In-store Music Experience Quantification

Nelson Dsouza

Jan 9, 2017

Preparation

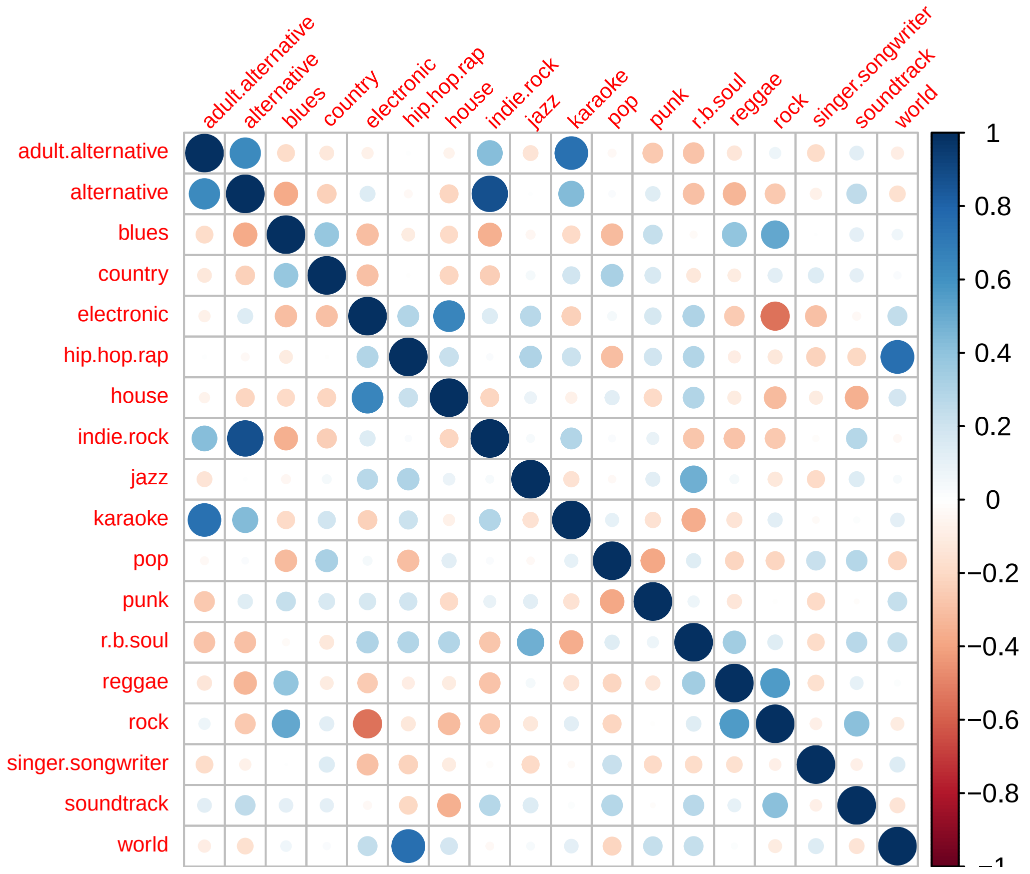
Reading libraries and data

```
library(ggplot2)
library(corrplot)
library(GPArotation)

ud <- read.csv("play_udist_final.csv")
```

Checking correlation

```
# Calculating the correlation matrix
ud.cor <- cor(ud[,-(1:2)])
corrplot(ud.cor, tl.cex = .7, tl.srt = 45)
```

