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 destinations 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.

```
# 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.

```
# 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 destination paces 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
library(dplyr)       # Data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyr)       # Data transformation
library(ggplot2)     # Data visualization
library(corrplot)    # Correlation analysis
```

```
## corrplot 0.95 loaded
```

```r
library(factoextra)  # Clustering analysis
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```r
library(cluster)     # Clustering algorithms
```

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 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
```

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 moderate ratings ($\approx 0.89 - 0.94$) with slightly lower variability, implying generally favorable but not exceptional experiences
- `Dance Clubs` have a broad spread (SD = 0.48), possibly due to differences in expectations or quality between locations

Overall 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.

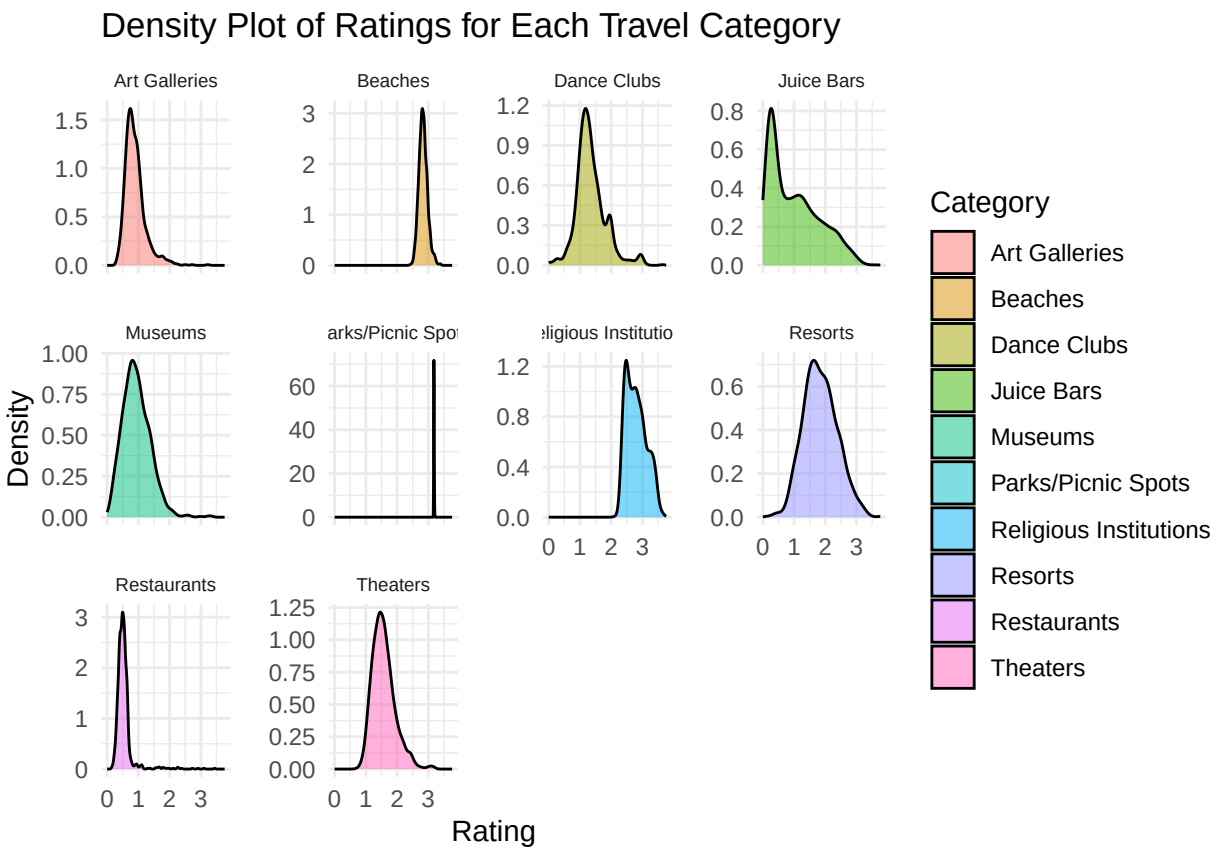
5.3 Data Distribution Analysis

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 of 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:

