CYO_marwa

Marwa

1/4/2020

Introduction

Survey data is one of the most essential sources for data science. However, dealing with survays can be challenging in many aspects. First of all, cleaning the data could be cumbersome and users usually leave some blank entries and that leads to many missing cells in the data table. Survays are also useful to understand the targeted audience demographics and to recommend things based on users' responses.

Dataset

The dataset chosen for this project is the food-world-cup which is a survey collected by the fiveThirtyEight website to ask people how much do they know and how much do they like certain cuisines around the globe. The survey also collects some other demographic information from participatnts like their age, income, education and the region they live in.

For each country mentioned in the survey, the users are asked to rate the food from that country on a scale from 0 t0 5 (They actually use N/A instead of 0) where :

5: I love this country's traditional cuisine. I think it's one of the best in the world.

4: I like this country's traditional cuisine. I think it's considerably above average.

3: I'm OK with this county's traditional cuisine. I think it's about average.

2: I dislike this country's traditional cuisine. I think it's considerably below average.

1: I hate this country's traditional cuisine. I think it's one of the worst in the world.

N/A: I'm unfamiliar with this country's traditional cuisine.

The user also enters whether he is knowledgable or not about world food in general and whether he is interested in learning about them or not.

a general look at the data table as imported from a csv file shows 48 columns and 1373 rows.

Data Cleaning and Pre-processing

The dataset is found in a form of a comma-separated file with many empty cells or N/As. In order to perform basic R functions, several data wrangling functions have to be applied:

1. Column names cleaning

The columns in the csv file had a long question asking "Please rate how much you like the traditional cuisine of (a country name) ?. Obviously we can get rid of the whole question and just preserve the country name as a column name. The following code took care of that using the user-defined function colClean:

```
colClean <- function(x){ colnames(x) <- gsub("Please.rate.how.much.you.like.the.traditional.cuisine.of.
MyData <- colClean(MyData)
```

2.Replacing NAs in numeric columns with 0

All country columns which are the columns ranging from 4 to 43 are processed with a function that replaces NAs with 0 so that the column is a numeric column with range from 0 to 5.

```
num_col_with_0 = function(i){MyData[,i] <- MyData[,i] %>% replace_na(0)}
# id is the set of numeric columns in the dataset
id <- c(4:43)
MyData[,id] <- lapply(MyData[,id], function(x) as.numeric(as.character(x)))
MyData[,id] <- sapply(id,num_col_with_0)
```

3. Replacing empty cells in factor columns with "Unknown"

Similar to the previous step, all empty cells are filled with unknown to be treated as a separate factor or category.

```
MyData <- MyData %>% mutate(Age = factor(Age), Income = factor(Income), Education = factor(Education), 
MyData <- MyData %>% mutate(Interest = factor(Interest))
levels(MyData$Age)[levels(MyData$Age) == ""] <- "unknown"
levels(MyData$Income)[levels(MyData$Income) == ""] <- "unknown"
levels(MyData$Education)[levels(MyData$Education) == ""] <- "unknown"
levels(MyData$Region)[levels(MyData$Region) == ""] <- "unknown"
levels(MyData$Gender)[levels(MyData$Gender) == ""] <- "unknown"
```

4. Removing some special characters

The second and the thrid column has some weired and additional characters the needed removal like this :

```
MyData$Interest <- str_replace(MyData$Interest,"\\ÃŠ","")
```

5. Transforming the Gender column into 1 and 0 numeric column

The reason behind this is to make it simpler for prediction. The ratio of empty cells in this column is small (9%) so we can easily assume we can convert them to one gender. However, I didn't apply that for this dataset but learned this transformation and would apply it in future when needed.

Analysis

To tackle the analysis of this dataset, three main approaches were performed.

1. General Exploratory Data Analysis

In this section, we get to know more about the dataset and its features.

First of all, we want to see which cuisine is the highest rated and which one is the least popular according to this dataset.

The best cuisine in this dataset is :

```
countries <- MyData[,id]
s <- colSums(countries)
best <- countries[which.max(s)] %>% colnames()

 best
```

```
## [1] "Italy."
```

The lowest rated cuisine is :

```
## [1] "Ivory.Coast."
```

Univariate Analysis

```
## # A tibble: 3 x 3
##   Gender  count  ratio
##   <fct>   <int>  <dbl>
## 1 unknown   134 0.0976
## 2 Female    660 0.481
## 3 Male      579 0.422
```

```
## # A tibble: 6 x 3
##   Income              count  ratio
##   <fct>               <int>  <dbl>
## 1 unknown               419 0.305
## 2 $0 - $24,999          138 0.101
## 3 $100,000 - $149,999   161 0.117
## 4 $150,000+             125 0.0910
## 5 $25,000 - $49,999     210 0.153
## 6 $50,000 - $99,999     320 0.233
```

```
## # A tibble: 6 x 3
##   Education                         count  ratio
##   <fct>                             <int>  <dbl>
## 1 unknown                             145 0.106
## 2 Bachelor degree                     386 0.281
## 3 Graduate degree                     325 0.237
## 4 High school degree                  115 0.0838
## 5 Less than high school degree         20 0.0146
## 6 Some college or Associate degree    382 0.278
```

```
## # A tibble: 5 x 3
##   Age     count  ratio
##   <fct>   <int>  <dbl>
## 1 unknown   134 0.0976
## 2 > 60      325 0.237
## 3 18-29     262 0.191
## 4 30-44     307 0.224
## 5 45-60     345 0.251
```

```
## # A tibble: 10 x 3
##    Region             count  ratio
##    <fct>              <int>  <dbl>
##  1 unknown              144 0.105
##  2 East North Central   188 0.137
##  3 East South Central    40 0.0291
##  4 Middle Atlantic      170 0.124
##  5 Mountain             102 0.0743
##  6 New England           73 0.0532
##  7 Pacific              215 0.157
##  8 South Atlantic       200 0.146
##  9 West North Central   107 0.0779
## 10 West South Central   134 0.0976
```

By examining the above tables, we can see that the Income column is the one with the largest number of empty cells (almost 30% of it is empty).

Correlation

To understand the relation between cuisines and if there are any correlation among the ratings of different cuisines. I first wanted to investigate the correlation among the ratings columns by obtaining the correlation matrix

```
x<- as.matrix(MyData[,id])
cor_mat = cor(x)
# I remove all columns with correlations above 0.5
hc = findCorrelation(cor_mat, cutoff=0.5)
 hc = sort(hc)
 reduced_Data = x[,-c(hc)]
 reduced_Data %>% colnames()
```

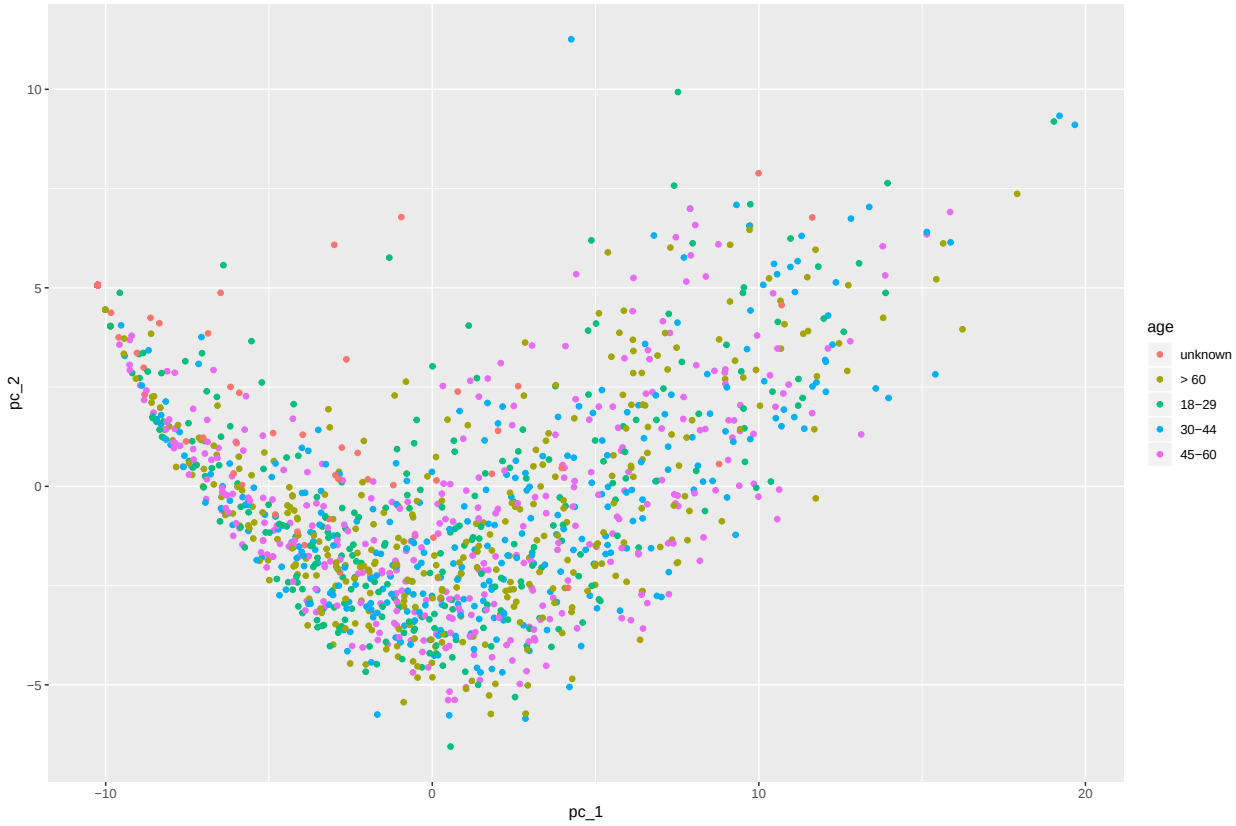
```
##  [1] "ŠAustralia."  "Croatia."     "Iran."        "Italy."
##  [5] "Nigeria."     "Portugal."    "Russia."      "South.Korea."
##  [9] "Switzerland." "Uruguay."     "India."       "Turkey."
## [13] "Ethiopia."    "Ireland."
```