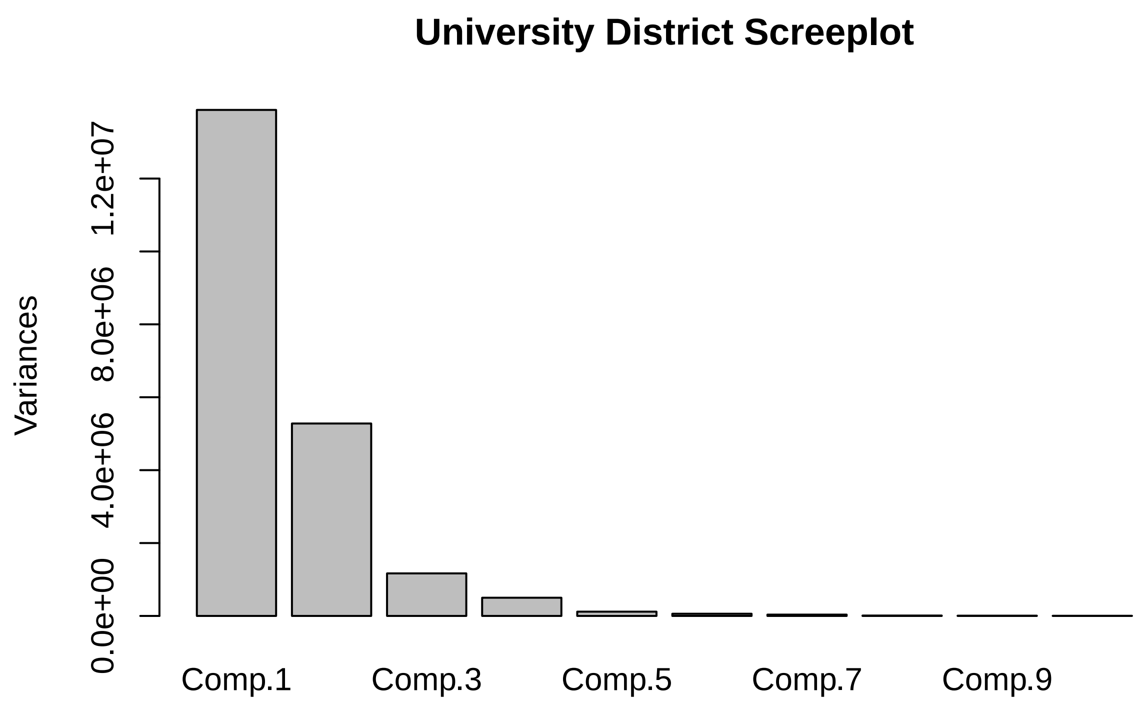


We see that there is good amount of correlation in the data.

PCA Analysis

Since we have 18 genre, we would require 18 dimensions to visualize the information. Let us see if we can reduce the dimensions using PCA.

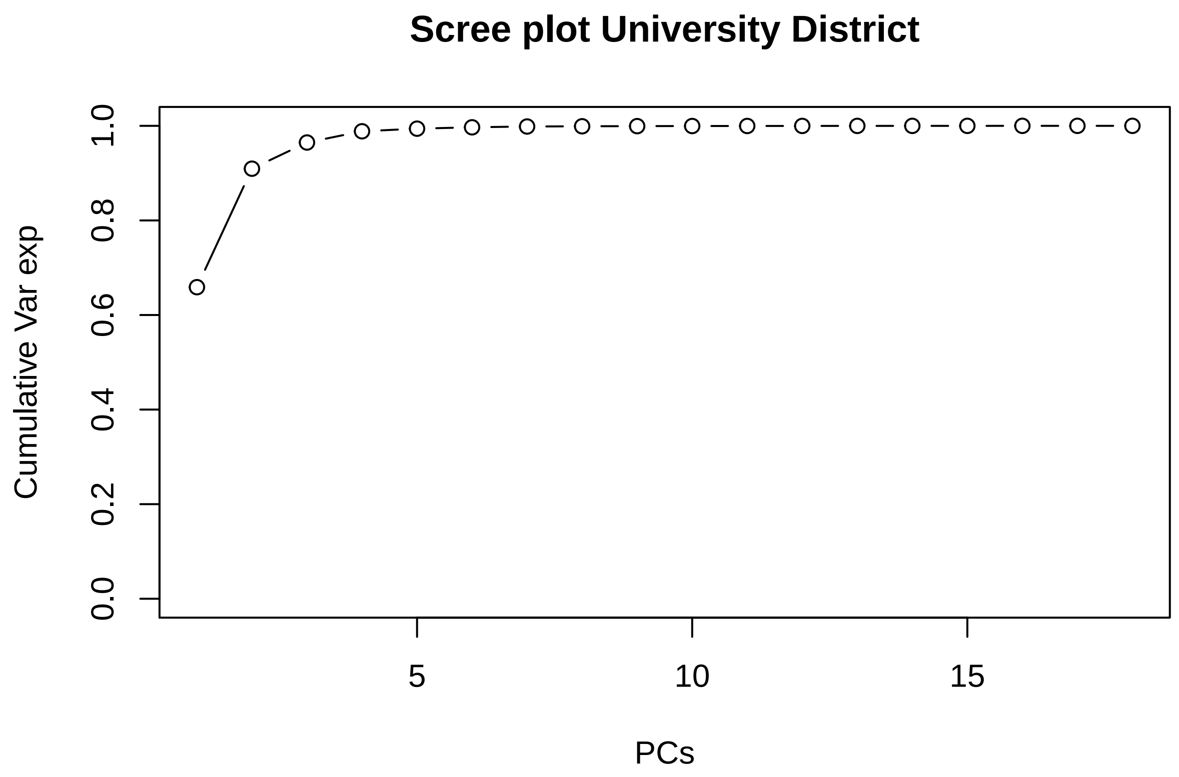
```r
ud.pc <- princomp(ud[,-(1:2)])
#print(ud.pc$loadings)
#summary(ud.pc)
plot(ud.pc, main = "University District Screeplot")
```