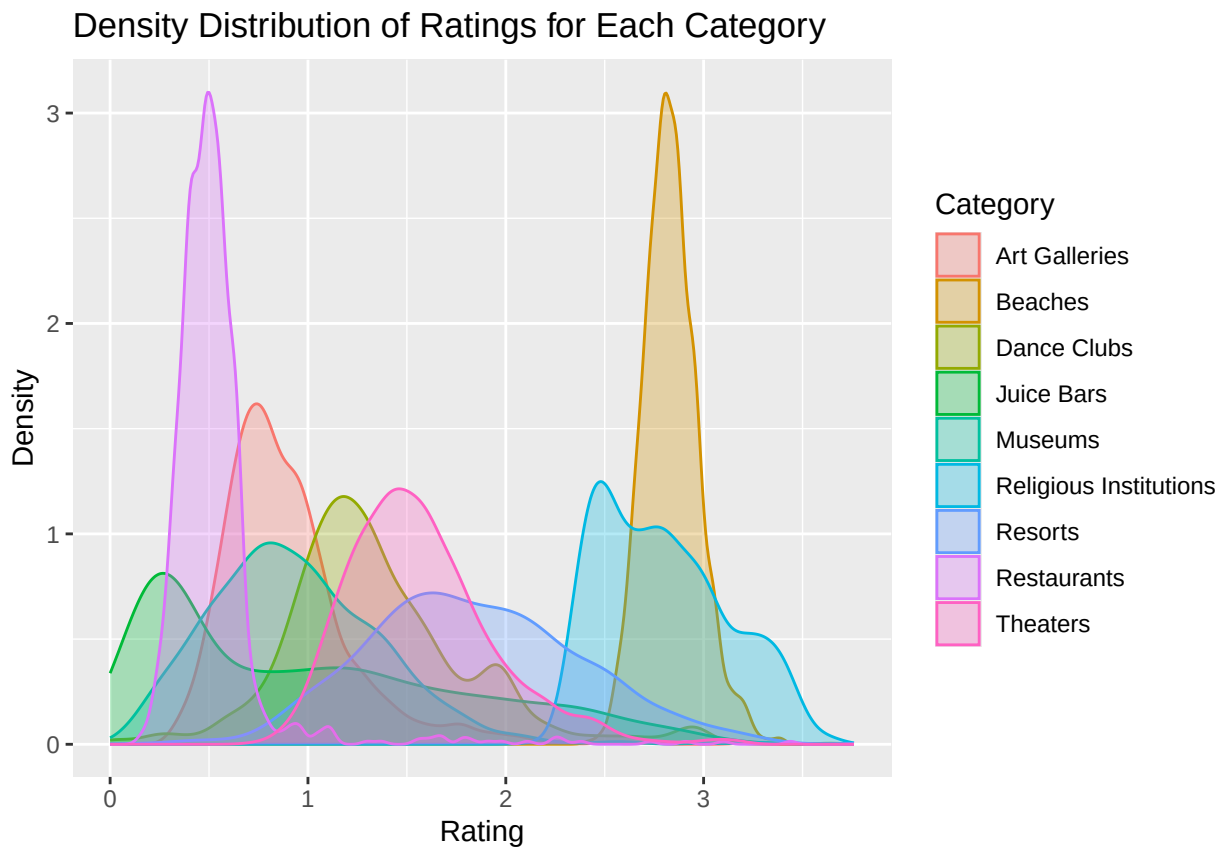
- Separate density graphs avoid overlap and are easier to read
- Some locations have wider curves that reflect a range of traveler perspectives, others have narrow peaks that indicate consistent ratings.
- `Dance Clubs` have shown a bimodal distribution, suggesting conflicting opinions where some travelers 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.

```
# 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")
```



