

# Music Preference

Raghavendran Shankar

9 April 2017

## Loading Libraries

```
#install.packages("corrplot")
#install.packages("rgl")
#install.packages("MASS")
#install.packages("cluster")
#install.packages("GGally")
library(corrplot)

## Warning: package 'corrplot' was built under R version 3.3.2

library(rgl)

## Warning: package 'rgl' was built under R version 3.3.3

library(MASS)

## Warning: package 'MASS' was built under R version 3.3.2

library(cluster)
library(GGally)

## Warning: package 'GGally' was built under R version 3.3.2
```

## Loading Survey Dataset

```
response <- read.csv("E:\\Datasets\\Survey Data\\responses.csv")
```

## Choosing the Music Preference variables across the survey to find music preference

```
song_pref <- response[1:19]
song_pref[is.na(song_pref)] <- 0 # Replacing NA values by 0
str(song_pref)

## 'data.frame': 1010 obs. of 19 variables:
## $ Music : num 5 4 5 5 5 5 5 5 5 5 ...
## $ Slow.songs.or.fast.songs: num 3 4 5 3 3 3 5 3 3 3 ...
## $ Dance : num 2 2 2 2 4 2 5 3 3 2 ...
## $ Folk : num 1 1 2 1 3 3 3 2 1 5 ...
## $ Country : num 2 1 3 1 2 2 1 1 1 2 ...
## $ Classical.music : num 2 1 4 1 4 3 2 2 2 2 ...
## $ Musical : num 1 2 5 1 3 3 2 2 4 5 ...
## $ Pop : num 5 3 3 2 5 2 5 4 3 3 ...
## $ Rock : num 5 5 5 2 3 5 3 5 5 5 ...
```

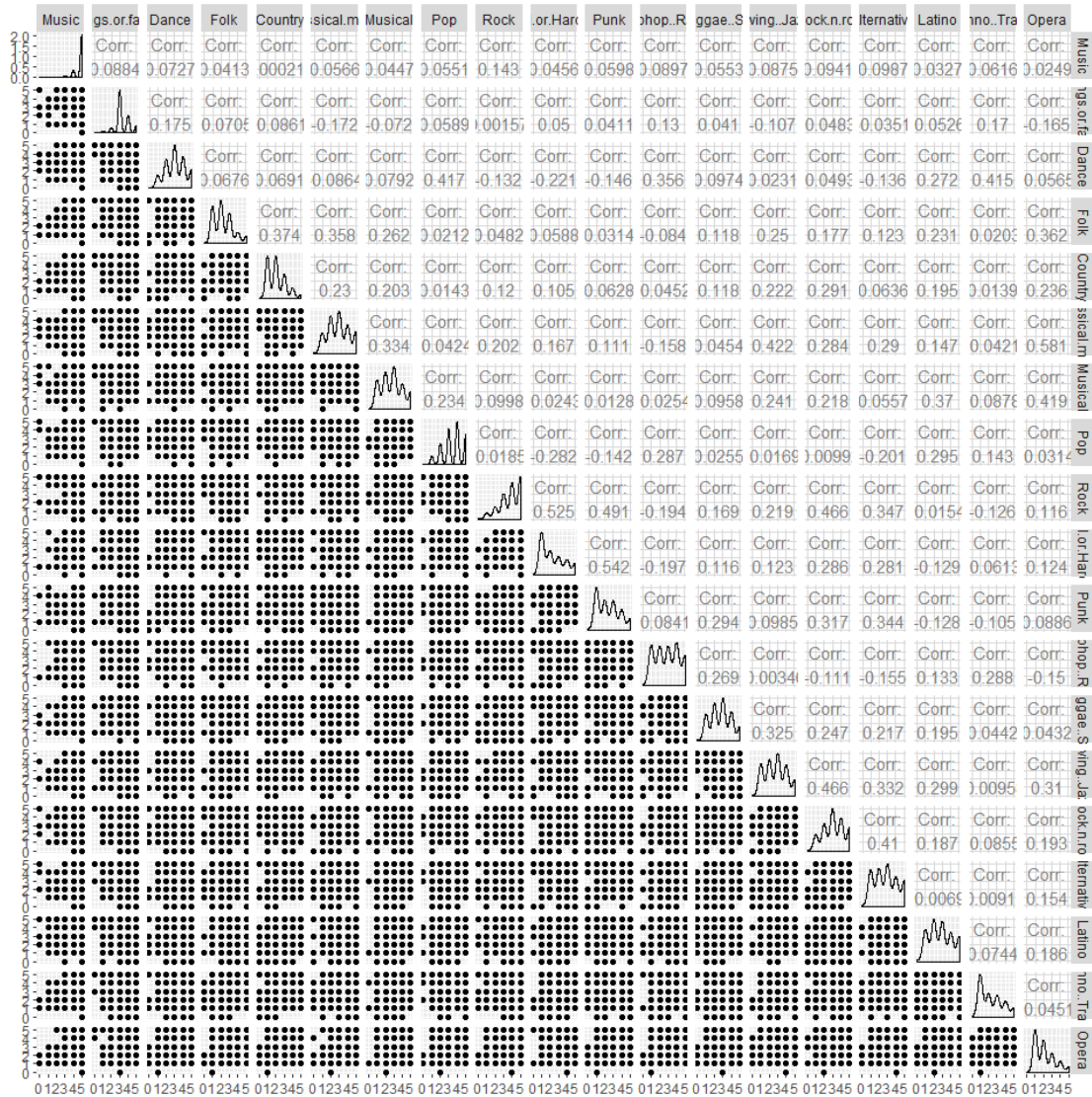
```
## $ Metal.or.Hardrock      : num  1 4 3 1 1 5 1 1 5 2 ...
## $ Punk                   : num  1 4 4 4 2 3 1 2 1 3 ...
## $ Hiphop..Rap            : num  1 1 1 2 5 4 3 3 1 2 ...
## $ Reggae..Ska            : num  1 3 4 2 3 3 1 2 2 4 ...
## $ Swing..Jazz            : num  1 1 3 1 2 4 1 2 2 4 ...
## $ Rock.n.roll            : num  3 4 5 2 1 4 2 3 2 4 ...
## $ Alternative            : num  1 4 5 5 2 5 3 1 0 4 ...
## $ Latino                 : num  1 2 5 1 4 3 3 2 1 5 ...
## $ Techno..Trance         : num  1 1 1 2 2 1 5 3 1 1 ...
## $ Opera                  : num  1 1 3 1 2 3 2 2 1 2 ...
```

```
head(song_pref,10) # Displaying the first 10 rows
```