We can see that the first two principal components captures roughly 90% of the variability for University District data. Thus the first two eigenvalues are much larger than any of the remaining eigenvalues for University District.

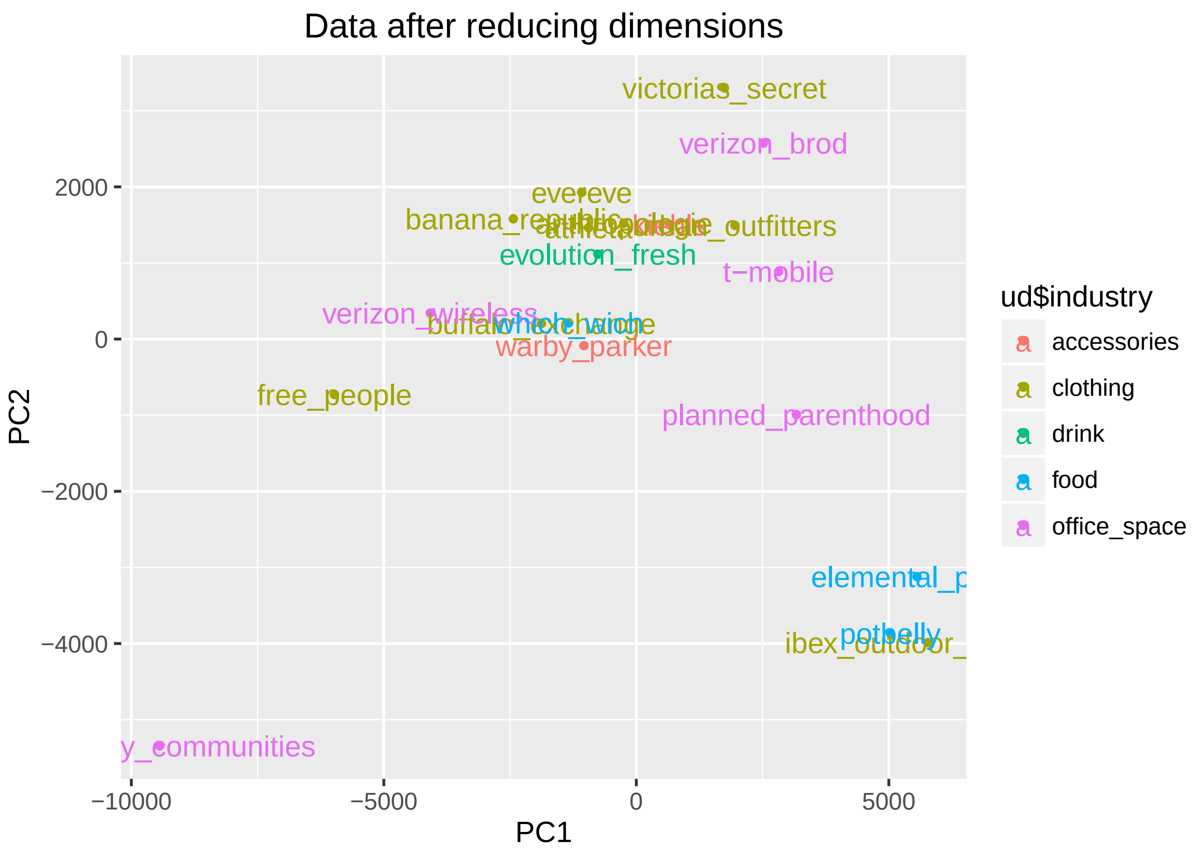
Now plotting cumulative variance explained to select how many principal components would be appropriate for adequately preserving “most of the information” in the data.

```r
plot(cumsum(ud.pc$sdev^2 / sum(ud.pc$sdev^2)), xlab = "PCs",
     ylab = "Cumulative Var exp", type = "b", ylim = c(0,1),
     main = "Scree plot University District")
```



