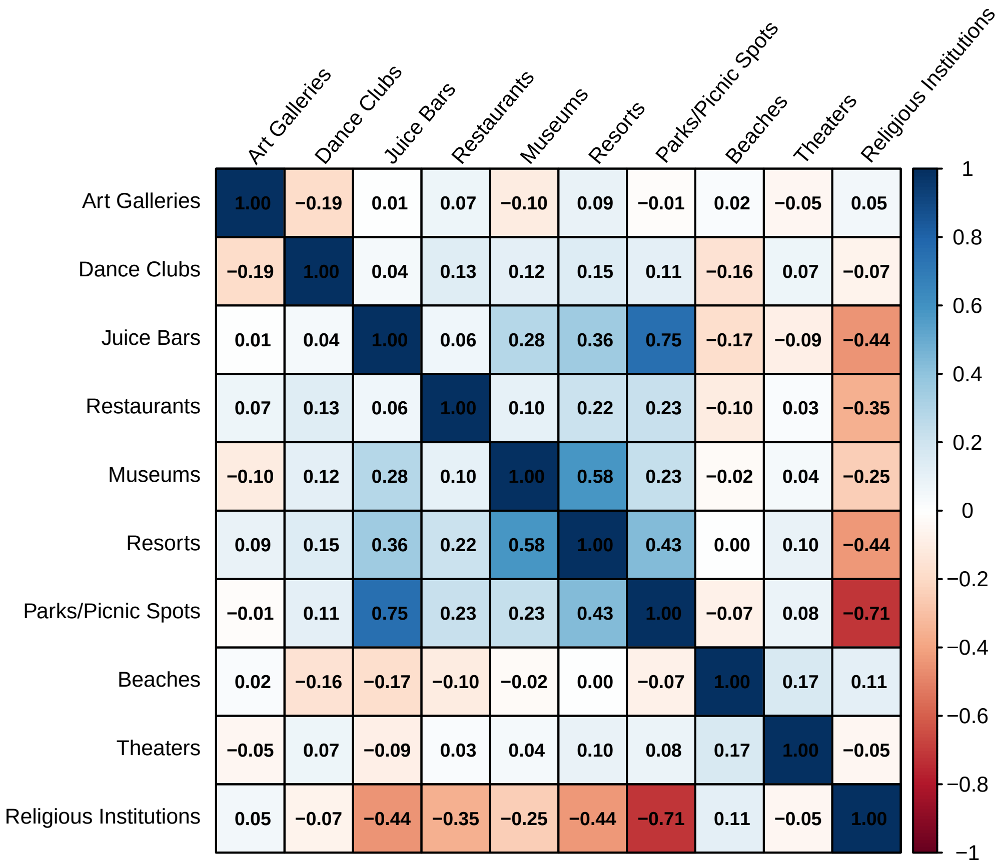
- `Dance Clubs` show many low outliers, indicating conflicting experiences. `Parks/Picnic Spots` have extremely concentrated ratings, making them an outlier in terms of consistency.

6. Correlation Analysis

Finding trends in traveler preferences requires an understanding of the connections between various travel categories. By identifying clusters of related interests or divergent preferences, correlation analysis assists us in determining whether particular categories have a tendency to be ranked similarly. Whereas a strong negative correlation denotes an ability for preferring one category over another, a strong positive correlation suggests that tourists who like one kind of place may also like another.

6.1 Correlation Matrix

```r
# computing correlation matrix (excluding User ID)
correlation.map <- cor(travelUpdate.data[, -1])

# displaying correlation heatmap
corrplot(correlation.map,
         addCoef.col = "black",   # Adding black coefficients
         cl.cex = 0.7,            # Reduce legend font size
         method = "color",
         type = "full",           # Display upper half
         tl.col = "black",        # Black text for labels
         tl.srt = 50,             # Rotate labels 45°
         number.cex = 0.6,        # Reduce coefficient font size
         tl.cex = 0.7,            # Reduce variable label size
         diag = TRUE,             # Hide diagonal
         addgrid.col = "black")   # Light gray borders
```



Insights

Strong Positive Correlations

- `Juice Bars` and `Parks/Picnic Spots` have the strongest positive correlation (0.75). `Museums` and `Resorts` are also positively correlated (0.58)

Strong Negative Correlations

- `Religious Institutions` and `Parks/Picnic Spots` show a strong negative correlation (-0.71). `Religious Institutions` and `Juice Bars` also have a notable negative correlation (-0.44)

Weaker Correlations

- Many locations, such as `Art Galleries` and `Dance Clubs`, show weak correlations (-0.19)

Overall

- Locations like `Parks/Picnic Spots`, `Juice Bars`, and `Museums` demonstrate stronger relationships with others, whereas `Religious Institutions` tend to show weaker or negative correlations. The plot will be highly useful for identifying clusters in the sections later on.