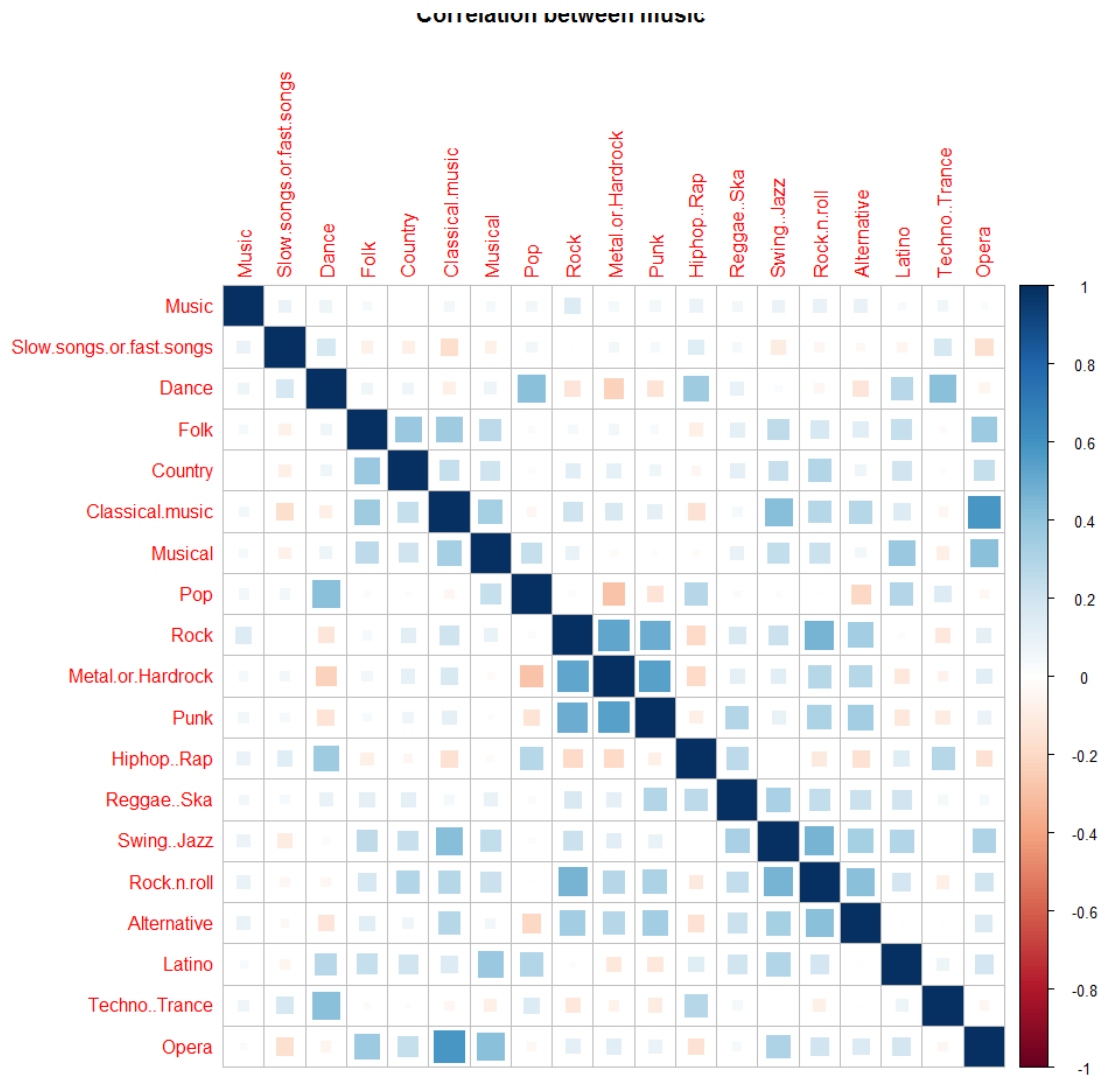
```
##      Music Slow.songs.or.fast.songs Dance Folk Country Classical.music
## 1      5                        3      2      1              2              2
## 2      4                        4      2      1              1              1
## 3      5                        5      2      2              3              4
## 4      5                        3      2      1              1              1
## 5      5                        3      4      3              2              4
## 6      5                        3      2      3              2              3
## 7      5                        5      5      3              1              2
## 8      5                        3      3      2              1              2
## 9      5                        3      3      1              1              2
## 10     5                        3      2      5              2              2
##      Musical Pop Rock Metal.or.Hardrock Punk Hiphop..Rap Reggae..Ska
## 1      1      5      5                        1      1              1              1
## 2      2      3      5                        4      4              1              3
## 3      5      3      5                        3      4              1              4
## 4      1      2      2                        1      4              2              2
## 5      3      5      3                        1      2              5              3
## 6      3      2      5                        5      3              4              3
## 7      2      5      3                        1      1              3              1
## 8      2      4      5                        1      2              3              2
## 9      4      3      5                        5      1              1              2
## 10     5      3      5                        2      3              2              4
##      Swing..Jazz Rock.n.roll Alternative Latino Techno..Trance Opera
## 1      1                        3      1      1              1              1
## 2      1                        4      4      2              1              1
## 3      3                        5      5      5              1              3
## 4      1                        2      5      1              2              1
## 5      2                        1      2      4              2              2
## 6      4                        4      5      3              1              3
## 7      1                        2      3      3              5              2
## 8      2                        3      1      2              3              2
## 9      2                        2      0      1              1              1
## 10     4                        4      4      5              1              2
```