Plotting first 2 components and setting color by industry

```
ggplot(ud,aes(ud.pc$scores[,1],ud.pc$scores[,2],
              color=ud$industry, label=ud[,1])) + xlab("PC1") + ylab("PC2") +
              ggtitle("Data after reducing dimensions") +
              geom_point(size=1) +
              geom_text()
```



From PCA we see that the first 2 components are enough to explain the bulk of the data which means we can compress the multiple genre dimensions into 2 dimensions.

Factor Analysis

```r
# Obtaining 2 latent factors since PCA suggested 2 dimensions
ud.fa <- factanal(ud[-(1:2)], factors=2, rotation = "none", scores = "regression")
ud.fa.vm <- factanal(ud[-(1:2)], factors=2, rotation="varimax", scores="regression")
ud.fa.om <- factanal(ud[-(1:2)], factors=2, rotation="oblimin", scores="regression")

ud.fa.vm
```

```
##
## Call:
## factanal(x = ud[-(1:2)], factors = 2, scores = "regression",     rotation = "varimax")
##
## Uniquenesses:
## adult.alternative       alternative             blues           country
##             0.564             0.005             0.794             0.875
##        electronic       hip.hop.rap             house        indie.rock
##             0.005             0.905             0.470             0.224
##              jazz           karaoke               pop              punk
##             0.925             0.723             0.997             0.958
##           r.b.soul            reggae              rock singer.songwriter
##             0.789             0.841             0.660             0.913
##        soundtrack             world
##             0.928             0.904
##
## Loadings:
##                    Factor1 Factor2
## adult.alternative   0.656
## alternative         0.928   0.367
## blues              -0.251  -0.378
## country            -0.117  -0.334
## electronic         -0.227   0.971
## hip.hop.rap        -0.145   0.271
## house              -0.456   0.568
## indie.rock          0.811   0.343
## jazz                        0.256
## karaoke             0.513  -0.119
## pop
## punk                        0.196
## r.b.soul           -0.403   0.220
## reggae             -0.240  -0.319
## rock                       -0.580
## singer.songwriter          -0.293
## soundtrack          0.268
## world              -0.246   0.190
##
##                 Factor1 Factor2
## SS loadings       2.939   2.581
## Proportion Var    0.163   0.143
## Cumulative Var    0.163   0.307
##
## Test of the hypothesis that 2 factors are sufficient.
## The chi square statistic is 134.1 on 118 degrees of freedom.
## The p-value is 0.148
```

We see that for University District data, the model gives a good fit.