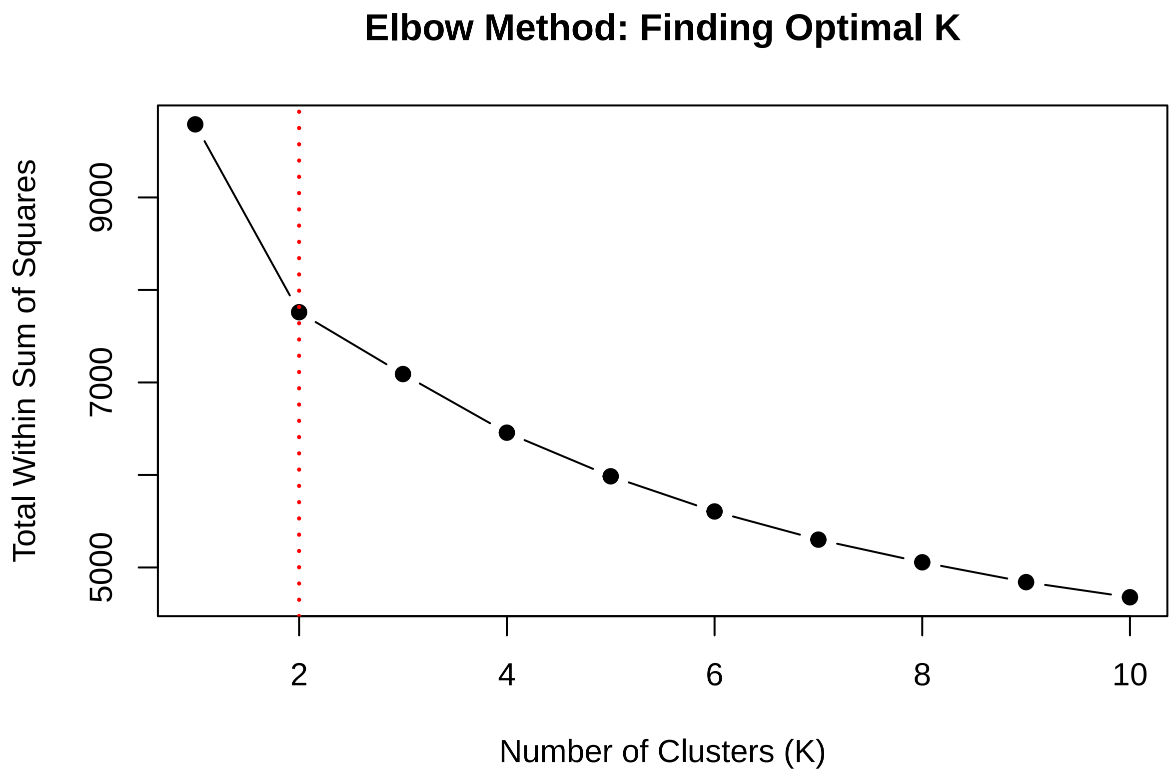
7. Cluster Analysis

7.1 K-Means Clustering: Finding the Optimal Number of Clusters

We use the Elbow Method to determine the optimal number of clusters (K). The idea is to identify the point where adding more clusters does not significantly decrease within-cluster variance.

The K-Means algorithm is used to group travelers based on similarity in their ratings. We determine the optimal number of clusters using the Elbow Method. We have `compute_ss` function which calculates the total within-cluster sum of squares for a given number of clusters, K, and outputs a representation of how tightly grouped the data points are within their respective clusters.

```
# Should consist list of all 10 categories
clustering.data <- travelUpdate.data[, -1]

# scaling the data (standardizing our ratings)
clustering.data <- scale(clustering.data)

# ensuring reproducibility
set.seed(123)

compute_ss = function(k) {
  return( kmeans(clustering.data, centers = k, nstart = 25)$tot.withinss )
}

# Compute the total within-cluster sum of squares for different K values
wss <- sapply(X=1:10, FUN=compute_ss)

# Plotting the Elbow Method
plot(1:10, wss, type = "b", xlim = c(1,10),
     pch = 19, frame = TRUE, main = "Elbow Method: Finding Optimal K",
     xlab = "Number of Clusters (K)", ylab = "Total Within Sum of Squares")

# Adding a dotted vertical line at K=2
abline(v = 2, lty = 3, col = "red", lwd = 2)
```



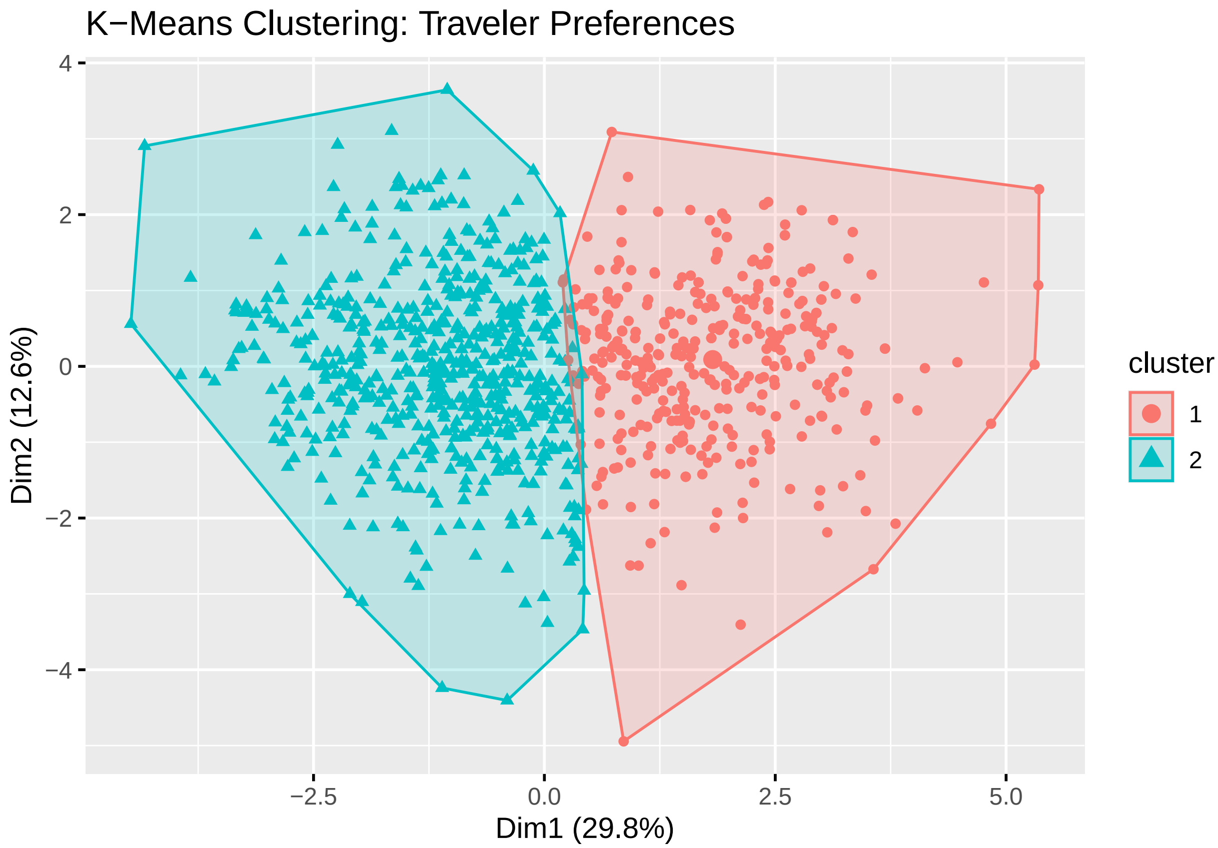
Insights

- Based on the graph, we observe that the `elbow point` occurs at `K = 2` (shown by dotted red line), where the within-cluster sum of squares stops decreasing significantly, suggesting that this is the optimal K value for our clusters.

7.2 K-Means Clustering: Grouping Travelers

Using K = 2, we apply K-Means clustering to group travelers based on their preferences.

```
# applying K-Means Clustering
k.optimal <- 2
kmeans.result <- kmeans(clustering.data, centers = k.optimal, nstart = 30)

# adding cluster labels to our `travelUpdate.data` dataset
travelUpdate.data$Cluster <- as.factor(kmeans.result$cluster)

# Visualizing clusters using PCA
fviz_cluster(kmeans.result, data = clustering.data,
             main = "K-Means Clustering: Traveler Preferences",
             ellipse.type = "convex", geom = "point")
```



Insights

Optimal Number of Clusters

- The visualization shows two distinct traveler groups, suggesting different preferences among travelers!