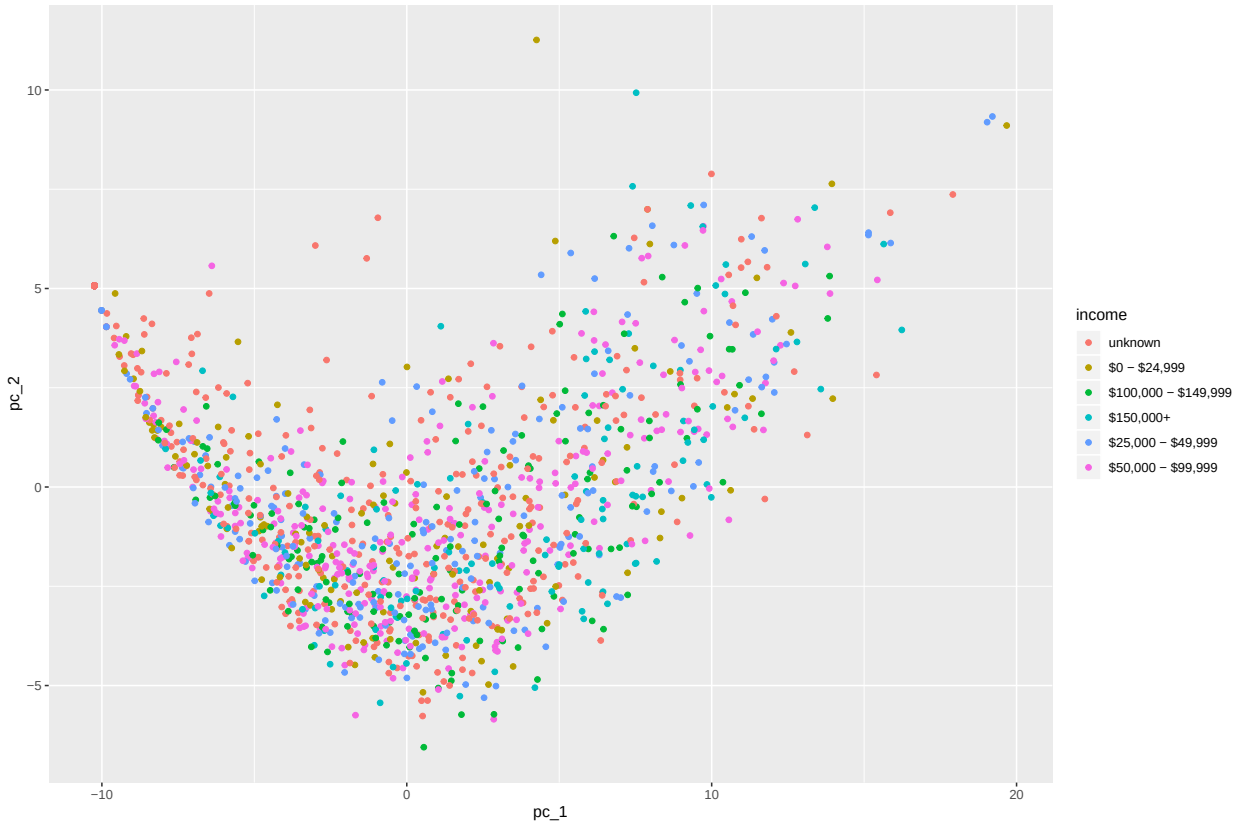
Here I try to retain only the variables with correlation larger or equal to 0.5. This reduced the dimensions of ratings from 40 to 14 and the independent cuisines are the following:

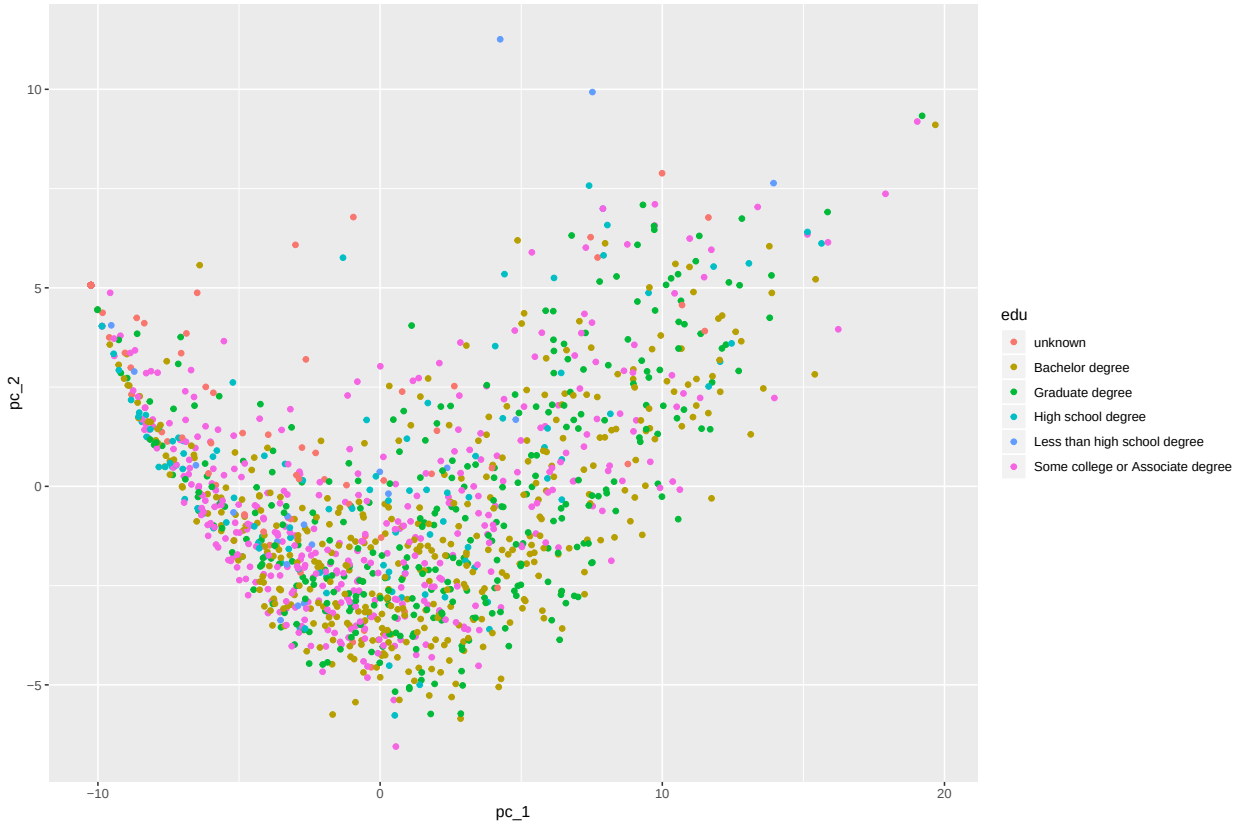
This may help later when we want to train the model to predict users ratings of some cousines.