Checking reconstructed correlaton matrix for goodness of fit

```
# Obtaining reconstructed matrix and subtracting the value of original matrix
ud.reconst.cor <- loadings(ud.fa) %*% t(loadings(ud.fa)) + diag(ud.fa$uniquenesses)
ud.diff <- round(abs(ud.reconst.cor - ud.cor), 3)
ud.per.diff = data.frame(rowSums(ud.diff)*100/nrow(ud.diff))
ud.per.diff
```

```
##                    rowSums.ud.diff....100.nrow.ud.diff.
## adult.alternative                            10.27777778
## alternative                                   0.10000000
## blues                                        11.44444444
## country                                      11.21111111
## electronic                                    0.08333333
## hip.hop.rap                                  13.78888889
## house                                         8.58888889
## indie.rock                                    3.88333333
## jazz                                          8.98333333
## karaoke                                      13.65555556
## pop                                          16.10555556
## punk                                         14.35000000
## r.b.soul                                     12.40555556
## reggae                                       11.90000000
## rock                                         12.71111111
## singer.songwriter                            11.56111111
## soundtrack                                   14.61666667
## world                                        11.96666667
```

We see the reconstructed correlation matrix is close enough to original matrix! Thus the model is useful.

Now let us see the factor loadings which will help in interpretation

```
# Storing factor loadings in dataframe
ud.factors <- data.frame(cbind(round(ud.fa.vm$loadings[1:18],2),
                               round(ud.fa.vm$loadings[19:36],2)))
colnames(ud.factors) <- c("Factor1","Factor2")
row.names(ud.factors) <- names(ud[3:20])

ud.factors[order(-ud.factors$Factor1),]
```

```
##                     Factor1 Factor2
## alternative            0.93    0.37
## indie.rock             0.81    0.34
## adult.alternative      0.66    0.07
## karaoke                0.51   -0.12
## soundtrack             0.27    0.02
## punk                   0.06    0.20
## singer.songwriter      0.03   -0.29
## pop                    0.01    0.05
## rock                  -0.06   -0.58
## jazz                  -0.10    0.26
## country               -0.12   -0.33
## hip.hop.rap           -0.14    0.27
## electronic            -0.23    0.97
## reggae                -0.24   -0.32
## blues                 -0.25   -0.38
## world                 -0.25    0.19
## r.b.soul              -0.40    0.22
## house                 -0.46    0.57
```