## Finding Correlation between types of music

```
ggpairs(song_pref)
```



```
M <- cor(song_pref)
corrplot(M,method = "square",main = 'Correlation between music')
```

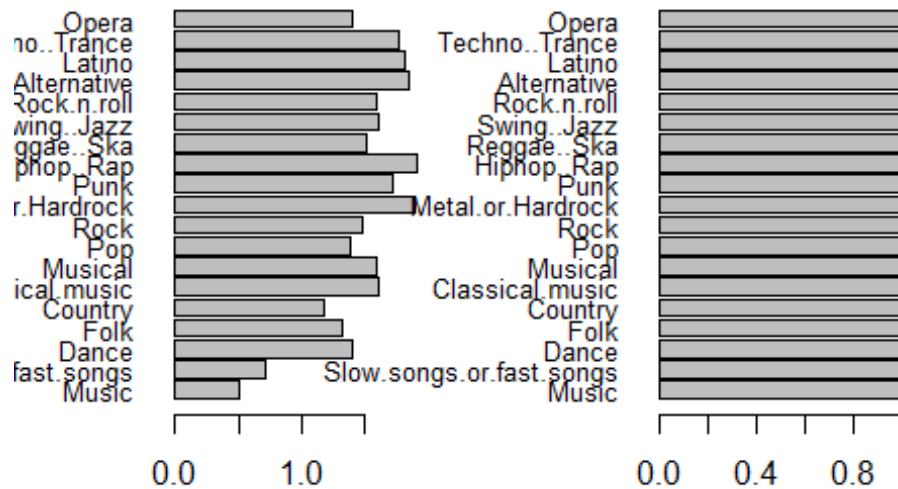


# From the corrplot, it is seen that people who rated much Likes for Rock also likes Punk, Metal or Hardrock, Rock n Roll. People who Liked Classical Music also Liked Opera; people who Liked Pop also Liked Dance and Hiphop and Rap. People who Liked Alternative Music also Liked Rock, Metal and Punk.

## Scaling the variables to have uniform variance for analysis

```
par(mfrow=c(1,2))
barplot(sapply(song_pref,var),horiz = T,las = 1,cex.names = 0.8,main = 'Non-uniform variability across the music types')
song_scaled <- data.frame(scale(song_pref))
barplot(sapply(song_scaled,var),horiz = T,las = 1,cex.names = 0.8,main = 'Uniformity across the music types')
```

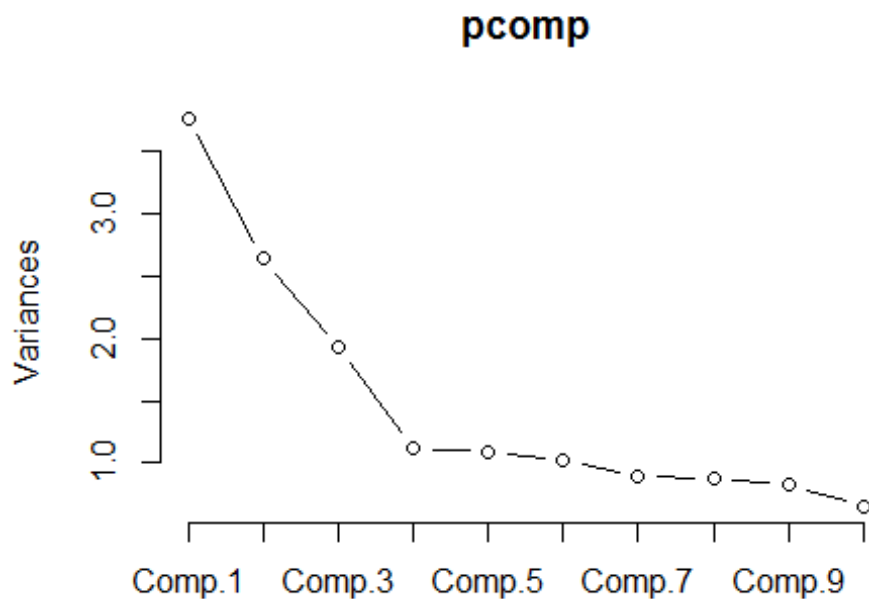
## form variability across the uniformity across the music



*# From the plot, it is seen that scaling makes the data uniform on a single scale to correctly compute scores from Principal Component Analysis(PCA)*