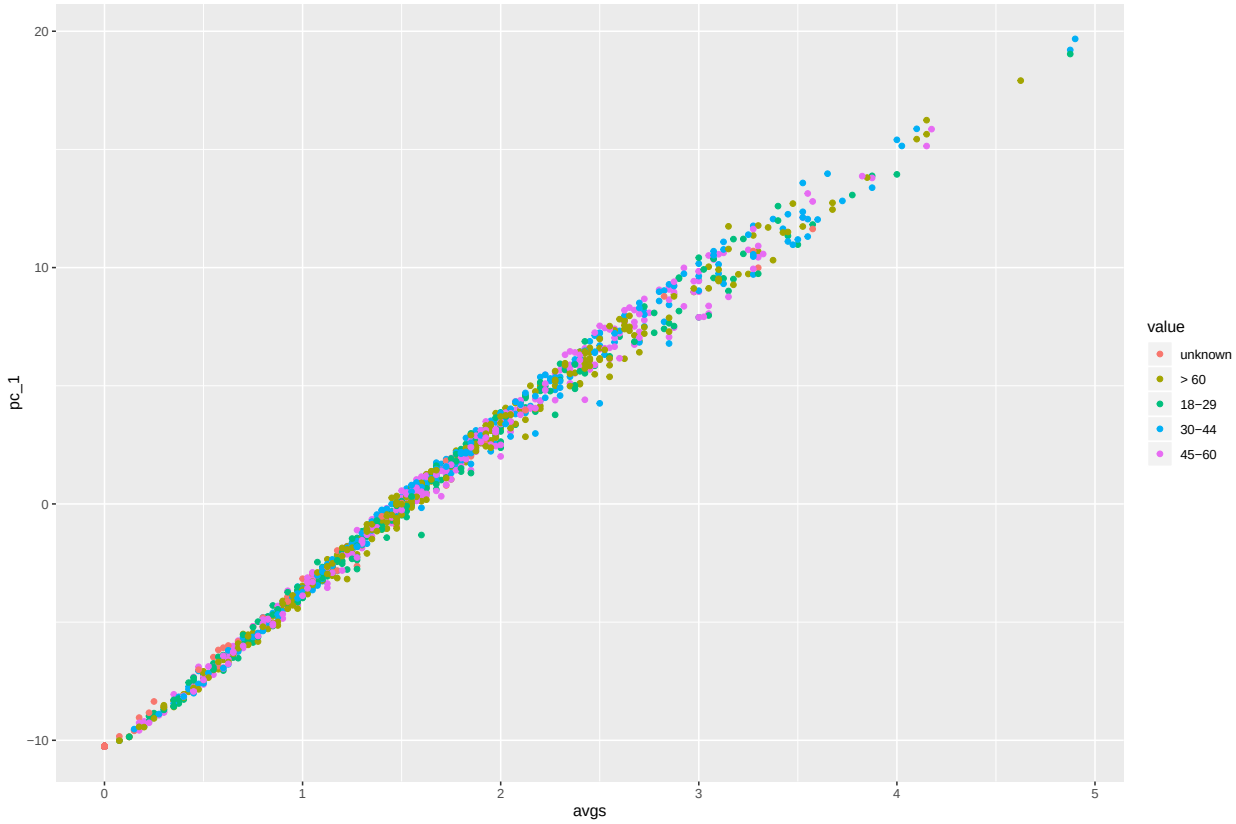
PCA

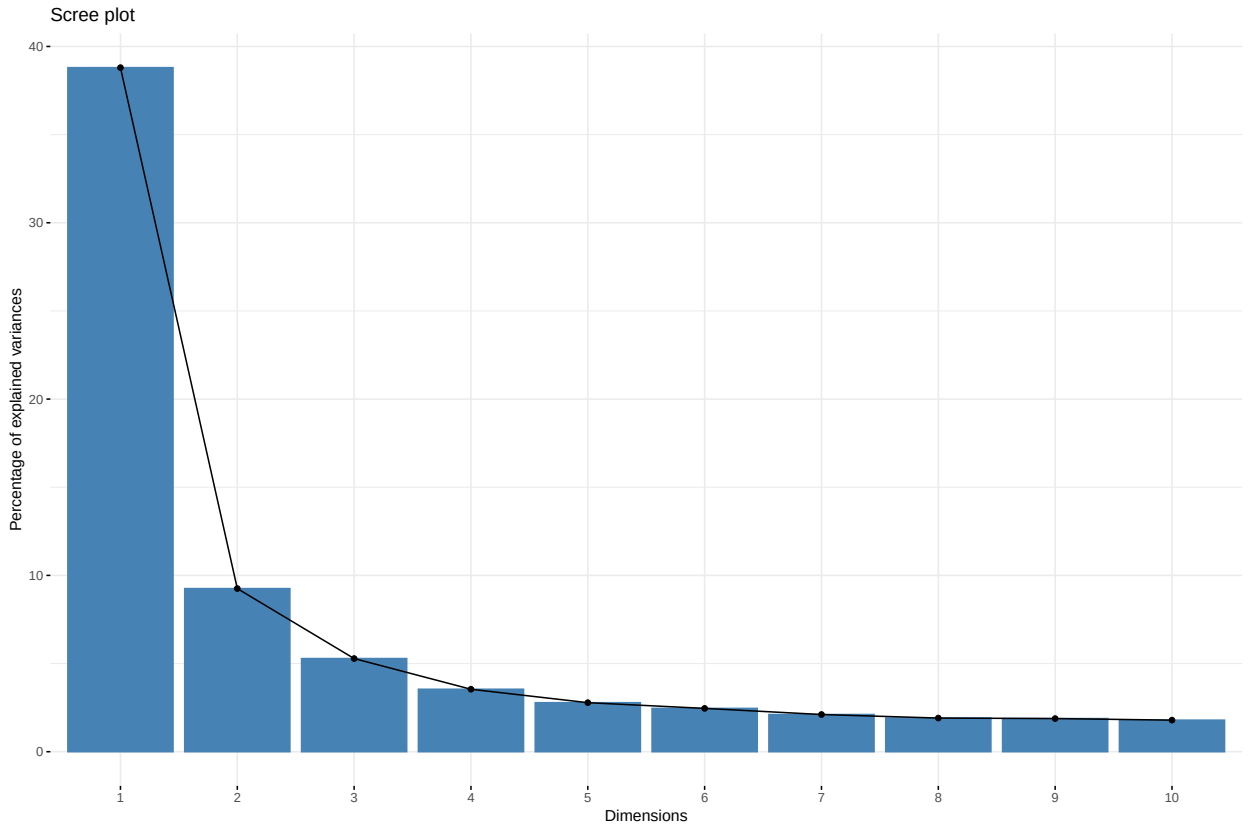
To understand the correlation behind some variables in the data, PCA is one of the most useful techniques. We try here to


10
5
0
−5
pc_2
−10
0
10
20
pc_1
age
unknown
> 60
18−29
30−44
45−60

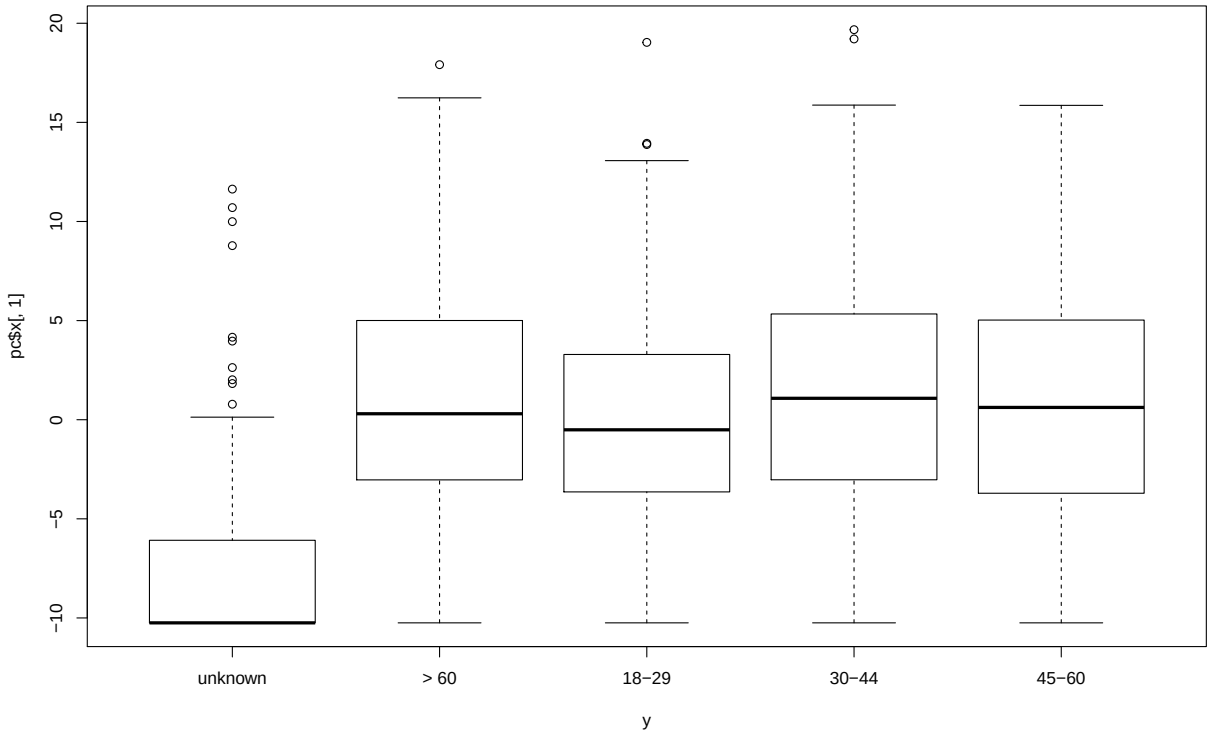

10
5
0
−5
pc_2
−10
0
10
20
pc_1
income
unknown
$0 − $24,999
$100,000 − $149,999
$150,000+
$25,000 − $49,999
$50,000 − $99,999

