Agglomerative Hierarchical Clustering

Implementing Clustering on TripAdvisor Customer Rating Data

Packages used

cluster: Used to create cluster from distance matrix

qplot: Used to create the heatmap and coloring the heatmap

Data Description:

The data is from tripadvior in which the users have given **reviews** to different kinds of destinations. The values has been normalized to lie between 0 and 4 and are in decimals too. The data has 980 rows and 11 columns. The columns can be defined as follows.

1. User ID
2. Art Gallery
3. Dance Club
4. Juice Bars
5. Restaurants
6. Museums
7. Resorts
8. Parking
9. Beaches
10. Theaters
11. Religious Institutes

Hierarchal Clustering (summary)

Clustering is a technique of grouping same kind of elements in table together. There can be many application of clustering, but the most important once are as follows:

1- To find the correlation between in the data.
2- To understand the pattern in unlabeled data.

By grouping same kind of columns together with the help of clustering, we can understand the correlation between columns can take appropriate steps.

Sometimes, the data does not have labels or names given to attributes or column, in that case clustering can help us understanding the patter in distinguishing the groups so that they can be analyzed separately.

Findings

1.

```
library(cluster)
install.packages("gplots") # installing the gplots package
library(gplots) #loading the gplot pakages
```

First, I have loaded appropriate libraries in order to do proper functionality. The cluster function will help creating the cluster, where as the gplots will help us create a heatmap.

2.

```
df <- read.csv(file.choose()) # laod the file with file name "tripadvisor"
names <- c("User ID","art gallery","dance club","juice bars","restaurants","m
colnames(df) <- names # assigning those names to the columns of data frame

str(df) # checking the stucture of the data frame
```

First, I am reading the file directly from the desktop as csv file, the name of the file is tripadvisor. Then, I am creating a names vector that has all the names stored in it that has to be assigned to the columns of data frame. Then, I am assigning the names to the columns.

```
> str(df) # checking the stucture of the data frame
'data.frame': 980 obs. of  11 variables:
 $ User ID    : Factor w/ 980 levels "User 1","User 10",..: 1 112 223 334 445 556 667 778 889 2 ...
 $ art gallery: num  0.93 1.02 1.22 0.45 0.51 0.99 0.9 0.74 1.12 0.7 ...
 $ dance club : num  1.8 2.2 0.8 1.8 1.2 1.28 1.36 1.4 1.76 1.36 ...
 $ juice bars : num  2.29 2.66 0.54 0.29 1.18 0.72 0.26 0.22 1.04 0.22 ...
 $ restaurants: num  0.62 0.64 0.53 0.57 0.57 0.27 0.32 0.41 0.64 0.26 ...
 $ museums    : num  0.8 1.42 0.24 0.46 1.54 0.74 0.86 0.82 0.82 1.5 ...
 $ resorts    : num  2.42 3.18 1.54 1.52 2.02 1.26 1.58 1.5 2.14 1.54 ...
 $ parking    : num  3.19 3.21 3.18 3.18 3.18 3.17 3.17 3.17 3.18 3.17 ...
 $ beaches    : num  2.79 2.63 2.8 2.96 2.78 2.89 2.66 2.81 2.79 2.82 ...
 $ theaters   : num  1.82 1.86 1.31 1.57 1.18 1.66 1.22 1.54 1.41 2.24 ...
 $ religious  : num  2.42 2.32 2.5 2.86 2.54 3.66 3.22 2.88 2.54 3.12 ...
```

Looks like that the structure of data frame depicts a clear picture. The first column is a factor because it is the user ID which is specific to each user. Then, all the other columns showing the reviews belong to that column are in numeric form.

3.