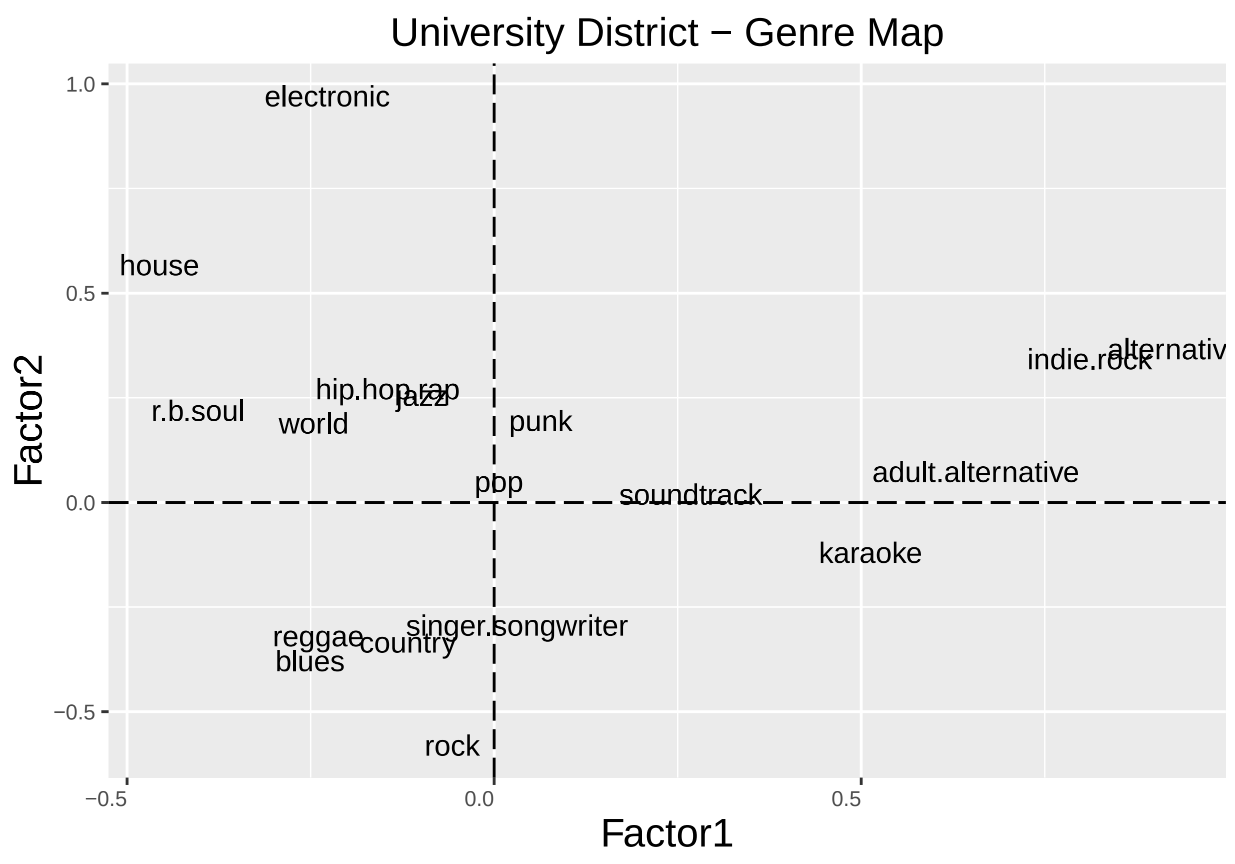
```
ud.factors[order(-ud.factors$Factor2),]
```

```
##                     Factor1 Factor2
## electronic            -0.23    0.97
## house                 -0.46    0.57
## alternative            0.93    0.37
## indie.rock             0.81    0.34
## hip.hop.rap           -0.14    0.27
## jazz                  -0.10    0.26
## r.b.soul              -0.40    0.22
## punk                   0.06    0.20
## world                -0.25    0.19
## adult.alternative     0.66    0.07
## pop                   0.01    0.05
## soundtrack            0.27    0.02
## karaoke               0.51   -0.12
## singer.songwriter     0.03   -0.29
## reggae               -0.24   -0.32
## country              -0.12   -0.33
## blues                -0.25   -0.38
## rock                 -0.06   -0.58
```

The above table give the loadings which can be used to interpret the visualization.