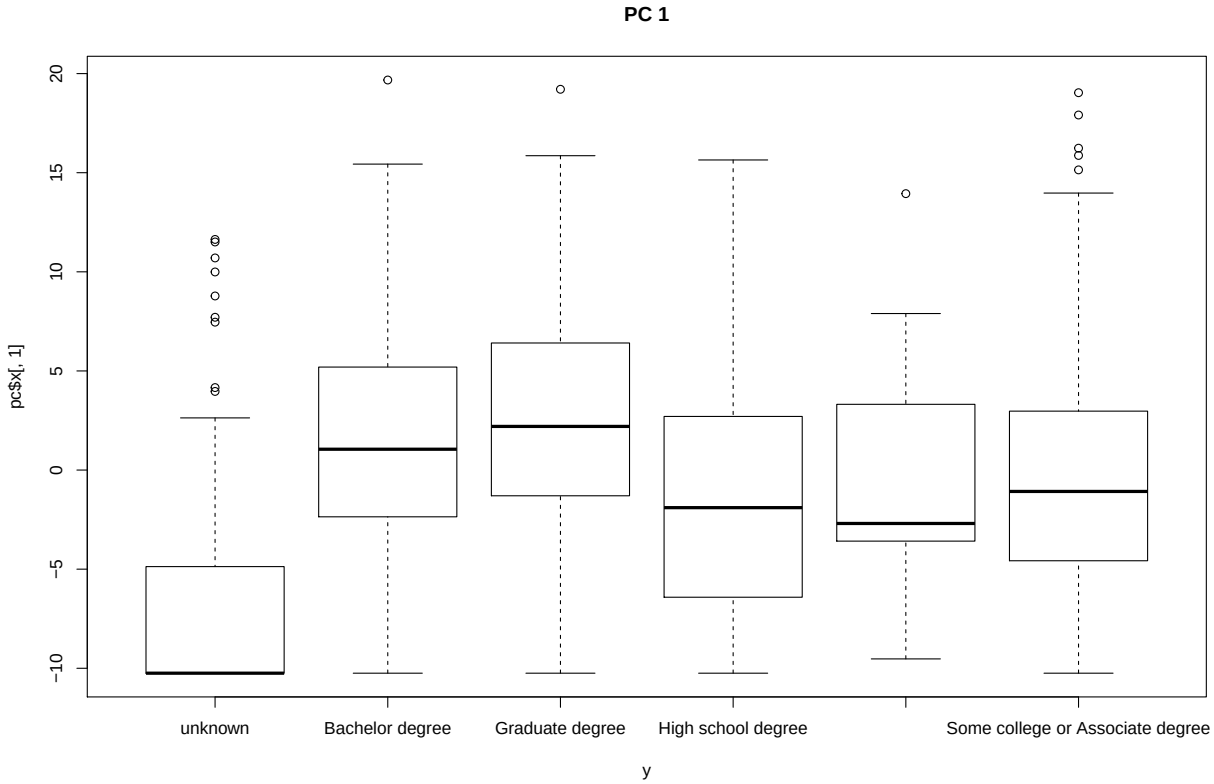

pc_2
pc_1
10
5
0
−5
−10
0
10
20
edu
unknown
Bachelor degree
Graduate degree
High school degree
Less than high school degree
Some college or Associate degree

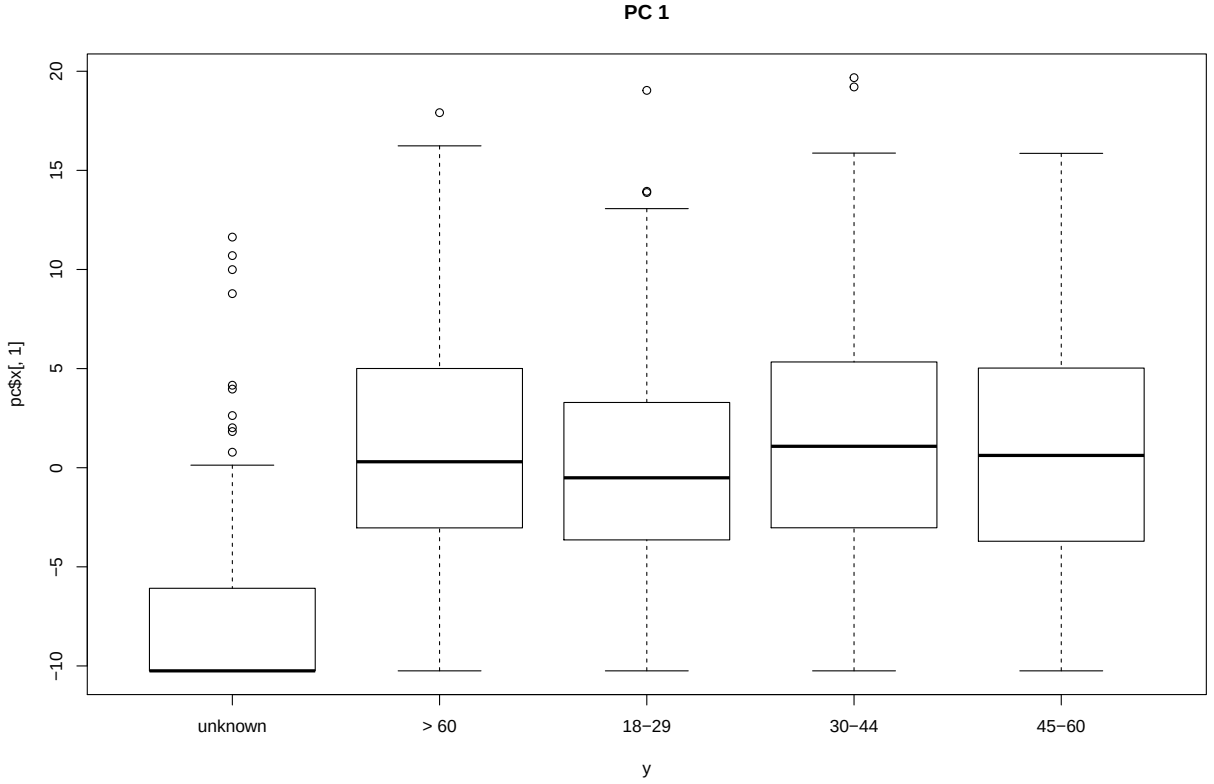

pc_1
avgs
20
10
0
−10
0
1
2
3
4
5
value
unknown
> 60
18−29
30−44
45−60

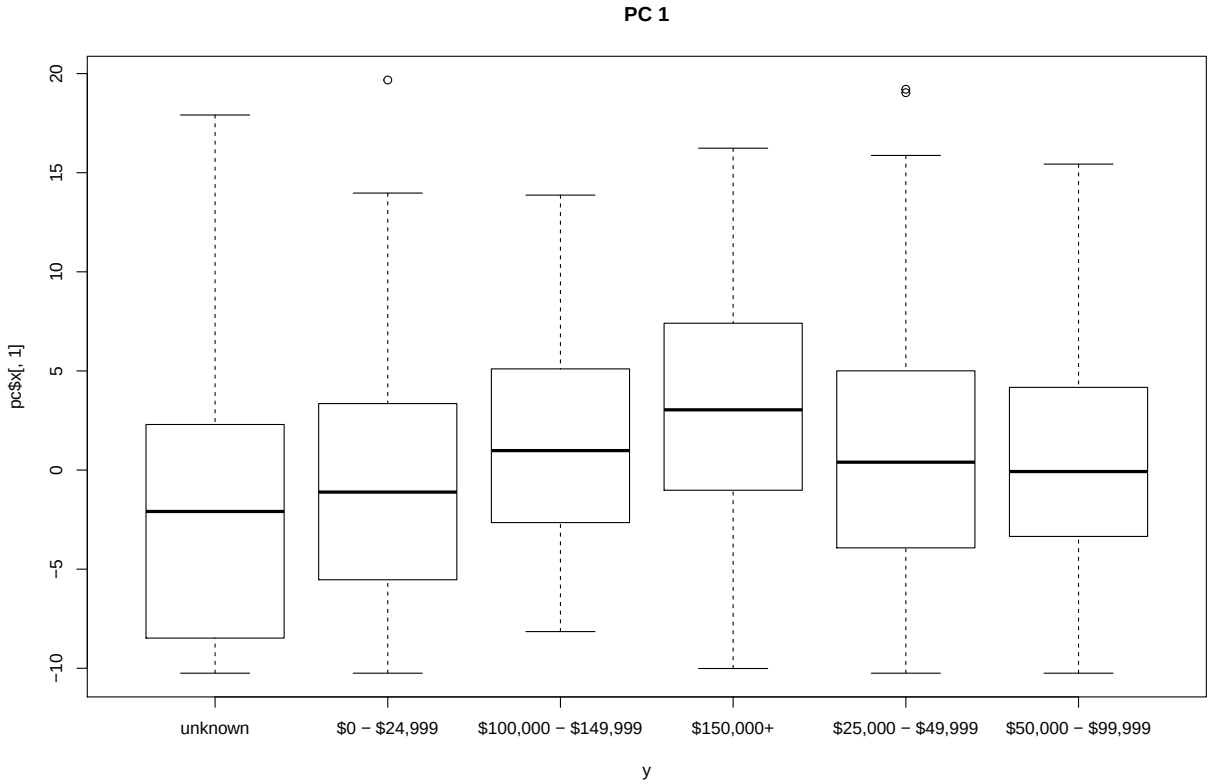

Scree plot
Percentage of explained variances
Dimensions
0
10
20
30
40
1
2
3
4
5
6
7
8
9
10

PC 1


pc$x[, 1]
y
unknown
> 60
18−29
30−44
45−60
−10
−5
0
5
10
15
20


PC 1
pc$x[, 1]
20
15
10
5
0
−5
−10
unknown
Bachelor degree
Graduate degree
High school degree
Some college or Associate degree
y


PC 1
pc$x[, 1]
20
15
10
5
0
−5
−10
unknown
> 60
18−29
30−44
45−60
y



2. Clustering

The obvious problem with this dataset is the large number of empty cells in the demographic columns which are Age, Income, Education and Region. It is understood that people sometimes don't like to enter their ages or incomes. They may just overlook these fields sometimes.