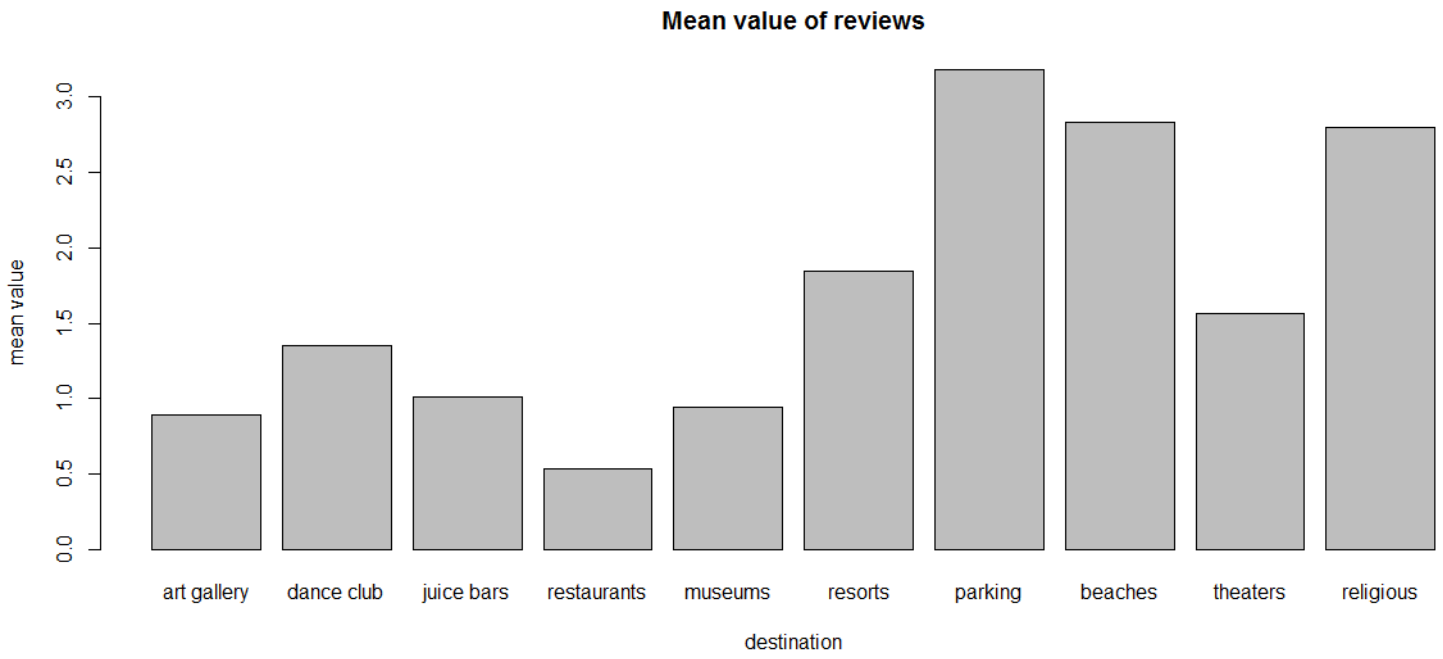
```
summary(df[,2:11]) # checking the summary of the data frame
barplot(sapply(df[,2:11],mean)) # creating a bar plot by iretrating across all columns.
```

```
> summary(df[,2:11]) # checking the summary of the data frame
  art gallery       dance club       juice bars      restaurants        museums          resorts          parking     
 Min.   :0.3400   Min.   :0.000   Min.   :0.130   Min.   :0.1500   Min.   :0.0600   Min.   :0.140   Min.   :3.160  
 1st Qu.:0.6700   1st Qu.:1.080   1st Qu.:0.270   1st Qu.:0.4100   1st Qu.:0.6400   1st Qu.:1.460   1st Qu.:3.180  
 Median :0.8300   Median :1.280   Median :0.820   Median :0.5000   Median :0.9000   Median :1.800   Median :3.180  
 Mean   :0.8932   Mean   :1.353   Mean   :1.013   Mean   :0.5325   Mean   :0.9397   Mean   :1.843   Mean   :3.181  
 3rd Qu.:1.0200   3rd Qu.:1.560   3rd Qu.:1.573   3rd Qu.:0.5800   3rd Qu.:1.2000   3rd Qu.:2.200   3rd Qu.:3.180  
 Max.   :3.2200   Max.   :3.640   Max.   :3.620   Max.   :3.4400   Max.   :3.3000   Max.   :3.760   Max.   :3.210  
    beaches         theaters       religious     
 Min.   :2.420   Min.   :0.740   Min.   :2.140  
 1st Qu.:2.740   1st Qu.:1.310   1st Qu.:2.540  
 Median :2.820   Median :1.540   Median :2.780  
 Mean   :2.835   Mean   :1.569   Mean   :2.799  
 3rd Qu.:2.910   3rd Qu.:1.760   3rd Qu.:3.040  
 Max.   :3.390   Max.   :3.170   Max.   :3.660  
>
```

Looking at the summary of the dataset, we can see that the minimum value of reviews in each one of the destination is zero and the maximum value is below 4.