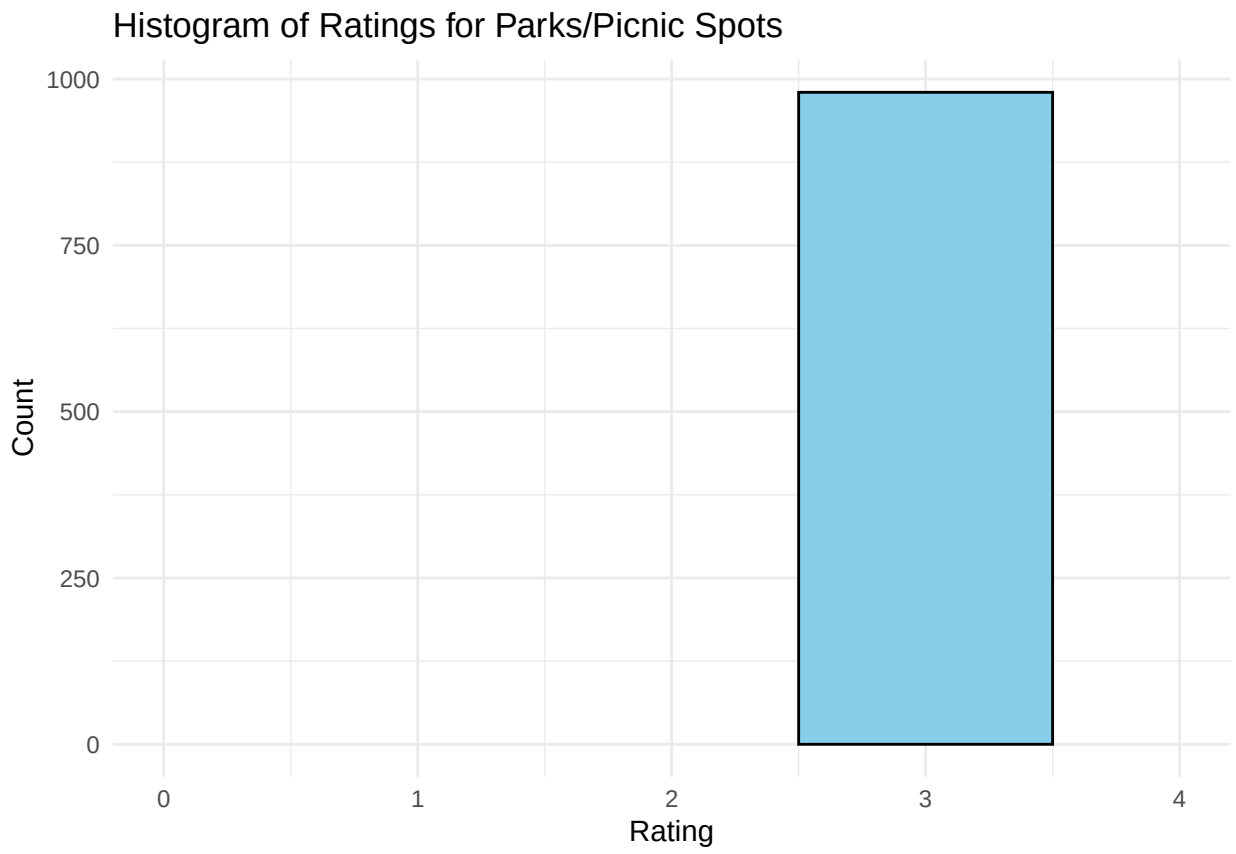
Insights:

- `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
- `Dance Clubs` exhibit a bimodal trend, reinforcing the inconsistent user experience observed earlier

5.33 Investigating Parks/Picnic Spots

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

```
# 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()
```