Summary

- The plot is a visual summary of your clustering results, indicating that there are distinct groups of travelers (each represented by a color) with similar rating profiles.

7.3 Analysing Cluster Centers

In order to understand what differentiates between the two clusters, we analyze the cluster centers by averaging rating per travel category within each group.

```
# viewing cluster centers to see which categories drive the differences
kmeans.result$centers
```

```
##   Art Galleries Dance Clubs Juice Bars Restaurants    Museums    Resorts
## 1    0.05206298   0.2520339  0.8687001   0.4211929  0.6378972  0.7535470
## 2   -0.03213128  -0.1555457 -0.5361284  -0.2599441 -0.3936857 -0.4650604
##   Parks/Picnic Spots    Beaches    Theaters Religious Institutions
## 1          0.8334284 -0.2248408 -0.02757251             -0.7989443
## 2         -0.5143601  0.1387631  0.01701670              0.4930779
```

Insights

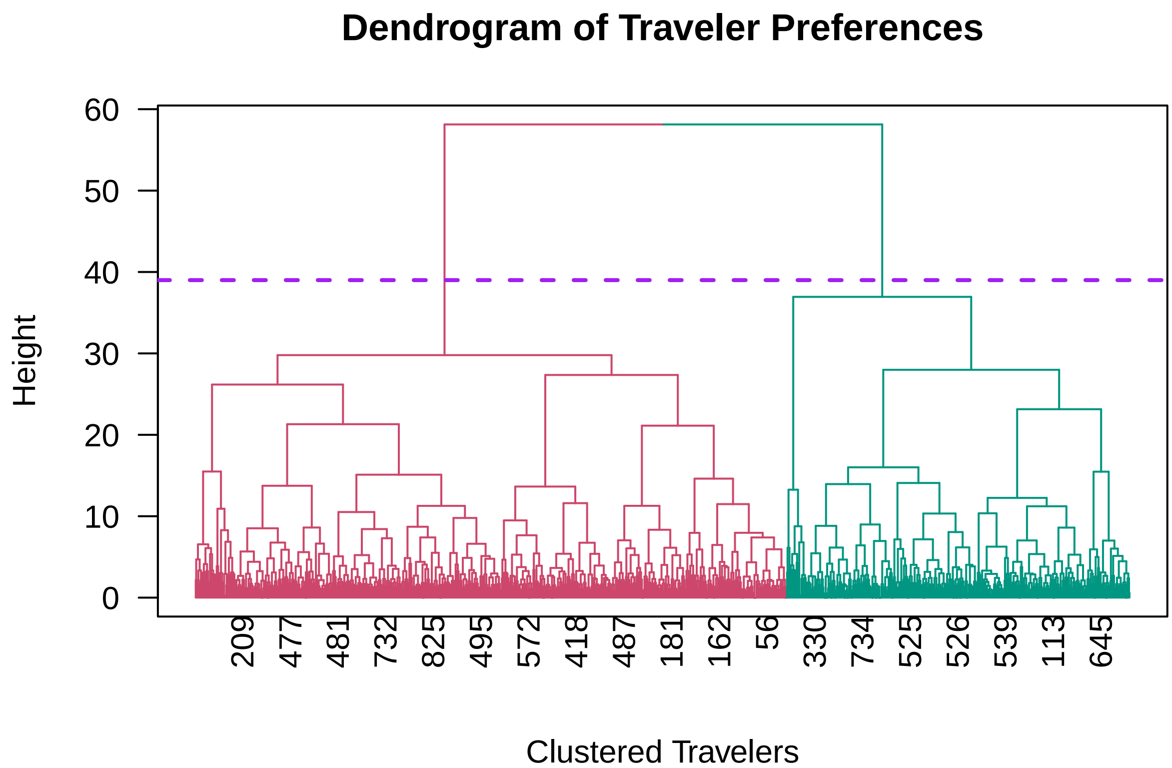
Cluster Centers

- Cluster 1 (Higher Ratings Group) prefers outdoor destinations like `Parks/Picnic Spots`, `Juice Bars`, and `Resorts`. `Parks/Picnic Spots` & `Juice Bars` are strongly preferred by Cluster 1 but less preferred by Cluster 2. Cluster 2 (Lower Ratings Group) rates `Religious Institutions`, `Beaches`, and `Theaters` higher than Cluster 1.

7.4 Hierarchical Clustering: Dendrogram Visualization

To confirm and clarify our clustering analysis, thus, we apply Hierarchical Clustering.

```
# computing distance matrix
travel.dist <- dist(clustering.data, method = "euclidean")

# performing hierarchical clustering using ward's method
hc <- hclust(travel.dist, method = "ward.D2")

# converting to dendrogram object for better visualization
dend <- as.dendrogram(hc)

# separating the dendrogram at k clusters and assigning colors
dend <- color_branches(dend, k = k.optimal)

# modifying x-axis labels to show every 50th traveler
dend_labels <- labels(dend)
dend_labels <- ifelse(1:length(dend_labels) %% 50 == 0, dend_labels, "")
labels(dend) <- dend_labels

# plotting the Dendrogram
plot(dend, main = "Dendrogram of Traveler Preferences",
     xlab = "Clustered Travelers",ylab = "Height",
     frame.plot = TRUE, las = 2, cex = 0.7)

# adding the horizontal cut-off line at appropriate height
abline(h = 39, col = "purple", lwd = 2, lty = 2)
```



Insights

- The hierarchical dendrogram confirms the presence of two major traveler groups, further establishing our K-Means findings. The dotted "purple" cutoff line in the plot serves as a threshold to define the number of clusters in the dendrogram which we can verify is 2. Our clustering analysis provides key insights into traveler preferences and addressing our tentative analysis inquiries. Let's look at the next section for our overview of our analysis.

8. Conclusion

Q1: Is it possible to determine distinct traveler preference groups?

- Yes, both K-Means and Hierarchical Clustering confirm the presence of two primary traveler groups!

Q2: Are there different traveler groups who share similar preferences?

- Indeed. Cluster 1 prefers outdoor and social experiences (`Parks`, `Juice Bars`, `Resorts`). Cluster 2 is more inclined towards cultural and religious locations (`Religious Institutions`, `Beaches`, `Theaters`)

Q3: Which traveler preferences are most common across all travel categories?

Most Common Preferences

- `Juice Bars` & `Parks/Picnic Spots` are highly correlated (+0.75), meaning travelers who like one tend to like the other. `Religious Institutions` compared with outdoor locations such as `Parks/Picnic Spots`, `Juice Bars`, `Restaurants`, `Museums`, and `Resorts` tend to show negative correlation, suggesting that users that prefer religious places don't like outdoor locations.