Moreover, most of the value of reviews are normally distributed because mean and median are almost equal.

To get a more clear picture, I have drawn a bar plot that is showing mean value of reviews belong to each one of the destinations. The graph is shown below.

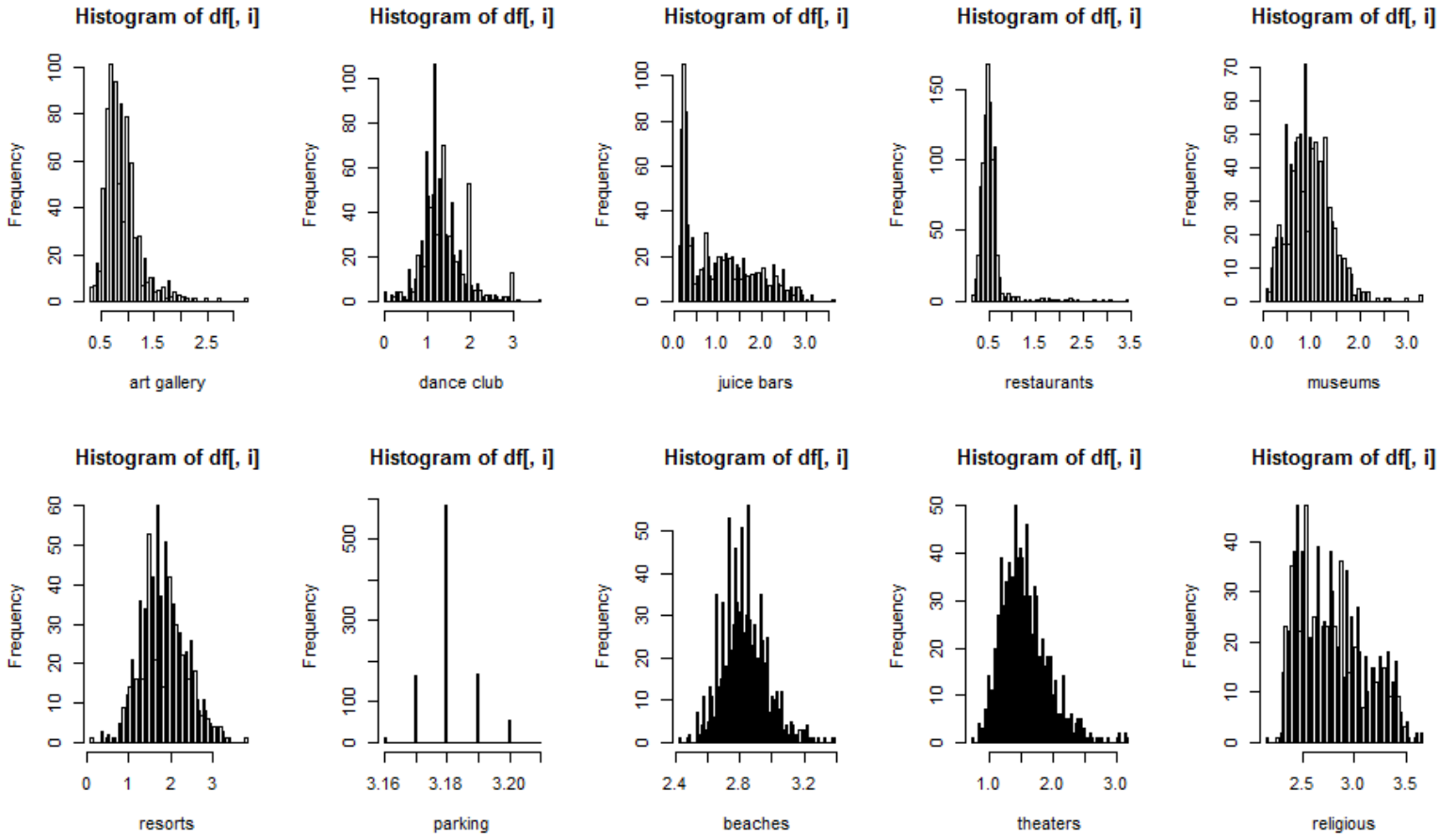


Looks like the mean value of reviews given to parking is highest. Meaning, parking is given the highest reviews on average. Moreover, it is surprising to know that restaurants are given the least ratings.