To deal with this problem, we have three different approaches:

1. Deletion of Rows with Empty Cells (or so called here "unknown")

This is the simplest yet not always useful since it leaves us with only few rows in sometimes. in our dataset, this is the number of rows we get if we delete rows with "unknown" categories

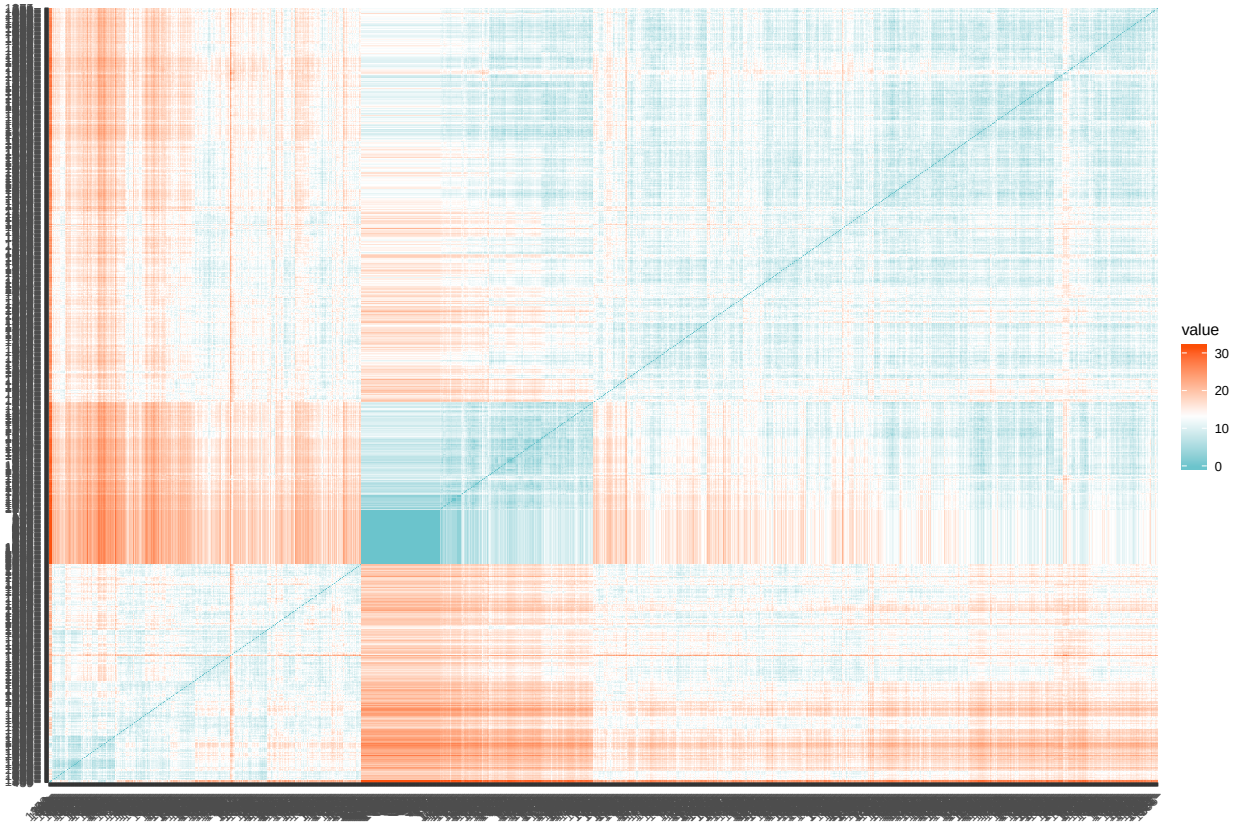
```
## [1] 946  48
```

2. Treat them as separate categories

This is a widely used approach as I read in some data science blogs and it is the simplest way to deal with the dataset without manipulation. I used this approach in this project.

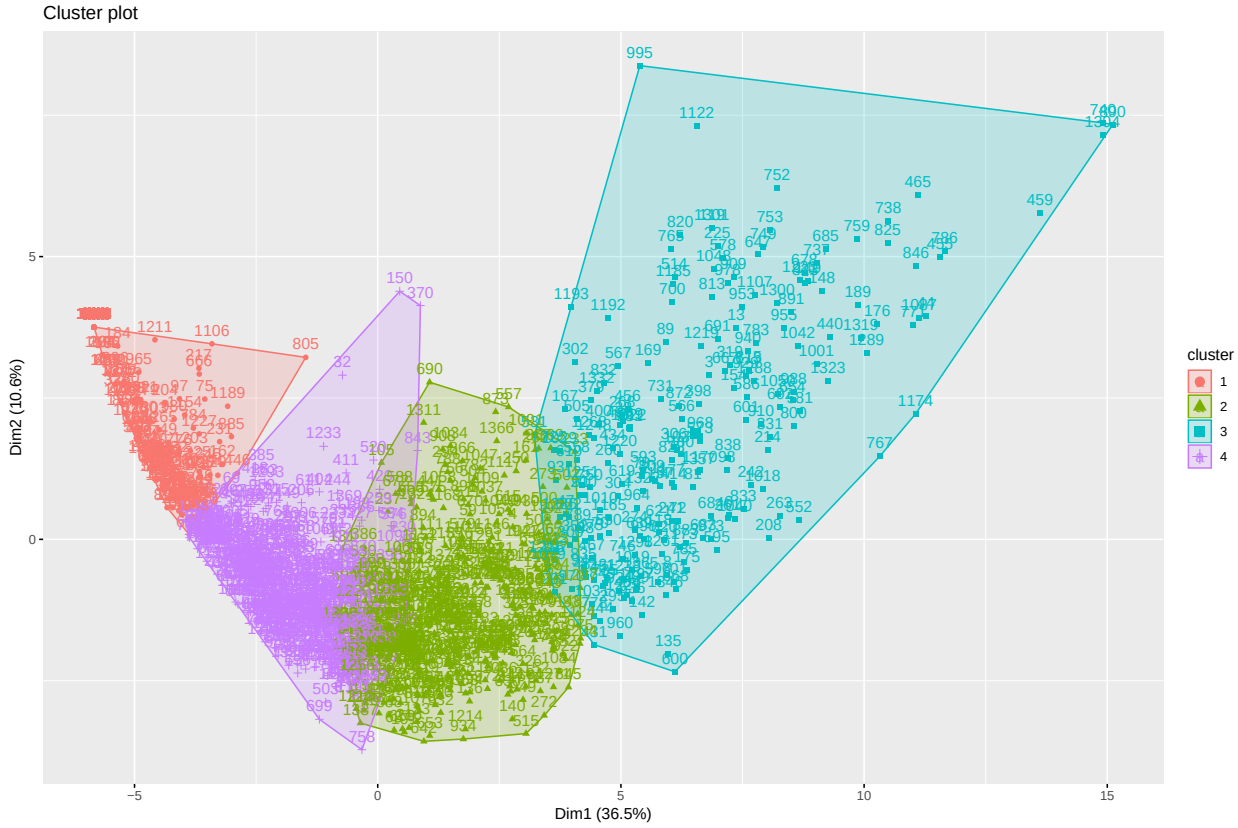
3. Perform Clustering and try to guess to which cluster they belong