9. Appendix: All R Code

9.1 Dataset Context

```r
# loading dataset
travel.data <- read.csv("tripadvisor_review.csv")
head(travel.data,1)
```

```
##   User.ID Category.1 Category.2 Category.3 Category.4 Category.5 Category.6
## 1  User 1       0.93        1.8       2.29       0.62        0.8       2.42
##   Category.7 Category.8 Category.9 Category.10
## 1       3.19       2.79       1.82        2.42
```

9.2 Checking for Missing Values

```r
# checking for missing values in each column (if any)
# sum of missing values per column
colSums(is.na(travel.data))
```

```
##     User.ID  Category.1  Category.2  Category.3  Category.4  Category.5
##           0           0           0           0           0           0
##  Category.6  Category.7  Category.8  Category.9 Category.10
##           0           0           0           0           0
```

9.3 Duplicate Rows

```r
# checking how many duplicate rows there are
sum(duplicated(travel.data))
```

```
## [1] 0
```

9.4 Replacing Column Names

```r
# destination names
new_column_names <- c("Art Galleries", "Dance Clubs", "Juice Bars",
                      "Restaurants","Museums", "Resorts", "Parks/Picnic Spots",
                      "Beaches", "Theaters", "Religious Institutions")

# reserving first column as it's `User.ID`
```

```
# replace the rest of column names with the destination names
colnames(travel.data)[-1] <- new_column_names

# storing it in another variable
travelUpdate.data <- travel.data
head(travelUpdate.data,1)
```

```
##   User.ID Art Galleries Dance Clubs Juice Bars Restaurants Museums Resorts
## 1  User 1          0.93         1.8       2.29        0.62     0.8    2.42
##   Parks/Picnic Spots Beaches Theaters Religious Institutions
## 1               3.19    2.79     1.82                   2.42
```

9.5 Loading Libraries

```
suppressPackageStartupMessages(library(dplyr))      # Data manipulation
library(tidyr)                                       # Data transformation
library(ggplot2)                                     # Data visualization
suppressPackageStartupMessages(library(corrplot))   # Correlation analysis
suppressPackageStartupMessages(library(factoextra)) # Clustering analysis
library(cluster)                                     # Clustering algorithms
suppressPackageStartupMessages(library(dendextend)) # Hierarchical Analysis (Dendograms)
```

9.6 Summary Statistics

```
# summary statistics for all destination places
summary.stats <- summary(travelUpdate.data[, -1])
summary.stats
```

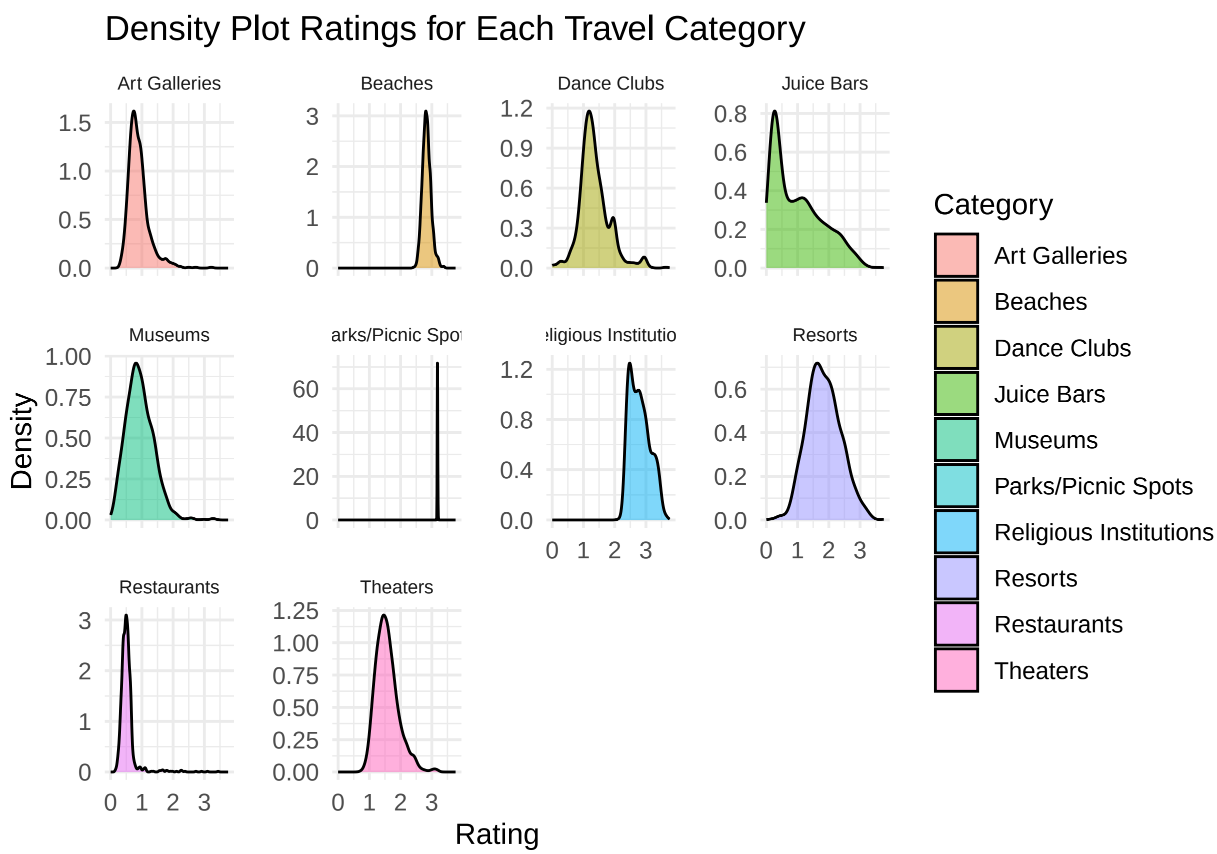
```
##  Art Galleries     Dance Clubs      Juice Bars     Restaurants
##  Min.   :0.3400   Min.   :0.000   Min.   :0.130   Min.   :0.1500
##  1st Qu.:0.6700   1st Qu.:1.080   1st Qu.:0.270   1st Qu.:0.4100
##  Median :0.8300   Median :1.280   Median :0.820   Median :0.5000
##  Mean   :0.8932   Mean   :1.353   Mean   :1.013   Mean   :0.5325
##  3rd Qu.:1.0200   3rd Qu.:1.560   3rd Qu.:1.573   3rd Qu.:0.5800
##  Max.   :3.2200   Max.   :3.640   Max.   :3.620   Max.   :3.4400
##     Museums          Resorts      Parks/Picnic Spots    Beaches
##  Min.   :0.0600   Min.   :0.140   Min.   :3.160      Min.   :2.420
##  1st Qu.:0.6400   1st Qu.:1.460   1st Qu.:3.180      1st Qu.:2.740
##  Median :0.9000   Median :1.800   Median :3.180      Median :2.820
##  Mean   :0.9397   Mean   :1.843   Mean   :3.181      Mean   :2.835
##  3rd Qu.:1.2000   3rd Qu.:2.200   3rd Qu.:3.180      3rd Qu.:2.910
##  Max.   :3.3000   Max.   :3.760   Max.   :3.210      Max.   :3.390
##     Theaters     Religious Institutions
##  Min.   :0.740   Min.   :2.140
##  1st Qu.:1.310   1st Qu.:2.540
##  Median :1.540   Median :2.780
##  Mean   :1.569   Mean   :2.799
##  3rd Qu.:1.760   3rd Qu.:3.040
##  Max.   :3.170   Max.   :3.660
```

```r
# standard deviations for all categories
sapply(travelUpdate.data[, -1], sd)
```

```
##          Art Galleries            Dance Clubs             Juice Bars
##            0.326912231            0.478280151            0.788606876
##            Restaurants                Museums                Resorts
##            0.279731330            0.437429966            0.539538040
##     Parks/Picnic Spots                Beaches               Theaters
##            0.007824448            0.137505488            0.364629454
## Religious Institutions
##            0.321379831
```