4.

```
par(mfrow=c(2,5)) # converting the window to show 10 plots at a time
# applying the for loop to create 10 histograms so that we can understand the distrubution of data.
for(i in 2:11)
{
hist(df[,i],breaks=100)
}
dev.off() # closing the window from showing 10 plots to 1 plot
```

In this step, I have first created a window that can hold 10 plots at a time so that we can get a big picture of the data. Then, using a for-loop to iterate through all the columns of the data frame, I have created a histogram in order to understand the distribution of data. In the end, I closed the window back to a single plot.



Looks like most of them are normally distributed. Except, parking juice bars and restaurants.

The juice bar and restaurants rating is rightly skewed. Meaning, people mostly have given bad rating to those places.

The parking ratings looks like it has been given on a 5 start scale that is why we have 5 straight lines.

5.

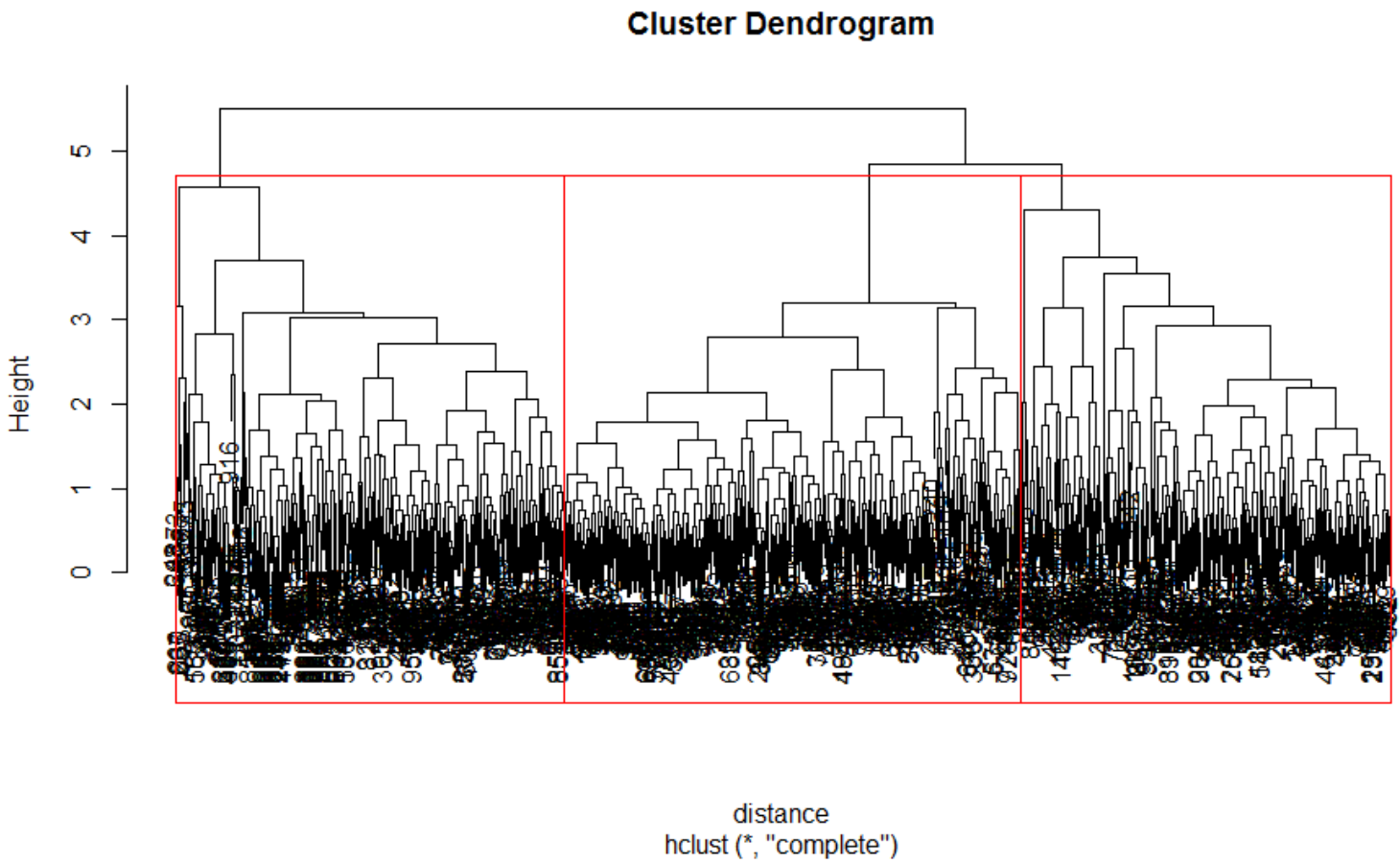
```
distance <- dist(df) # calculating the distance of each point from each other point
hc <- hclust(distance) # creating a cluster of numbeers colse to each other
plot(hc) # ploting the cluters to see a visulization of clusters
rect.hclust(hc, k = 3)  # creatinga rectange around 3 big main clusters
```