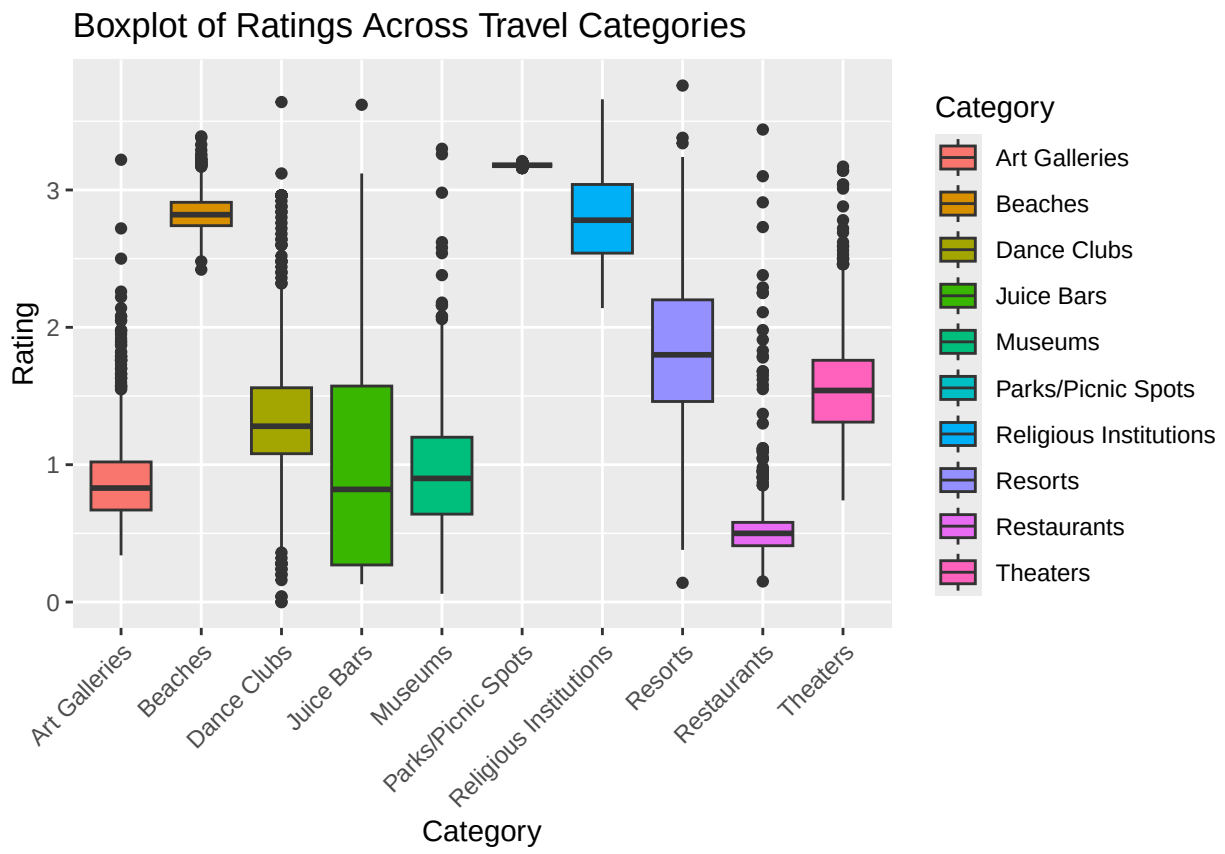
Insights:

- The histogram reveals 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

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 of Ratings Across Travel Categories")
```



Interpretation & 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
- `Dance Clubs` display many low outliers, reinforcing the polarized user experience observed in density plots

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 & Diversity of Ratings:

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

Outliers & Unexpected Trends:

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

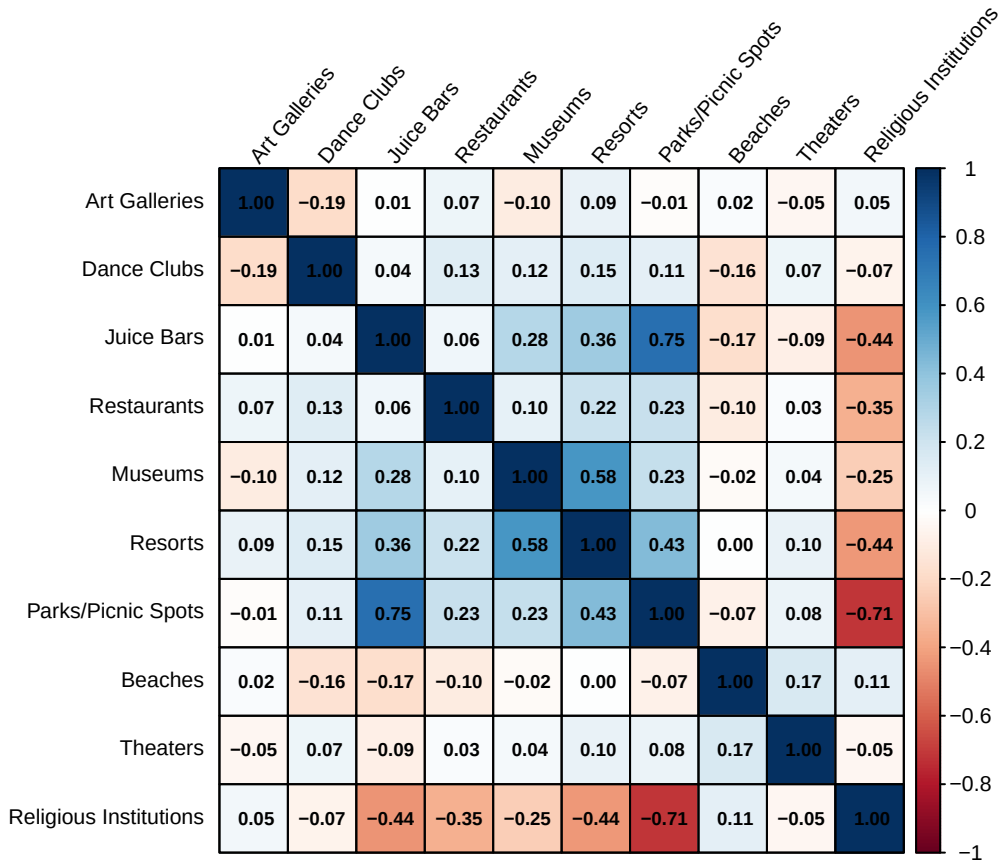
6. Correlation Analysis

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:

Categories like `Parks/Picnic Spots`, `Juice Bars`, and `Museums` demonstrate stronger relationships with other variables, while categories like `Religious Institutions` tend to show weaker or negative correlations. The plot will be highly useful for identifying clusters of related activities or inversely related categories that may represent contrasting preferences or behaviors.

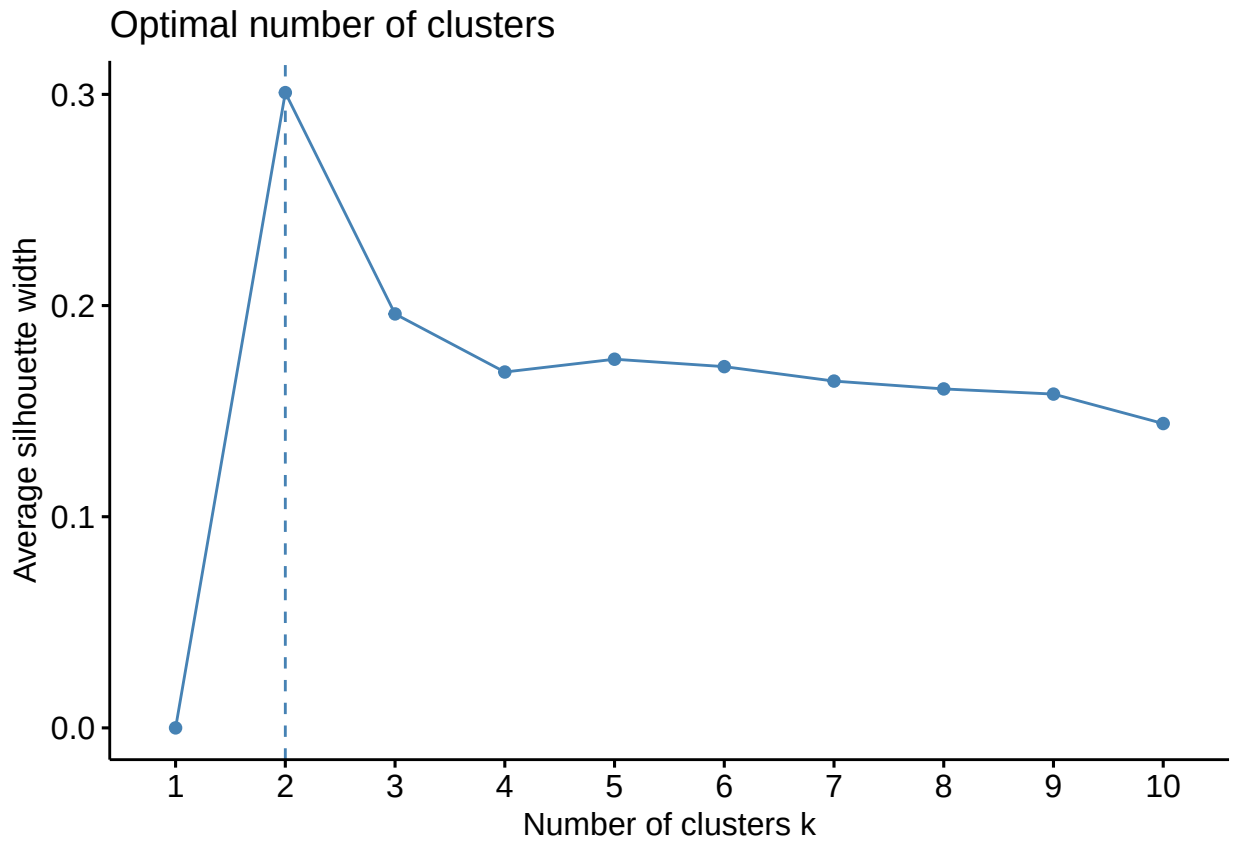
8. Cluster Analysis