## Perform Principal Component Analysis to compute scores of each people according to their music preferences

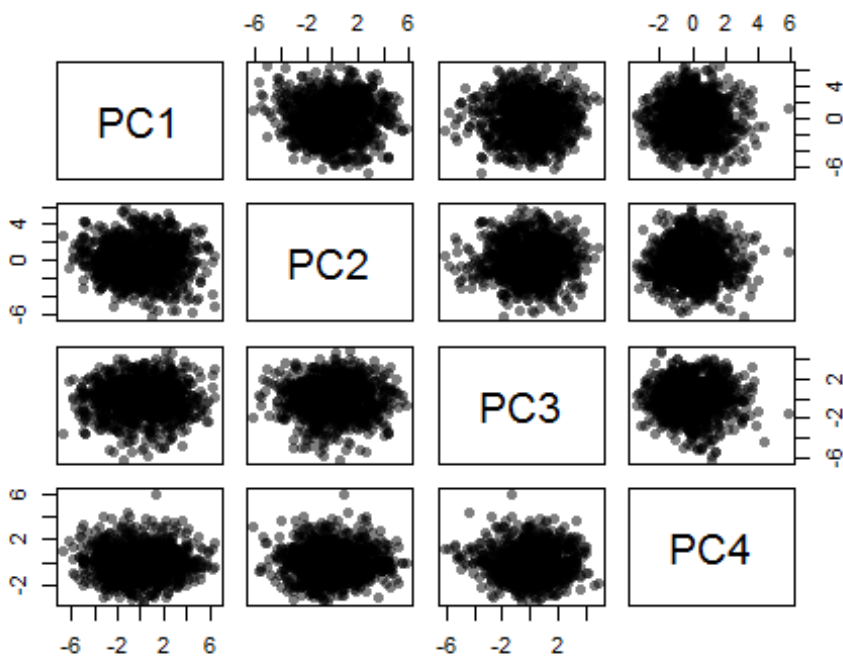
```
pcomp <- princomp(x = song_scaled)
plot(pcomp, type = 'l')
```



```
summary(pcomp)
```

```
## Importance of components:
##               Comp.1   Comp.2   Comp.3   Comp.4   Comp.5
## Standard deviation  1.9373906 1.6222463 1.3858967 1.05819280 1.04484604
## Proportion of Variance 0.1977475 0.1386469 0.1011902 0.05899378 0.05751501
## Cumulative Proportion 0.1977475 0.3363944 0.4375846 0.49657835 0.55409336
##               Comp.6   Comp.7   Comp.8   Comp.9
## Standard deviation  1.00825429 0.94753020 0.93525239 0.91036005
## Proportion of Variance 0.05355706 0.04730017 0.04608231 0.04366194
## Cumulative Proportion 0.60765043 0.65495060 0.70103291 0.74469485
##               Comp.10  Comp.11  Comp.12  Comp.13
## Standard deviation  0.80895737 0.79996696 0.7661622 0.70950475
## Proportion of Variance 0.03447687 0.03371481 0.0309256 0.02652084
## Cumulative Proportion 0.77917172 0.81288653 0.8438121 0.87033297
##               Comp.14  Comp.15  Comp.16  Comp.17
## Standard deviation  0.68473274 0.68091054 0.65008875 0.63411245
## Proportion of Variance 0.02470124 0.02442625 0.02226496 0.02118406
## Cumulative Proportion 0.89503421 0.91946045 0.94172541 0.96290947
##               Comp.18  Comp.19
## Standard deviation  0.60455120 0.58184203
## Proportion of Variance 0.01925497 0.01783556
## Cumulative Proportion 0.98216444 1.00000000
```

```
pc <- prcomp(song_pref)
comp <- data.frame(pc$x[,1:4])
plot(comp, pch=16, col=rgb(0,0,0,0.5))
```



```
plot3d(comp$PC1, comp$PC2, comp$PC3)
```

```
head(comp$PC1, 10)
```

```
## [1] -2.8356672  0.7012355  4.6426467 -2.1685545 -2.0067212  3.5840044
## [7] -3.5109857 -2.0832440 -1.2694981  3.0448873
```

*# The scores for people are calculated according to their music preference. Person 976 gave high ratings (5) to more types of music and person 488 gave low or no ratings for most of the music types.*