First, I am calculating distance of each point from each other point. The dist function will help us creating distance matrix that have distance calculated of each point from each other point.

Then, using hclust function and giving distance matrix as argument, we can create cluster of data points. In other words, the hclust function will automatically create groups of elements that are closer to each other.

To have a better picture of the grouping, we can create a dendrogram by using a plot command.

At last, the rect.hclust command help us create a rectangle around clusters. I have given 3 as parameter to have rectangle around 3 most big clusters.



As we can see, the red line is surrounding three section of this dendrogram. These three section are three different clusters, and each element in those cluster have same property. Meaning, each reviews that user have given has been clustered, and the reviews given in one cluster have same property because they are close to each other. We can use individual cluster to extract the user ID and can use that ID to understand the property of those user, but this is out of scope of this analysis.

6.

```
cu <- cutree(hc,3) # cutting the tree into 3 most big clusters
table(cu) # calculating how many value each cluster holds
```

Using cutree command I have cut the above shown tree into three parts, each part is one cluster similar to the red rectangle showing in above graph. After cutting the tree, we can use table command to see how many user lies in each one of the cluster.

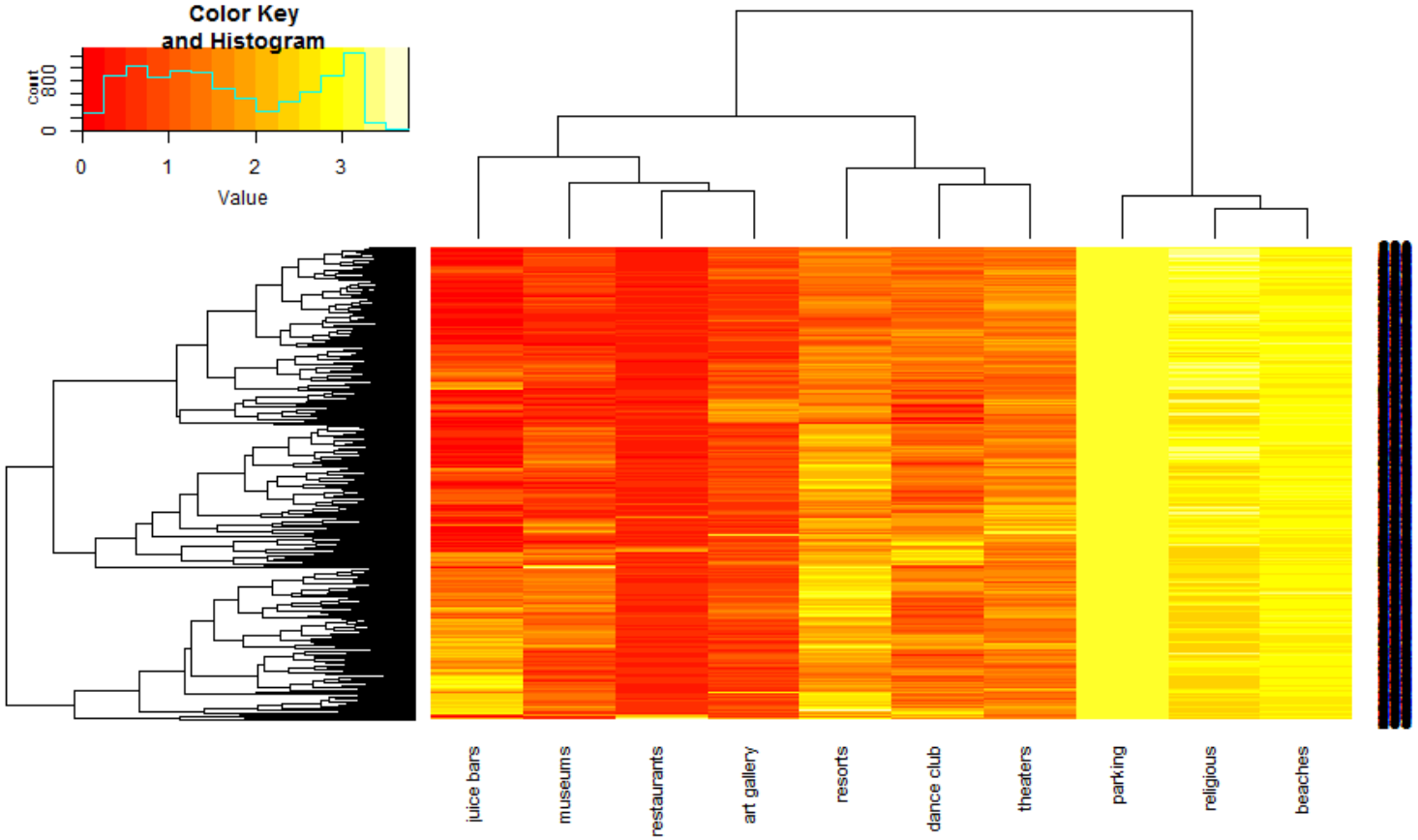
```
> table(cu) # calculating how many value each cluster holds
cu
  1   2   3
314 368 298
```