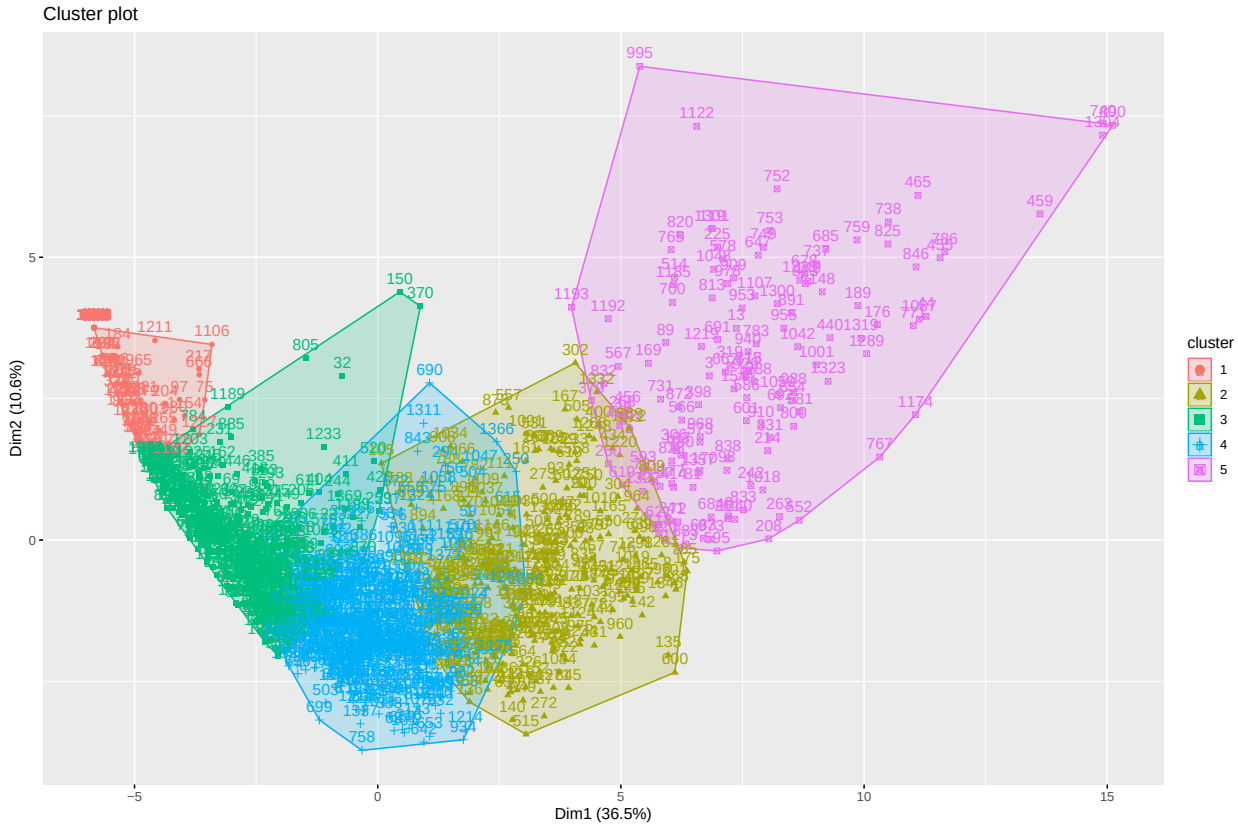
I chose to use K-means clustering for the dataset. Using the "Cluster" library. First of all, I tried to visualize the distance matrix between all the cuisines. Using the get_dist() function, I was able to compute the distance between any two users. Then, by using the fviz_dist() function, I could visualise the distance matrix.



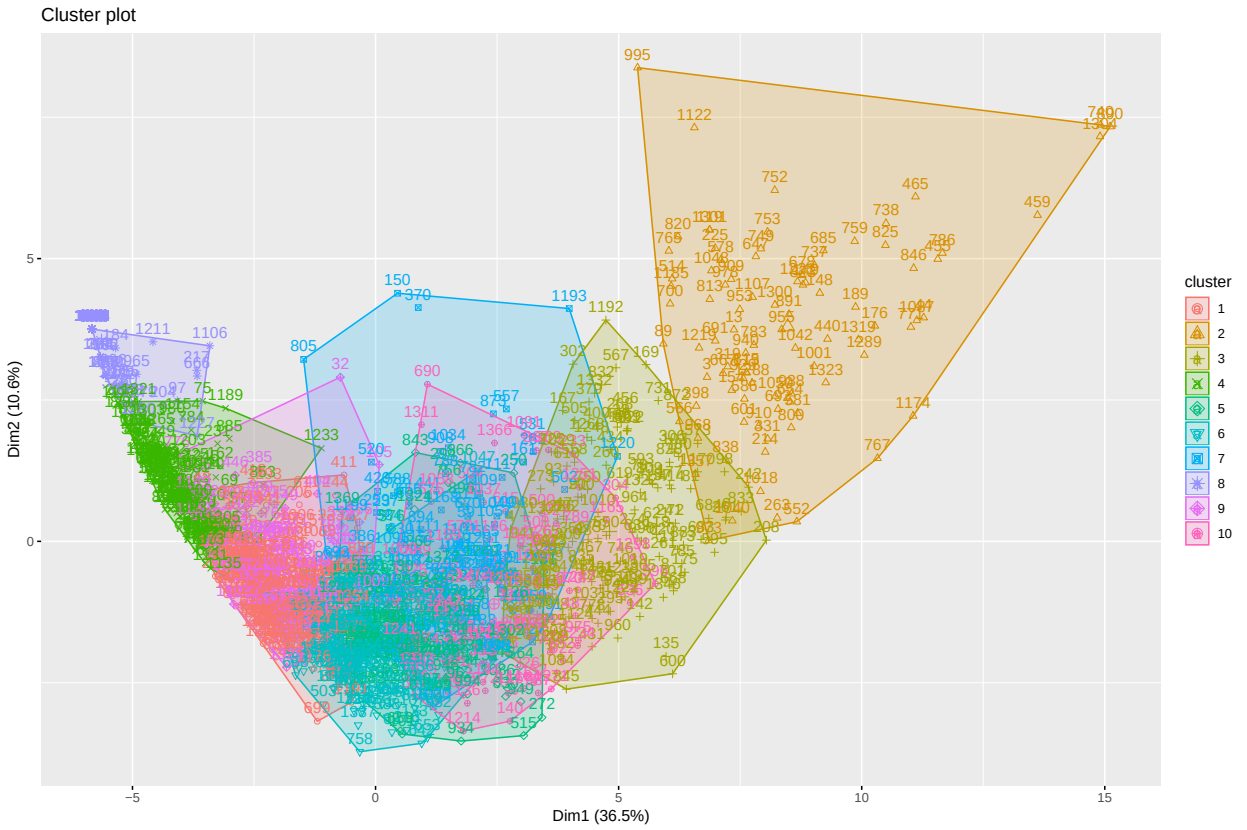
The blue color indicates low distance among any two users, while the orange mean they have a high distance value which means that they are pretty different at least in people's rating. (People who like one dislike the other or vice versa)

After performing k-means clustering for the data points and with different k values we can see the following figures with the numer of clusters (k) on the left side of the figure:






```
## Warning: did not converge in 10 iterations
```



It seems from the above figures that 5 clusters look enough.

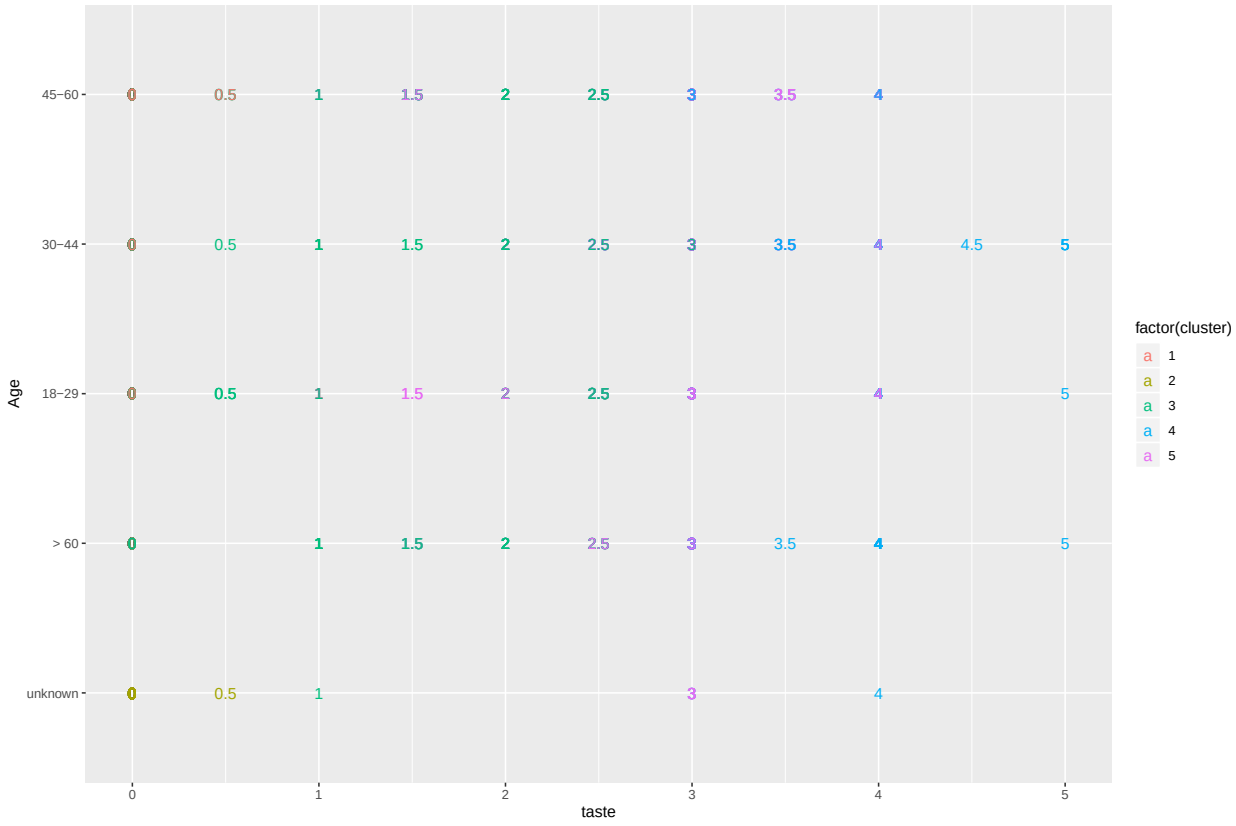
I try to plot the data points and see how they look with each cluster. I add the (taste) column here to indicate how the user evaluates and how much he knows about world cuisines. the taste is simpy the weighted median of the user's evaluations. The figures below show the results with a choice of k = 5.

```
##
##     unknown $0 - $24,999 $100,000 - $149,999 $150,000+ $25,000 - $49,999
##   1      98           50                  46        24                59
##   2     116           12                   0         1                10
##   3     114           42                  57        40                76
##   4      26           13                  13        19                27
##   5      65           21                  45        41                38
##
##     $50,000 - $99,999
##   1                85
##   2                20
##   3               120
##   4                37
##   5                58
```

This clustering can give us a clue on how to assign the "unknown" income to the right category and thus improve the prediction. This can be done in the same manner with any other categorical variable with empty cells like Education, Age and Region.