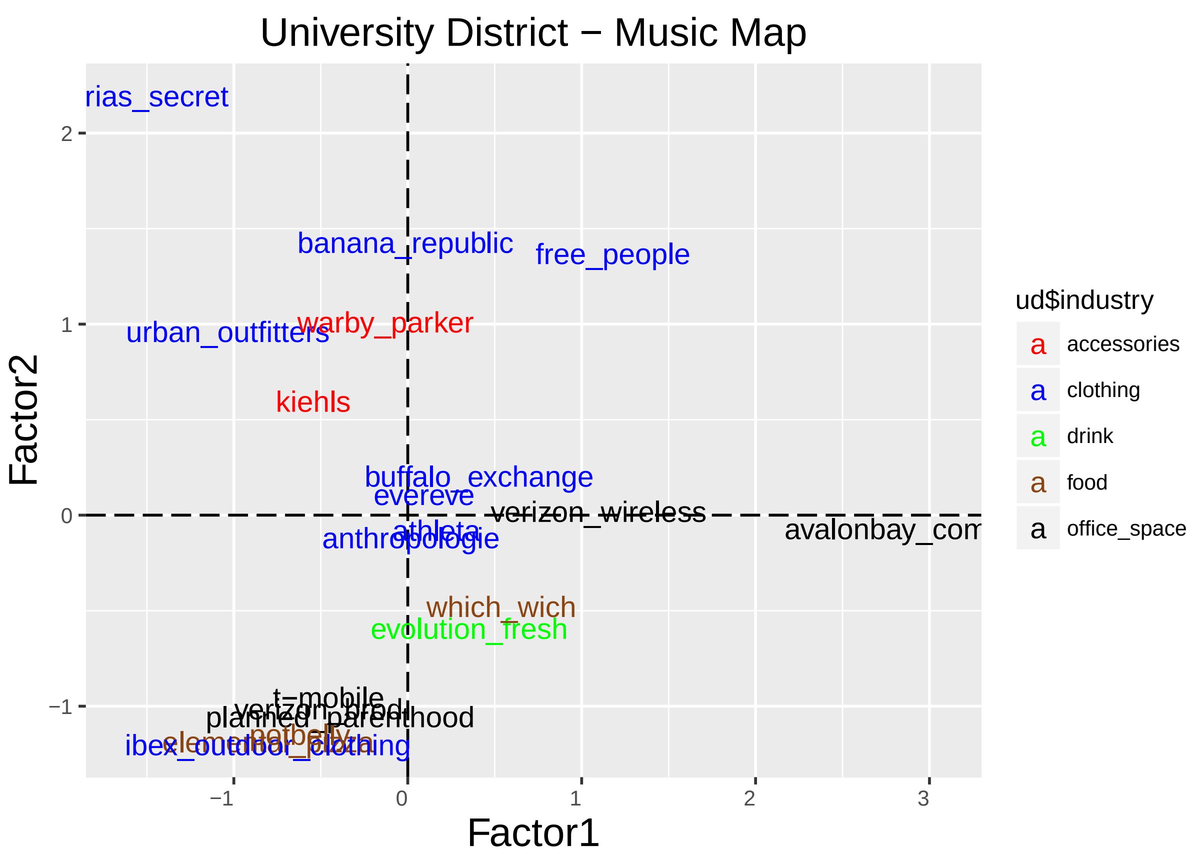
Plotting the genres on 2 factors

```r
ud.fa.vm.ld <- data.frame(ud.fa.vm$loadings[,0:2])
ggplot(ud.fa.vm.ld,aes(ud.fa.vm.ld[,1],ud.fa.vm.ld[,2],
          label=row.names(ud.fa.vm.ld))) +
          geom_text(size=4) +
          geom_vline(xintercept = 0, linetype = "longdash") +
          geom_hline(yintercept = 0, linetype = "longdash") +
          xlab("Factor1") + ylab("Factor2") +
          ggtitle("University District - Genre Map") +
          theme(text = element_text(size=10), axis.text.x = element_text(hjust=1)) +
          #ggtitle("Latent Factor Plot - U District") +
          theme(plot.title = element_text(lineheight=3, size=15)) +
          theme(axis.title.x = element_text(size = rel(1.5), angle = 00))+
          theme(axis.title.y = element_text(size = rel(1.5), angle = 90))
```



Plotting the UDstrict stores for the 2 latent factors

```
ud.fa.vm.sc <- data.frame(ud.fa.vm$scores)
ggplot(ud.fa.vm.sc,aes(ud.fa.vm.sc[,1],ud.fa.vm.sc[,2], color=ud$industry,
          label=ud[,1])) +
          geom_vline(xintercept = 0, linetype = "longdash") +
          geom_hline(yintercept = 0, linetype = "longdash") +
          geom_text(size=4) +
          scale_colour_manual(values = c("red","blue", "green", "chocolate4", "black")) +
          xlab("Score1") + ylab("Score2") +
          xlab("Factor1") + ylab("Factor2") +
          ggtitle("University District - Music Map") +
          theme(text = element_text(size=10), axis.text.x = element_text(hjust=1)) +
          #ggtitle("Score Plot - U District") +
          theme(plot.title = element_text(lineheight=3, size=15)) +
          theme(axis.title.x = element_text(size = rel(1.5), angle = 00))+
          theme(axis.title.y = element_text(size = rel(1.5), angle = 90))
```



This gives us a 'MUSIC MAP' which helps visualize the quantified in-store music experience.