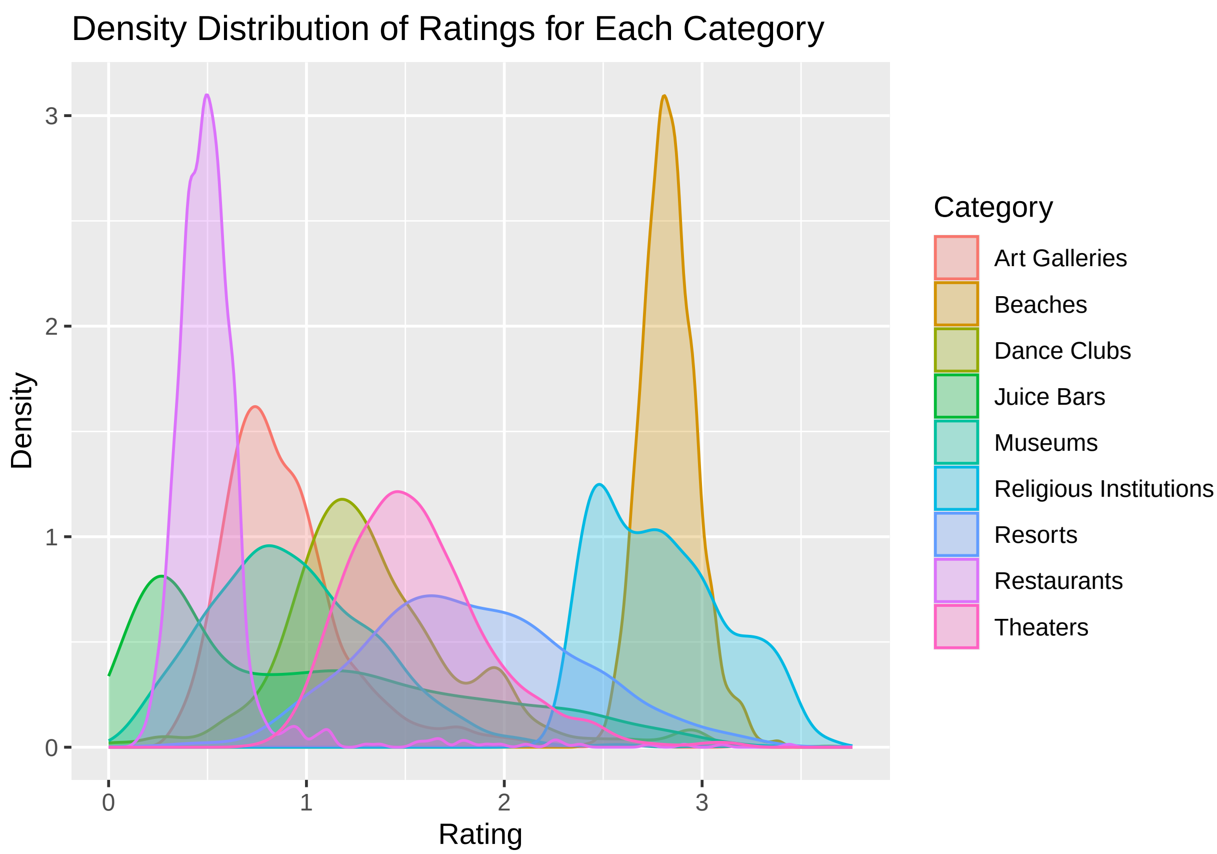
9.7 Density Plots for Individual Travel Categories

```r
# Converting `travelUpdate.data` into long data format for ggplot visualization
# excluding the `User.ID` column
long_data <- pivot_longer(travelUpdate.data, cols = c(-User.ID),
                          names_to = "Category", values_to = "Rating")

# Density plot for each category (separate plots)
ggplot(long_data, aes(x = Rating, fill = Category)) +
  geom_density(alpha = 0.5) +
  labs(title = "Density Plot Ratings for Each Travel Category",
       x = "Rating", y = "Density") + theme_minimal() +
  facet_wrap(~ Category, scales = "free_y") + # Creates separate density plots
  theme(strip.text = element_text(size = 7),  # Increase facet label font size
        panel.spacing = unit(1, "lines"))     # Increase space between facets
```