## Finding optimal number of Cluster groups by MultiDimensional Scaling

```
corr <- cor(song_pref, use = "pairwise.complete")
```

```
d1 <- 1 - abs(corr)
```

```
m1 <- isoMDS(d1, k=1)
```

```
## initial value 46.791257
```

```
## iter 5 value 43.196286
```

```
## iter 5 value 43.154571
```

```
## final value 42.791662
```

```
## converged
```

```
m2 <- isoMDS(d1, k=2)
```

```
## initial value 28.792650
```

```
## iter 5 value 23.091496
```

```
## iter 10 value 21.574851
## iter 15 value 20.712091
## final value 20.597135
## converged
```

```
m3 <- isoMDS(d1,k = 3)
```

```
## initial value 19.845447
## iter 5 value 13.201871
## iter 10 value 13.043078
## iter 10 value 13.041779
## iter 10 value 13.041779
## final value 13.041779
## converged
```

```
m4 <- isoMDS(d1,k = 4)
```

```
## initial value 15.805226
## iter 5 value 10.103091
## iter 10 value 9.576352
## iter 15 value 9.454134
## iter 15 value 9.451235
## final value 9.405361
## converged
```

```
m5 <- isoMDS(d1,k = 5)
```

```
## initial value 11.159101
## iter 5 value 7.166341
## iter 10 value 6.878580
## iter 10 value 6.875312
## iter 10 value 6.870301
## final value 6.870301
## converged
```

```
m6 <- isoMDS(d1,k = 6)
```

```
## initial value 9.051266
## iter 5 value 5.722641
## iter 10 value 5.476304
## iter 15 value 5.308863
## iter 20 value 5.180768
## iter 25 value 5.061131
## final value 5.021239
## converged
```

```
m7 <- isoMDS(d1,k = 7)
```

```
## initial value 6.682884
## iter 5 value 4.592642
## iter 10 value 4.435990
## iter 15 value 4.358908
```



```

## final value 4.331576
## converged

m8 <- isoMDS(d1,k = 8)

## initial value 4.345403
## iter 5 value 3.334247
## iter 10 value 3.200544
## iter 15 value 3.106903
## iter 20 value 2.934833
## iter 25 value 2.876290
## iter 30 value 2.718178
## iter 35 value 2.633435
## iter 40 value 2.562814
## iter 45 value 2.523962
## iter 45 value 2.521973
## iter 45 value 2.521312
## final value 2.521312
## converged

m9 <- isoMDS(d1,k = 9)

## initial value 3.278420
## iter 5 value 2.535457
## iter 10 value 2.104445
## iter 15 value 1.911082
## iter 20 value 1.810801
## iter 25 value 1.769690
## iter 30 value 1.754760
## iter 35 value 1.740696
## final value 1.733753
## converged

m10 <- isoMDS(d1,k = 10)

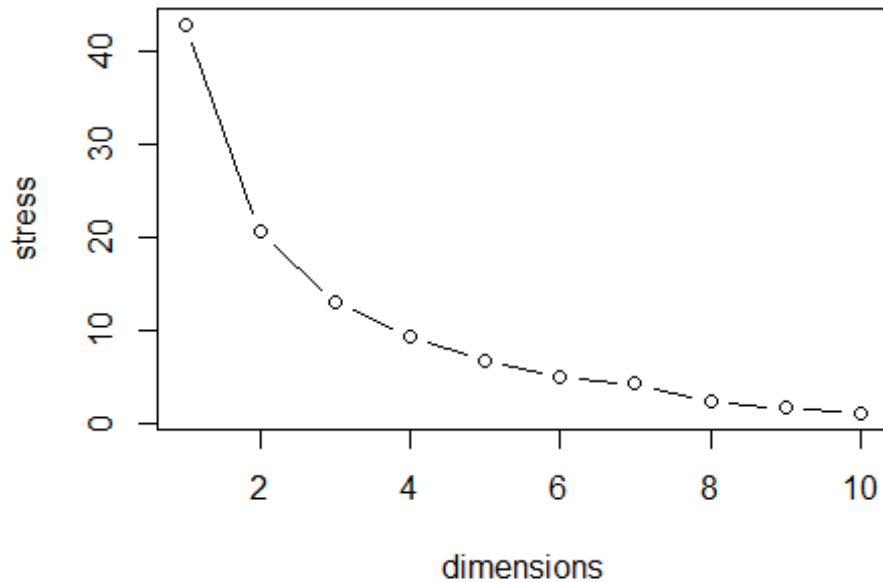
## initial value 2.510726
## iter 5 value 1.727957
## iter 10 value 1.477324
## iter 15 value 1.400163
## iter 20 value 1.334455
## iter 25 value 1.259512
## iter 30 value 1.219989
## iter 35 value 1.206281
## iter 35 value 1.205833
## iter 35 value 1.205678
## final value 1.205678
## converged

stress <-
c(m1$stress,m2$stress,m3$stress,m4$stress,m5$stress,m6$stress,m7$stress,m8$st
ress,m9$stress,m10$stress)

```

```
dimensions <- 1:10
```

```
plot(dimensions, stress, type = 'b')
```



*# From the stress plot, it is seen that from stress point 5 to 6 there is not much difference and the stress is low. So, 6 clusters were taken for cluster analysis.*