It is clear that first cluster has 314 users, the second have highest 368 user and third have lowest 298 users.

7.

```
heatmap.2(as.matrix(df[,2:11]),trace="none",cexRow=1,cexCol=0.8,margin=c(7,10)) # creating a heatmap that is clu
aggregate(df[,2:11],by=list(cu),mean) # calculating mean of each kind of review in all three clusters to get the
```

To get an overall picture of the whole analysis we have done so far, we can create a heatmap using heatmap.2 command and passing appropriate arguments. The heatmap will look like this.



It is clear by looking at the colors of heatmap that the data is divided into three main clusters, the left one which is extremen red are all the values that have low rating, the middle one is less dark and have a bit higher reviews rating, and the right most one is dark yellow which have highest ratings given by the user.

On the x-axis we can also see the name of the destinations. The parking, religious and beaches are given highest rating and hence they are clustered together. Whereas juice bars, museums and restaurants are given least ratings and hence are grouped together.

To take a step further, I have calculated the mean value of ratings belongs to each destination, separately for all three clusters. It looks something like this.

```
> aggregate(df[,2:11],by=list(cu),mean) # calculating mean of each kind of review in all three clusters to get th
ference among them.
  Group.1 art gallery dance club juice bars restaurants   museums  resorts  parking  beaches theaters religious
1       1   0.8669745   1.355414  1.8923248   0.6165924 1.1903185 2.218344 3.187707 2.808025 1.542580  2.576178
2       2   0.9460870   1.199891  0.5879348   0.4388043 0.6692391 1.400000 3.176440 2.846005 1.474239  2.991929
3       3   0.8555034   1.538255  0.6123826   0.5595973 1.0097315 1.994228 3.179362 2.850034 1.715302  2.796275
```

From this I observed that, in each of the three cluster people have given almost equal ratings to all of the destinations except juice bars and museums.

The second cluster has given bad reviews to museums, whereas first cluster has given very good reviews to juice bars.

References:

Anonymous. June11, 2017. *HowToDataViz.* “How to Make an R Heatmap with Annotations and Legend”. https://www.youtube.com/watch?v=T7_j444LMZs

Anonymous. “Travel Review Dataset”. *University of Carolina Irvin*. https://archive.ics.uci.edu/ml/datasets/Travel+Reviews

Davo. May 15, 2018. “Making a heatmap in R with the pheatmap package”. *DAVE TANG’S BLOG.* https://davetang.org/muse/2018/05/15/making-a-heatmap-in-r-with-the-pheatmap-package/