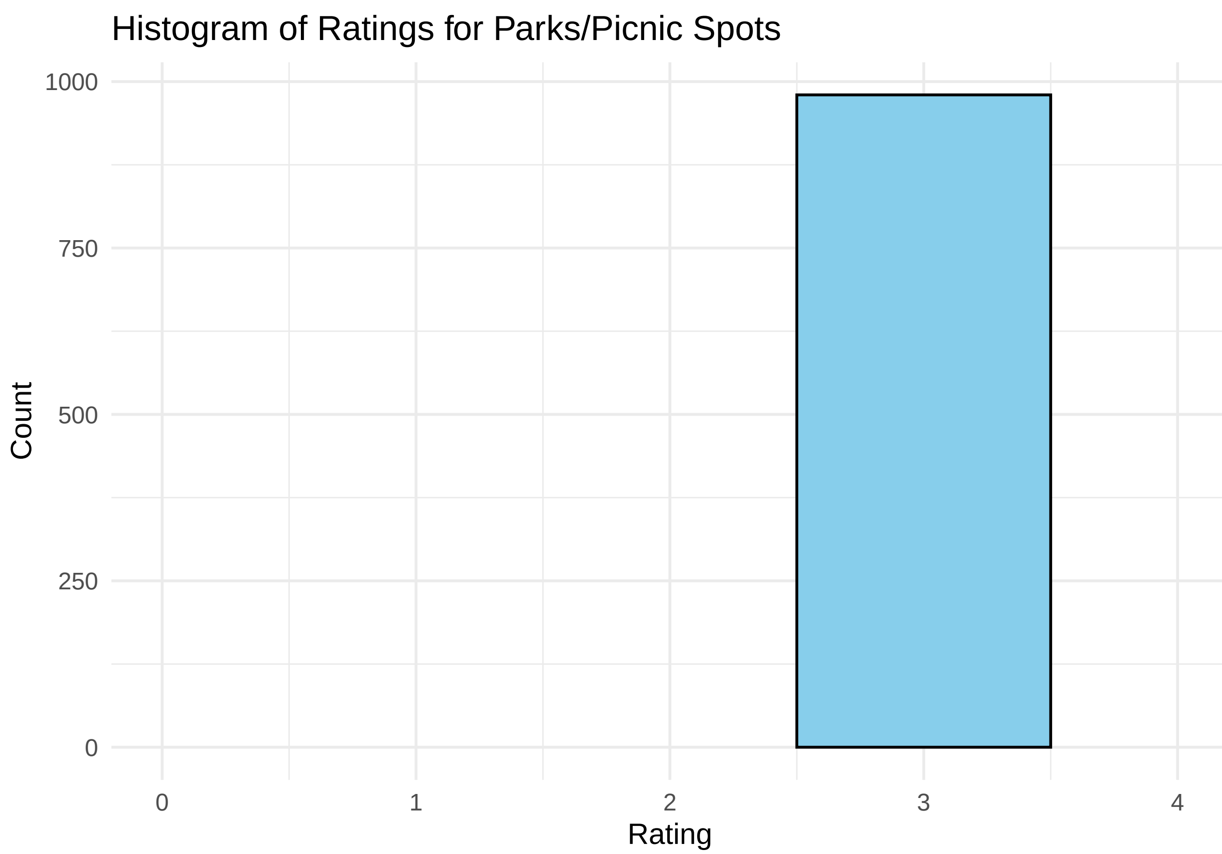
9.8 Combined Density Plot

```
# Density Plot for each category (all together excluding `Parks/Picnic Spots`)
long_data.temp <- pivot_longer(travelUpdate.data,
                               cols = c(-User.ID,-"Parks/Picnic Spots"),
                               names_to = "Category", values_to = "Rating")

ggplot(long_data.temp, aes(x = Rating, fill = Category, color = Category)) +
  geom_density(alpha = 0.3) +
  labs(title = "Density Distribution of Ratings for Each Category",
       x = "Rating", y = "Density")
```



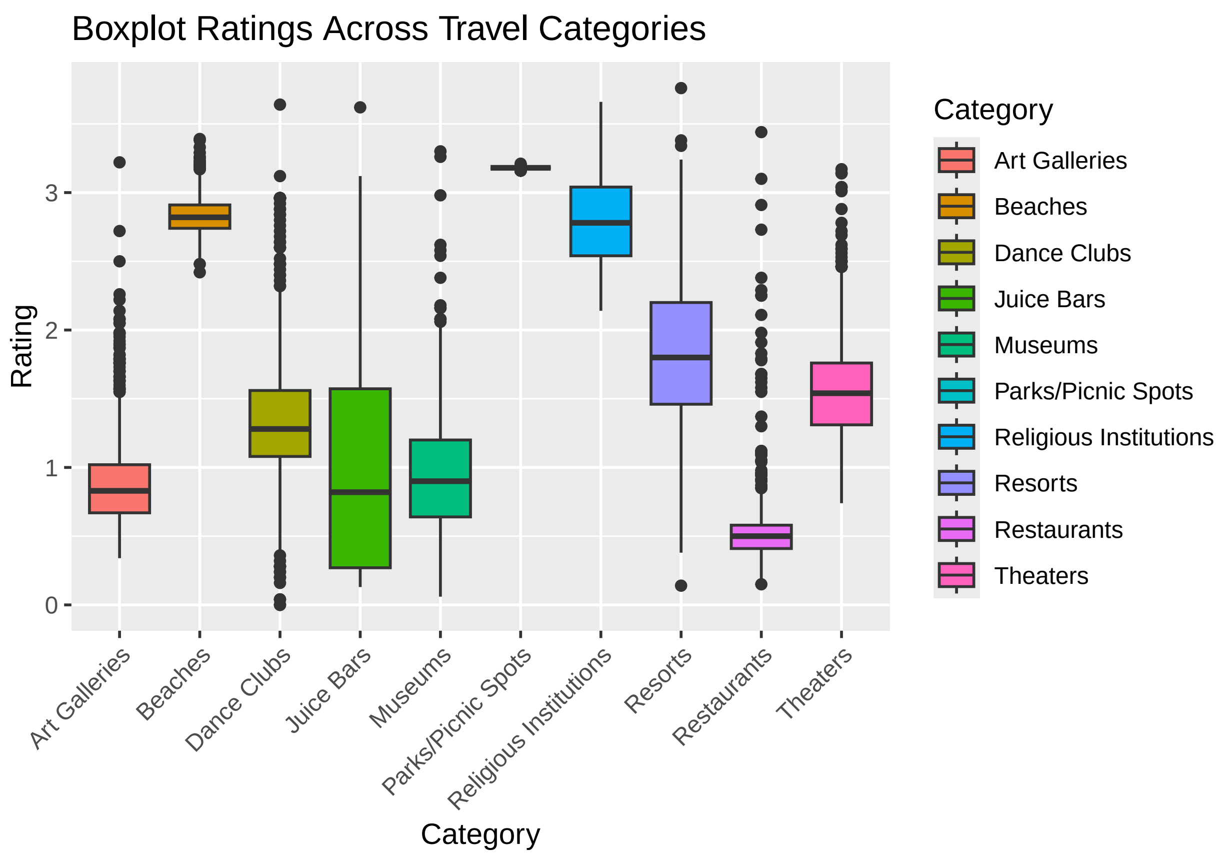
9.9 Investigating Parks/Picnic Spots

```
# Histogram for Parks/Picnic Spots Ratings
ggplot(long_data %>% filter(Category == "Parks/Picnic Spots"),
       aes(x = Rating)) +
  geom_histogram(binwidth = 1, fill = "skyblue", color = "black") +
  labs(title = "Histogram of Ratings for Parks/Picnic Spots",
       x = "Rating", y = "Count") +  coord_cartesian(xlim = c(0, 4)) +
  theme_minimal()
```