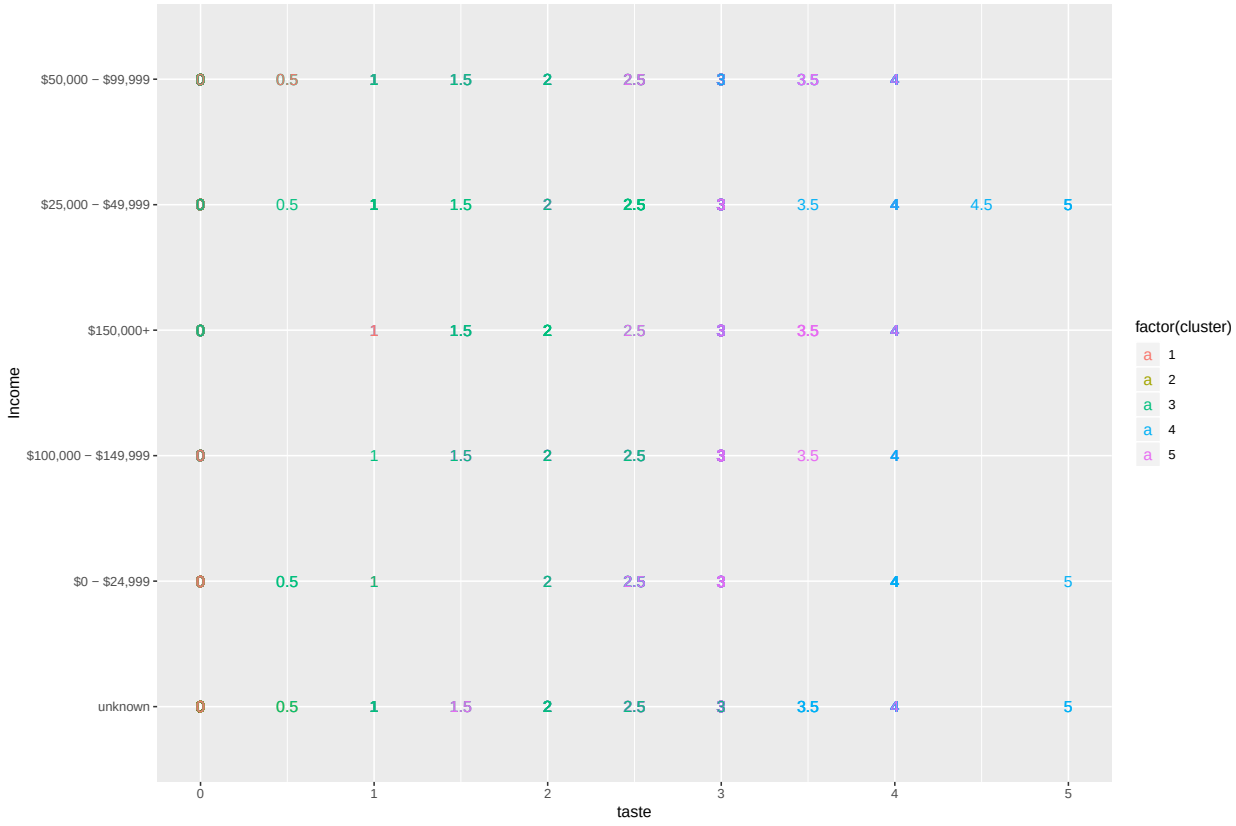
The following figures are the row weighted medians of users' ratings plotted against each one of the categorial columns which we are trying to predict.

```
MyData %>%
     as_tibble() %>%
     mutate(cluster = knn$cluster,
            taste = rowWeightedMedians(x)) %>%
     ggplot(aes(taste, Age, color = factor(cluster), label = taste)) +
     geom_text()
```

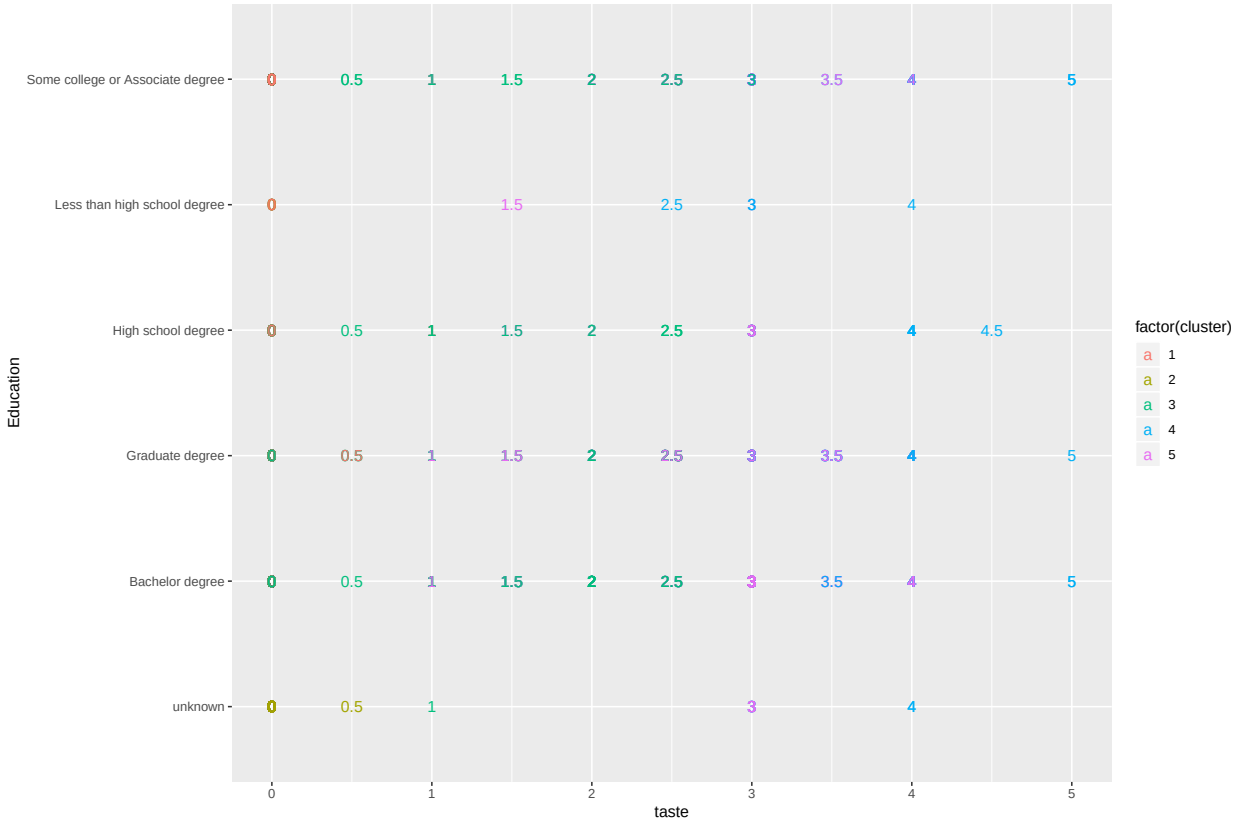



```
MyData %>%


     as_tibble() %>%
     mutate(cluster = knn$cluster,
            taste = rowWeightedMedians(x)) %>%
     ggplot(aes(taste, Income, color = factor(cluster), label = taste)) +
     geom_text()
```




```
MyData %>%
    as_tibble() %>%
    mutate(cluster = knn$cluster,
           taste = rowWeightedMedians(x)) %>%
    ggplot(aes(taste, Education, color = factor(cluster), label = taste)) +
    geom_text()
```




```
MyData %>%
    as_tibble() %>%
    mutate(cluster = knn$cluster,
           taste = rowWeightedMedians(x)) %>%
    ggplot(aes(taste, Region, color = factor(cluster), label = taste)) +
    geom_text()
```