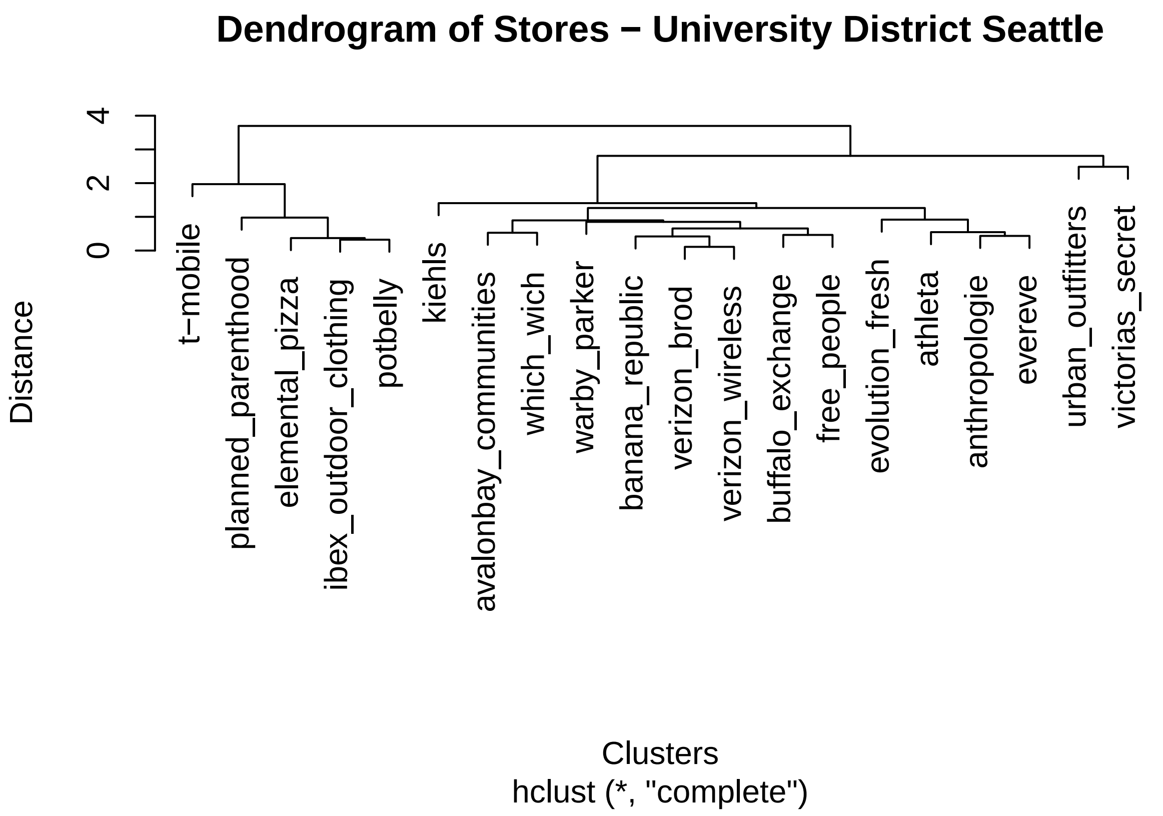
Cluster Analysis

```
# Normalizing data
ud.normalized.rows <- t(apply(ud[,-(1:2)], 1, function(x)(x-min(x))/(max(x)-min(x))))
ud.norm <- cbind(ud[,1:2], ud.normalized.rows)

# Nested for loops for calculating distance matrix
ud.distance <- matrix(0, nrow = 20, ncol = 20)
colnames(ud.distance) <- ud[,1]
row.names(ud.distance) <- ud[,1]

for(i in 1:20){
  for(j in 1:i){
    ud.distance[i, j] <- sum(sqrt((ud.norm[i,3:20] - ud.norm[j,3:20])^2))
  }
}
ud.dist <- as.dist(ud.distance)

# Applying clustering
ud.clusters <- hclust(ud.dist, method = "complete")
plot(ud.clusters, main="Dendrogram of Stores - University District Seattle",
     xlab="Clusters", ylab="Distance")
```



Clusters are an alternative method of visualizing the ‘musical distance’ between the different stores.