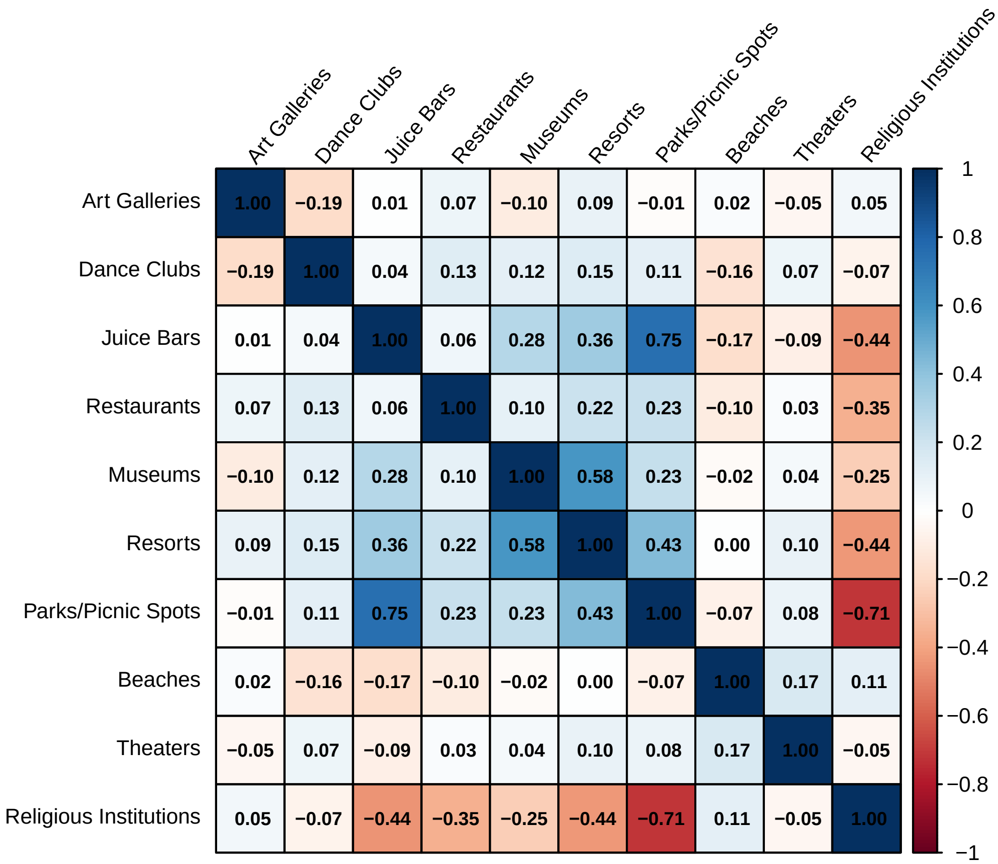
9.10 Boxplot Analysis - Comparing Ratings Across Categories

```
ggplot(long_data, aes(x = Category, y = Rating, fill = Category)) +
  theme(axis.text.x= element_text(angle = 45, hjust = 1)) + geom_boxplot() +
  labs(title = "Boxplot Ratings Across Travel Categories")
```



9.11 Correlation Matrix

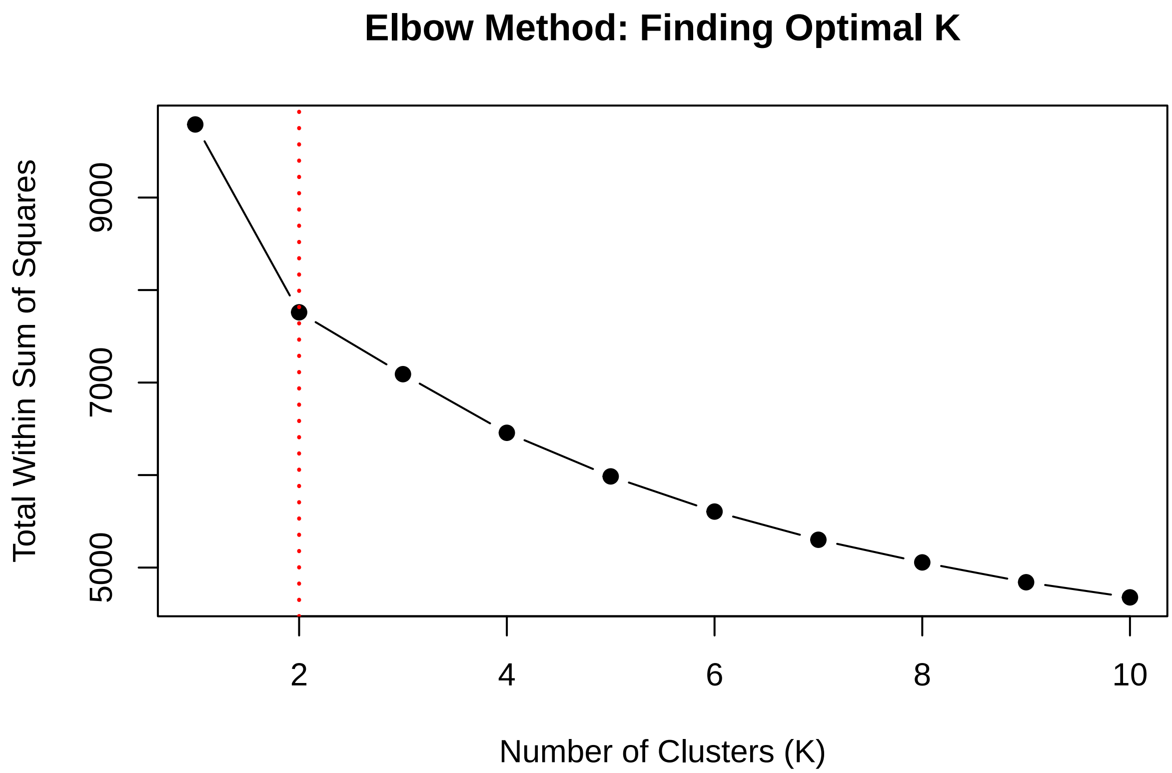
```
# computing correlation matrix (excluding User ID)
correlation.map <- cor(travelUpdate.data[, -1])

# displaying correlation heatmap
corrplot(correlation.map,
         addCoef.col = "black",    # Adding black coefficients
         cl.cex = 0.7,             # Reduce legend font size
         method = "color",
         type = "full",            # Display upper half
         tl.col = "black",         # Black text for labels
         tl.srt = 50,              # Rotate labels 45°
         number.cex = 0.6,         # Reduce coefficient font size
         tl.cex = 0.7,             # Reduce variable label size
         diag = TRUE,              # Hide diagonal
         addgrid.col = "black")    # Light gray borders
```