```r
# setting row names to the UserID
row.names(travel.data) <- travel.data$User.ID

# Selecting only numeric category columns for clustering
data_clust <- travel.data[,-1]

# K-Means Clustering

# Determining the optimal number of clusters via the silhouette method
fviz_nbclust(data_clust, kmeans, method = "silhouette")
```

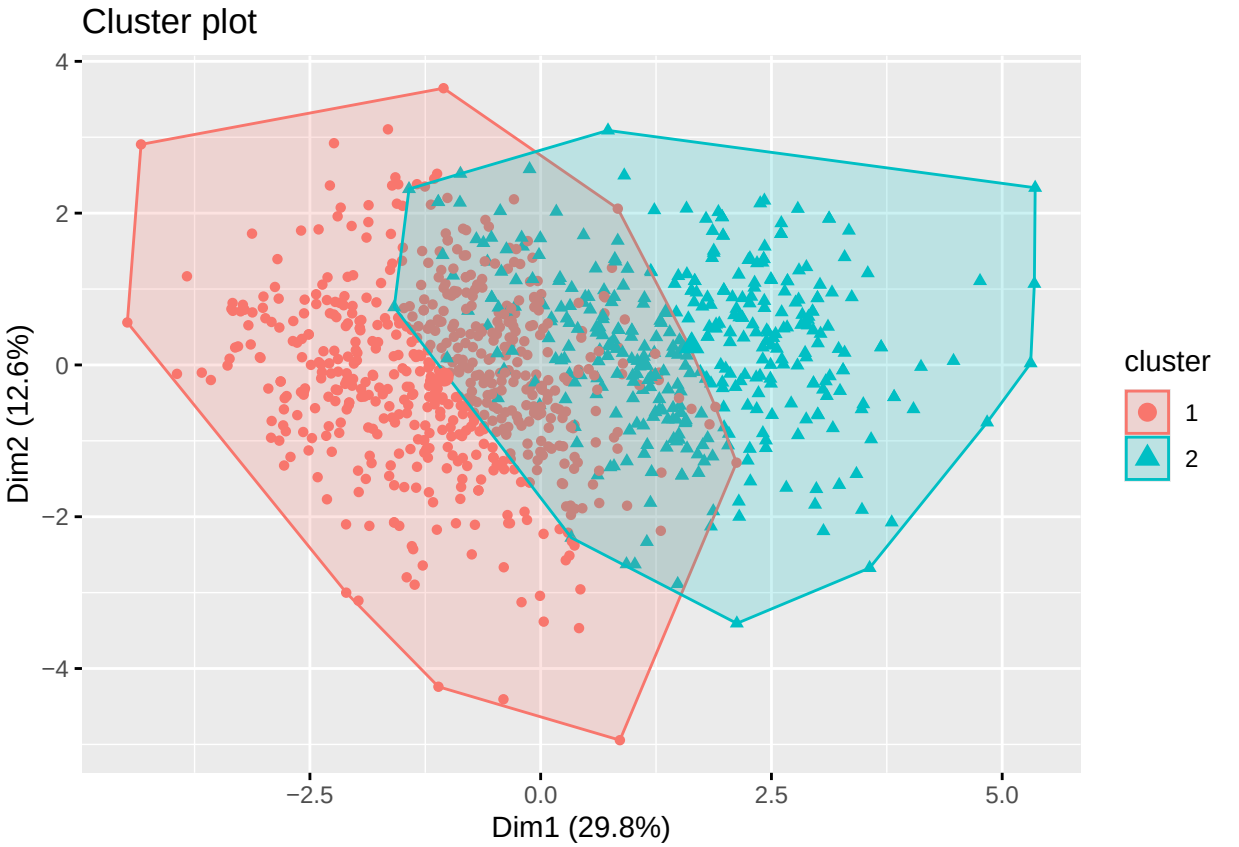


We can see from above silhouette plot the optimal number of clusters is $k = 2$.

```r
# Running K-Means Clustering
library(factoextra)
library(cluster)

# For reproducibility
set.seed(123)
optimal_Num_Clusters <- 2
kmeans_result <- kmeans(data_clust, centers = optimal_Num_Clusters, nstart = 25)

# Visualize the clusters
fviz_cluster(kmeans_result, data = data_clust, geom = "point")
```




```
# Helps to view cluster center to see which categories drive the differences
kmeans_result$centers
```