Results

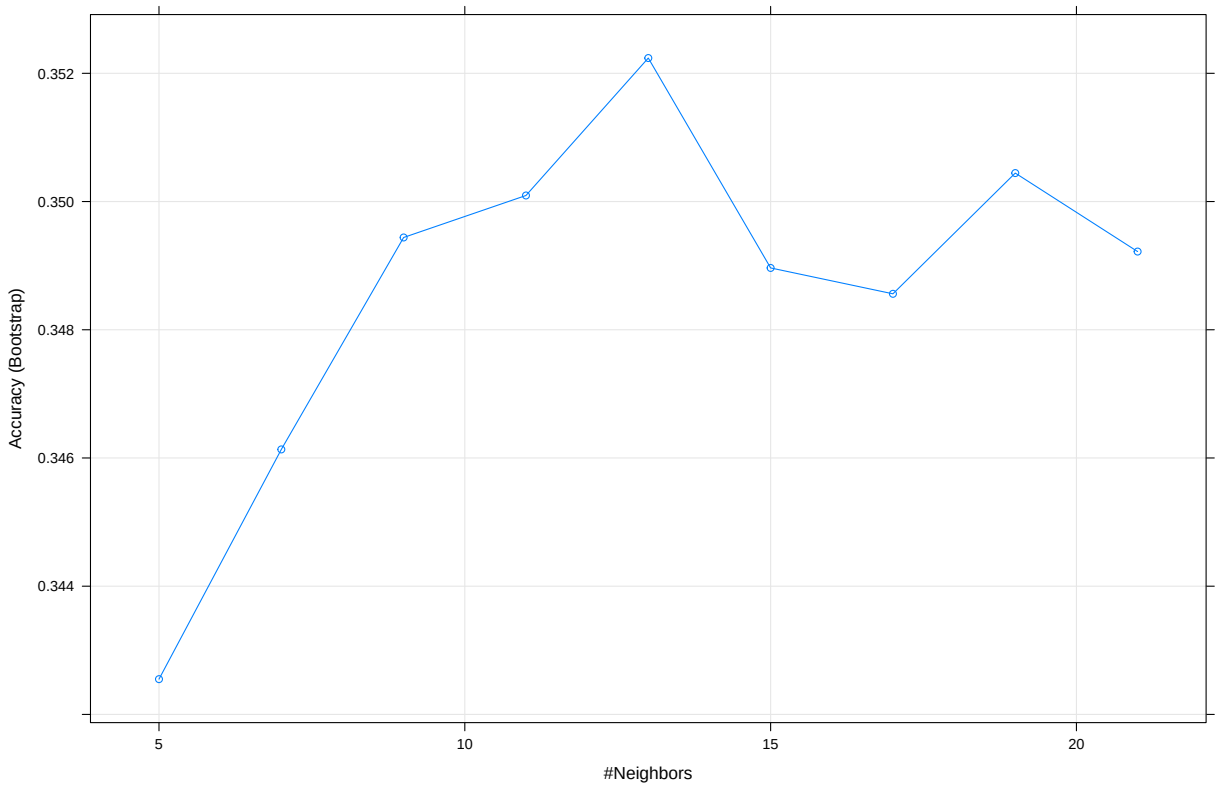
1. Classification

The main part after exploring the dataset is now to develop a machine learning supervised prediction tool on this dataset. My goal is to be able to predict the Age and gender of the user based on his ratings of the cuisines and the education and region he came from. This problem can be modeled as a classification problem; therefore, we can try some popular classification models like decision trees or random forests.

I start training by dividing the dataset into training and test set with the ratio (80-20). Then with the training set I train the model with knn and random forest. I got very low accuracy values for all the models and whith all the categorical variales that I was trying to predict. This means that the variables in the dataset are completely independent (I couldnt get more than 50% accuracy).

In the following figures, I am plotting the results for trying to predict the Age from the dataset.

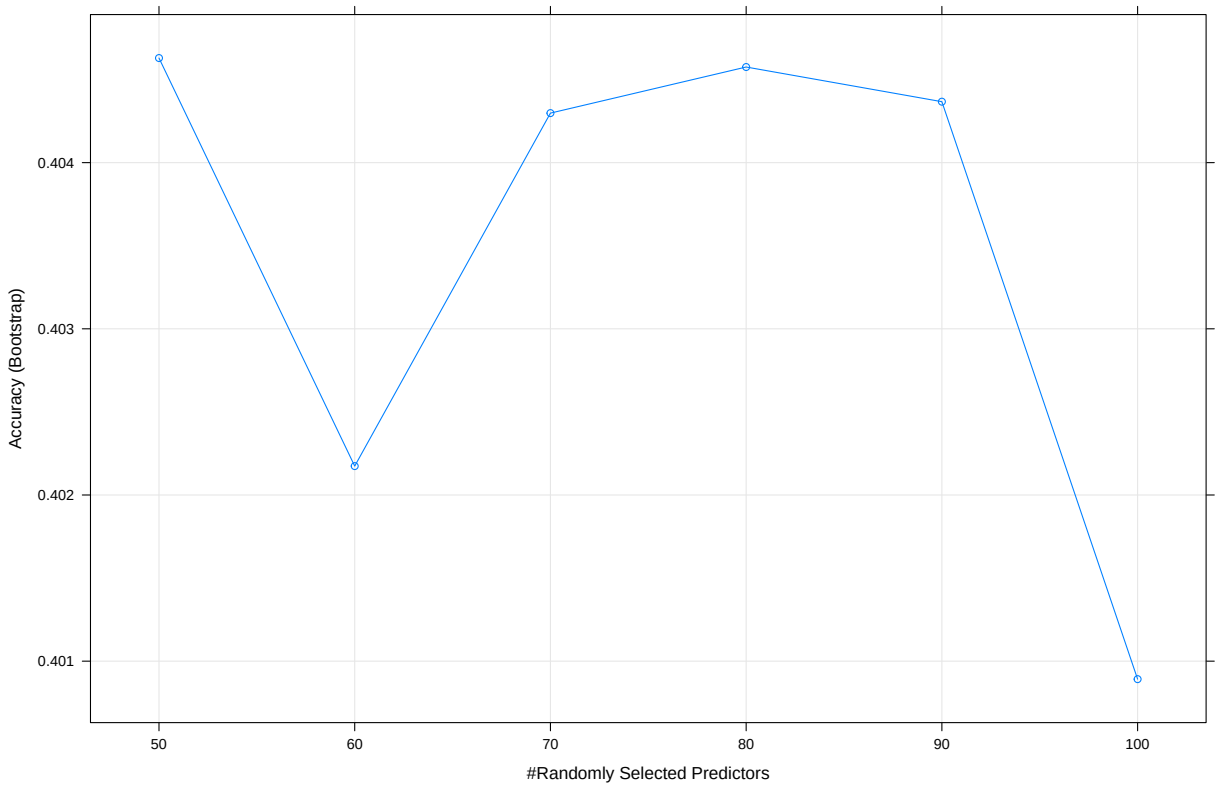
```
plot(train_knn)
```




```
confusionMatrix(data = y_hat_knn, reference = test_set$Age)$overall["Accuracy"]
```

```
##  Accuracy
## 0.3478261
```

```
 plot(train_rf)
```




```
  confusionMatrix(data = y_hat_rf, reference = test_set$Age)$overall["Accuracy"]
```

```
## Accuracy
## 0.442029
```

2. Recommendation

Another approach we can take with this dataset is to deal with cuisine evaluations to build a recommender system to predict and recommend cuisines to users based on their evaluations. I transform the ratings of cuisines into a 1/-1 values where 1 (instead of 3, 4 and 5) means the user liked the cuisine and -1 (instead of 1 and 2) means the user didn't like the cuisine. Ofcourse 0 is for the cuisines that we will predict user evaluation for them (recommend or not).

```
set.seed(1)
x <- MyData[,id]
# Convert bad ratings (1 and 2) to -1
x[x<3 & x>0] <- -1
# Convert good ratings (3,4 and 5) into 1
x[x>=3] <- 1

x <- as.matrix(x)

ratings <- as(x, "realRatingMatrix")

e <- evaluationScheme(ratings, method="split", train=0.8, given=-3)
```

```r
#3 ratings of 20% of users are excluded for testing

model_pop <- Recommender(getData(e, "train"), "POPULAR")
# I tried the popular method
prediction_pop <- predict(model_pop, getData(e, "known"), type="ratings")

rmse_popular <- calcPredictionAccuracy(prediction_pop, getData(e, "unknown"))[1]
rmse_popular
```

```
##      RMSE
## 0.4846735
```

```r
# and I tried the User-Based Collaorative Filtering (UBCF)
model <- Recommender(getData(e, "train"), method = "UBCF",
                         param=list(normalize = "center", method="Cosine", nn=50))
prediction <- predict(model, getData(e, "known"), type="ratings")

rmse_ubcf <- calcPredictionAccuracy(prediction, getData(e, "unknown"))[1]
rmse_ubcf
```

```
##      RMSE
## 0.4590306
```

Conclusion

The challenges that this dataset pose have been expressed by detail. A simple model within the time limit have been developed to understand the taste of each respondent in the survey and a model can now predict what to suggest to users in future. This was achieved with good accuracy.

In future, the project should include further investigations on how to impute the empty categories to improve the quality of the dataset. I will spend more time to achieve that because this is a very important skill for a data scientist. Another imrovement is to use an ensemble method with knn and rf together for example in order to get higher accuracy.