9.12 K-Means Clustering: Finding the Optimal Number of Clusters

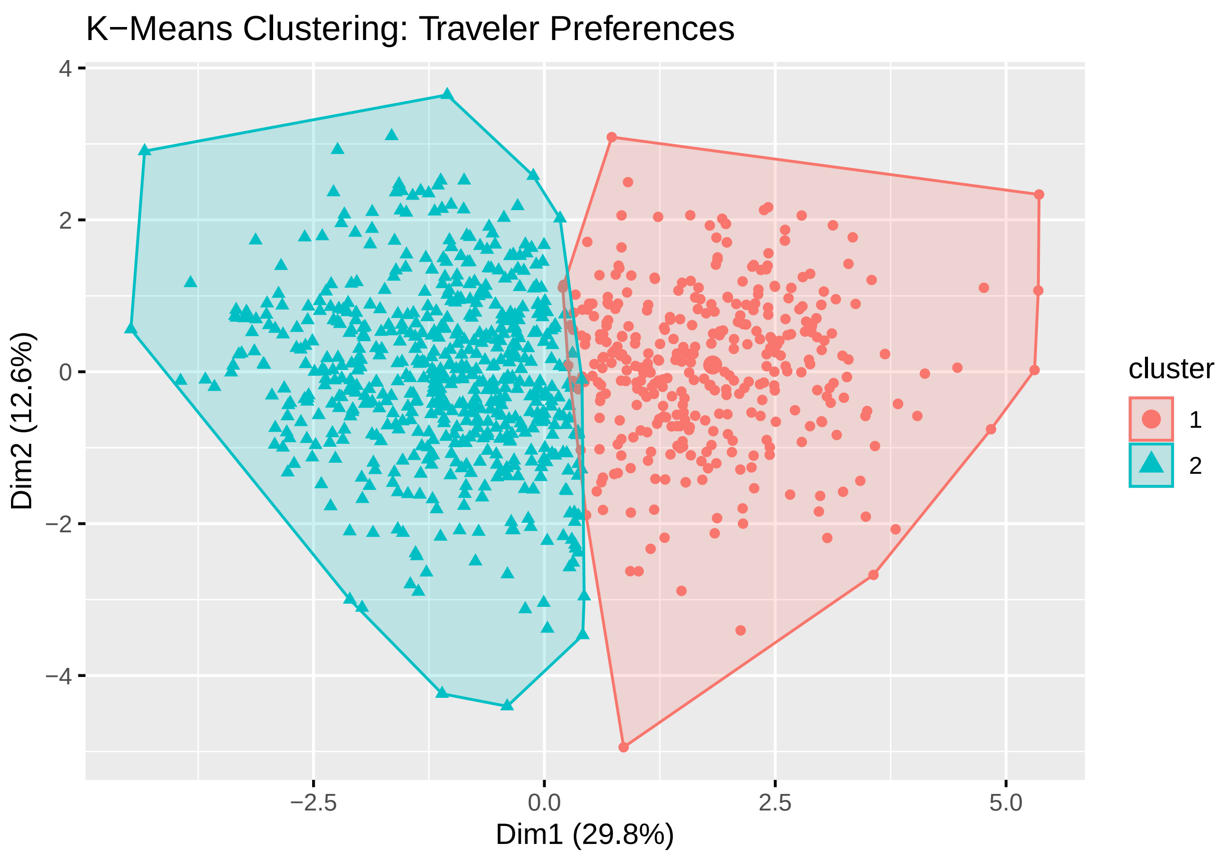
```r
# Should consist list of all 10 categories
clustering.data <- travelUpdate.data[, -1]

# scaling the data (standardizing our ratings)
clustering.data <- scale(clustering.data)

# ensuring reproducibility
set.seed(123)

compute_ss = function(k) {
  return( kmeans(clustering.data, centers = k, nstart = 25)$tot.withinss )
}

# Compute the total within-cluster sum of squares for different K values
wss <- sapply(X=1:10, FUN=compute_ss)

# Plotting the Elbow Method
plot(1:10, wss, type = "b", xlim = c(1,10),
     pch = 19, frame = TRUE, main = "Elbow Method: Finding Optimal K",
     xlab = "Number of Clusters (K)", ylab = "Total Within Sum of Squares")
```

```r
# Adding a dotted vertical line at K=2
abline(v = 2, lty = 3, col = "red", lwd = 2)
```



9.13 K-Means Clustering: Grouping Travelers

```r
# applying K-Means Clustering
k.optimal <- 2
kmeans.result <- kmeans(clustering.data, centers = k.optimal, nstart = 30)

# adding cluster labels to our `travelUpdate.data` dataset
travelUpdate.data$Cluster <- as.factor(kmeans.result$cluster)

# Visualizing clusters using PCA
fviz_cluster(kmeans.result, data = clustering.data,
             main = "K-Means Clustering: Traveler Preferences",
             ellipse.type = "convex", geom = "point")
```



9.14 Analysing Cluster Centers

```
# viewing cluster centers to see which categories drive the differences
kmeans.result$centers
```

```
##   Art Galleries Dance Clubs  Juice Bars Restaurants    Museums    Resorts
## 1    0.05206298   0.2520339   0.8687001   0.4211929  0.6378972  0.7535470
## 2   -0.03213128  -0.1555457  -0.5361284  -0.2599441 -0.3936857 -0.4650604
##   Parks/Picnic Spots    Beaches     Theaters Religious Institutions
## 1          0.8334284 -0.2248408 -0.02757251             -0.7989443
## 2         -0.5143601  0.1387631  0.01701670              0.4930779
```

9.15 Hierarchical Clustering: Dendrogram Visualization

```
# computing distance matrix
travel.dist <- dist(clustering.data, method = "euclidean")

# performing hierarchical clustering using ward's method
hc <- hclust(travel.dist, method = "ward.D2")

# converting to dendrogram object for better visualization
dend <- as.dendrogram(hc)
```

```r
# separating the dendrogram at k clusters and assigning colors
dend <- color_branches(dend, k = k.optimal)

# modifying x-axis labels to show every 50th traveler
dend_labels <- labels(dend)
dend_labels <- ifelse(1:length(dend_labels) %% 50 == 0, dend_labels, "")
labels(dend) <- dend_labels

# plotting the Dendrogram
plot(dend, main = "Dendrogram of Traveler Preferences",
     xlab = "Clustered Travelers",ylab = "Height",
     frame.plot = TRUE, las = 2, cex = 0.7)

# adding the horizontal cut-off line at appropriate height
abline(h = 39, col = "purple", lwd = 2, lty = 2)
```