```
##   Art Galleries Dance Clubs Juice Bars Restaurants   Museums  Resorts
## 1     0.8850167    1.309900   0.491990   0.5003679 0.7862542 1.625284
## 2     0.9059948    1.419476   1.829398   0.5828010 1.1800000 2.183560
##   Parks/Picnic Spots  Beaches Theaters Religious Institutions
## 1           3.176973 2.854331 1.597124                2.925485
## 2           3.187147 2.804895 1.526099                2.601571
```

Insights:

Optimal Number of Clusters

The silhouette plot indicates that 2 is the best number of clusters for the dataset, which is good because a higher silhouette width means better separation between clusters and tighter grouping within each cluster.

Cluster Plot

In above cluster plot we can see the 2 clusters that are well-separated visually,indicating that travellers can be grouped into two distinct preference groups based on their ratings.

Cluster Centers The cluster centers show the average ratings for each category within each cluster. This helps us understand the characteristics of each cluster.

Overall, 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, and the centroids (blue triangles) give you a quick look at the average profile for each group.

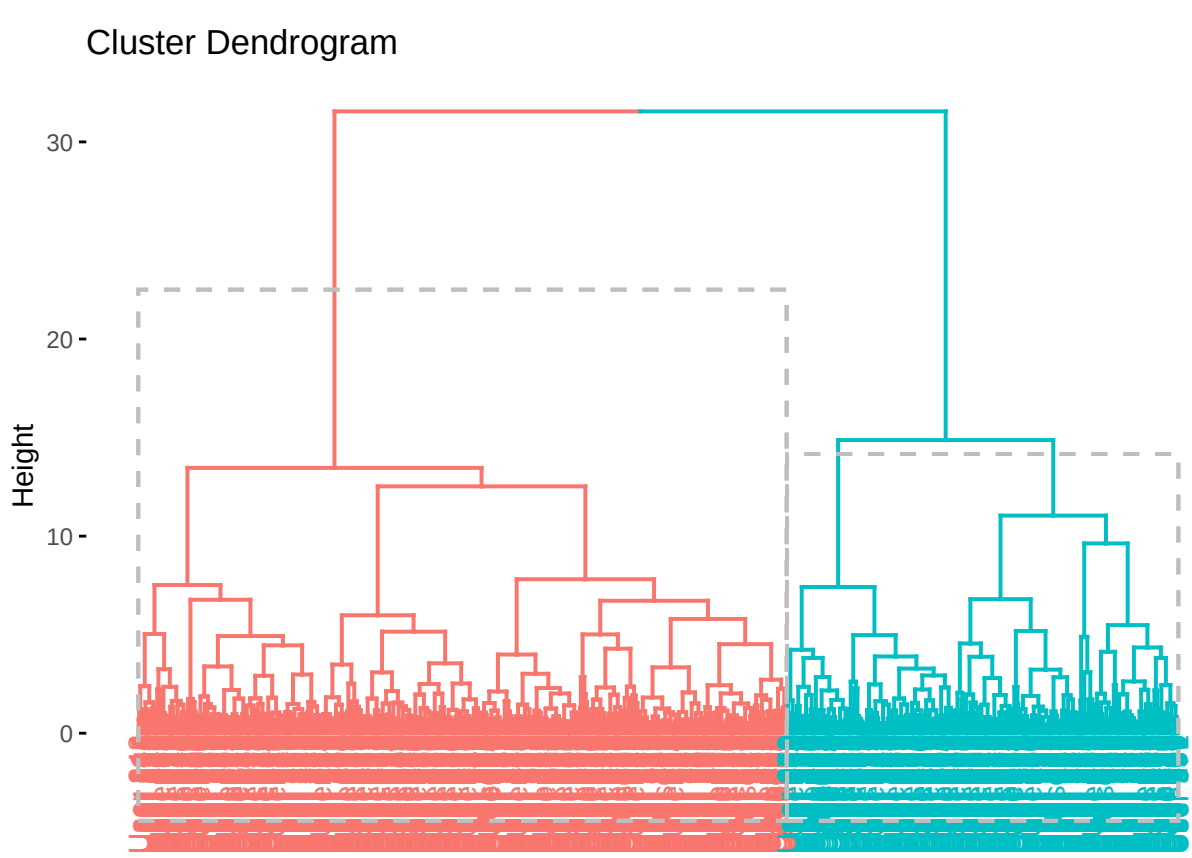
```r
# Hierarchical clustering

# Performing hierarchical clustering with k = 2 optimal clusters
hc_result <- hcut(data_clust, k = optimal_Num_Clusters)

# Visualizing the Dendrogram
fviz_dend(hc_result, rect = TRUE)
```

```
## Warning: The ‘<scale>‘ argument of ‘guides()‘ cannot be ‘FALSE‘. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.

## Warning in data.frame(xmin = unlist(xleft), ymin = unlist(ybottom), xmax =
## unlist(xright), : row names were found from a short variable and have been
## discarded
```



#

9. Conclusion

10. Appendix: Full R Code