## Clara Clustering to group people by their Music Preference

```
c1 <- clara(song_pref, k = 6)
```

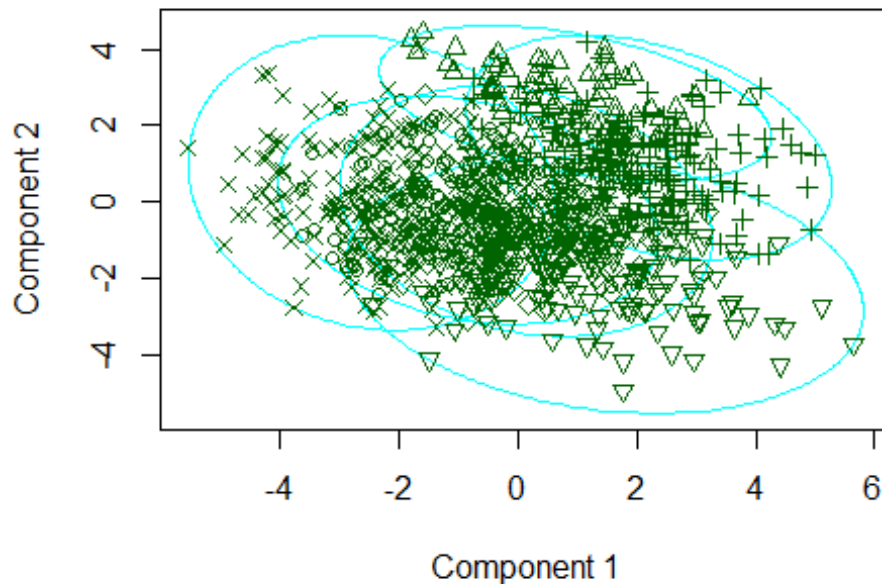
```
c1$clustering
```

```
##      [1] 1 2 3 4 4 3 1 1 5 6 1 3 3 5 2 2 1 5 6 3 3 2 4 5 3 4 6 1 4 6 6 4 1 3
##     [35] 3 5 5 2 6 2 4 6 1 1 6 6 5 1 4 2 5 1 6 6 6 6 3 3 1 2 3 6 6 4 5 4 1 4
##     [69] 2 5 4 4 5 4 4 2 4 1 6 6 6 6 3 5 3 5 6 1 1 3 5 2 3 5 4 6 2 4 1 4 6 2
##    [103] 3 6 6 3 6 2 3 5 1 6 1 1 3 1 2 4 2 1 3 1 3 3 3 3 4 6 1 1 3 4 1 3 6 2
##    [137] 4 6 4 6 1 1 4 4 3 1 4 4 6 4 3 2 6 6 3 6 1 4 1 6 2 1 6 1 1 4 3 6 1 3
##    [171] 1 4 3 5 4 3 2 3 3 6 2 6 3 6 3 4 5 1 5 3 3 5 5 3 2 4 6 4 6 4 5 1 4 1
##    [205] 3 3 3 5 6 6 6 6 3 6 3 4 2 5 4 4 6 6 5 4 3 3 4 3 6 6 5 5 1 1 1 3 5 4
##    [239] 4 3 4 4 5 3 5 2 6 3 5 6 4 6 5 4 5 6 4 6 3 6 6 4 6 1 6 4 5 6 1 3 2 2
##    [273] 3 6 4 6 1 1 6 4 1 6 4 6 4 5 6 1 1 5 5 6 5 3 3 6 1 1 6 4 3 3 5 5 5 3
##    [307] 6 3 1 4 1 6 1 5 6 4 4 1 4 6 1 5 2 5 1 6 1 5 2 1 3 6 4 5 4 6 1 6 5 3
##    [341] 5 4 6 5 5 1 4 4 6 1 1 3 2 4 5 4 6 4 6 5 6 6 3 3 2 2 4 2 4 4 5 6 1 4
##    [375] 1 4 4 3 4 6 1 3 4 3 2 6 1 4 6 4 1 4 4 4 1 1 6 4 6 3 1 5 1 6 6 6 6 6
##    [409] 6 5 2 5 1 1 1 1 1 2 5 6 2 5 4 3 4 3 2 3 5 2 1 6 2 5 1 2 1 4 4 1 1 4
##    [443] 5 6 2 4 4 4 1 5 5 1 1 4 2 1 4 1 6 3 4 6 1 4 1 4 5 4 6 1 4 2 5 2 1 5
##    [477] 3 3 1 5 6 3 4 1 4 1 1 4 3 2 4 5 3 4 3 2 5 3 1 1 6 6 3 5 4 2 4 6 2 4
```

```
## [511] 6 4 4 1 5 3 6 5 6 4 4 6 6 1 2 6 4 1 5 3 1 2 1 3 4 5 4 3 1 3 6 4 6 1
## [545] 3 1 3 3 2 6 3 4 3 4 5 4 4 1 5 4 6 6 4 1 4 5 4 4 1 4 4 4 4 1 4 6 4 4
## [579] 5 1 4 1 6 5 1 6 1 2 4 2 3 4 4 3 3 2 1 1 6 1 1 3 1 6 4 5 1 5 6 6 3 1
## [613] 6 6 1 3 1 5 6 5 1 1 6 4 5 4 6 4 2 6 1 5 6 4 6 5 6 5 5 2 4 6 6 5 4 6
## [647] 5 5 5 6 4 6 1 3 1 1 3 4 4 3 2 4 4 3 5 2 2 4 3 6 4 6 5 5 4 5 4 4 3 2
## [681] 6 6 6 5 4 5 6 3 5 1 1 6 6 1 6 6 5 5 6 4 4 6 3 6 1 6 5 3 4 6 4 1 5 6
## [715] 6 3 6 3 3 1 6 3 6 5 5 1 4 1 4 6 2 6 4 6 5 4 5 3 4 3 6 3 6 2 3 1 3 6
## [749] 6 2 6 3 6 3 1 2 1 6 4 6 1 4 4 3 5 2 6 6 1 6 4 3 4 6 4 6 1 1 4 6 1 5
## [783] 3 5 6 6 3 4 4 6 2 1 4 6 4 5 4 4 1 3 4 1 1 3 5 3 4 5 1 4 3 3 4 4 6 6
## [817] 3 1 1 4 3 3 4 4 3 3 3 3 6 3 6 6 3 3 3 1 3 3 1 5 5 1 1 6 6 4 6 3 2 5
## [851] 4 4 4 5 4 6 6 1 2 6 6 1 1 1 6 5 3 1 4 3 1 3 1 6 4 4 4 2 6 1 4 3 1 1
## [885] 1 4 2 6 1 3 1 1 5 6 1 4 3 1 3 4 5 1 4 2 5 6 6 4 1 6 1 3 1 3 3 6 2 2
## [919] 4 3 5 1 1 3 4 5 3 1 6 6 4 5 5 3 1 4 1 3 6 1 1 1 1 4 5 6 3 4 5 1 2 1
## [953] 1 5 6 2 5 4 4 3 3 6 5 6 3 1 3 2 6 2 6 2 3 3 2 6 5 1 2 5 3 3 2 1 2 2
## [987] 6 2 3 6 1 2 1 1 2 4 4 4 1 6 4 4 2 6 6 6 1 4 1 4
```

```
plot(c1)
```

**clusplot(clara(x = song\_pref, k = 6))**



These two components explain 33.64 % of the point variab

**Silhouette plot of clara(x = song\_pref, k = 6)**

n = 52

6 clusters  $C_j$

$j: n_j | \text{ave}_{i \in C_j} s_i$   
1: 8 | 0.04

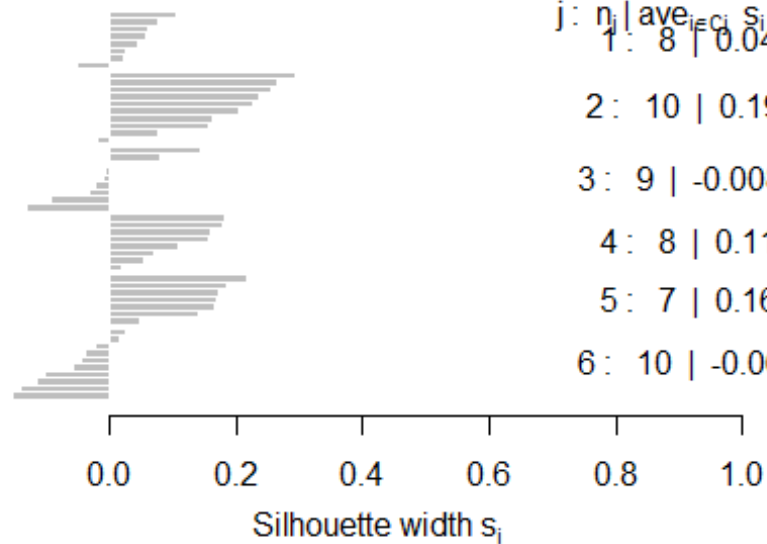
2: 10 | 0.19

3: 9 | -0.008

4: 8 | 0.11

5: 7 | 0.16

6: 10 | -0.06



Average silhouette width : 0.07

# Conclusion: There are 6 clusters which denotes 6 types of people and their different attitudes towards music preferences. The comparison between the first component score of PCA analysis and the clusters gives the same result. # Here, the higher score from PCA first component is clustered in 6th group

*which denotes that these people have strong liking towards all types of music followed by 3rd group of people and the lowest scores from PCA is clustered in 4th group which says that these people have low or no rating towards most of the music types.*