Spotify Dataset Clustering - Profiling Analysis

2023-01-05

# Spotify Dataset Clustering - Profiling Analysis

This report is a continuation of the clustering.Rmd process. The SpotifyClustersData.csv was sourced from that dataset.

# Replace with your own working directory if needed  
WD <- "C:/Users/Joseph/Documents/Codes/2022/mvtec-2022/finalproject/04-clustering"  
setwd(WD)  
  
# dd <- read.csv("data/SpotifyClustersData.csv",header=T, sep=",", dec='.');  
  
c2 <- read.csv("data/c2.csv", header=T) %>% pull()  
  
dd <- read.csv("data/cleaneddata-withtitles.csv") %>%  
 select("key", "tempo\_cat",  
 "loudness", "energy", "acousticness", "instrumentalness", "valence") %>%  
 bind\_cols(tibble(c2))  
head(dd, 10)

## key tempo\_cat loudness energy acousticness instrumentalness valence c2  
## 1 A Allegro -5.274 0.854 0.049800 2.53e-04 0.699 1  
## 2 C Allegro -3.464 0.855 0.003410 4.28e-02 0.609 1  
## 3 A#/Bb Allegro -5.308 0.617 0.171000 0.00e+00 0.759 1  
## 4 D Andante -13.106 0.235 0.917000 2.64e-01 0.382 2  
## 5 A Moderato -10.344 0.714 0.000726 9.08e-01 0.269 3  
## 6 D Moderato -6.901 0.771 0.001190 2.61e-02 0.743 1  
## 7 G Andante -2.976 0.992 0.004000 2.73e-05 0.200 1  
## 8 F Allegro -13.509 0.573 0.906000 8.60e-01 0.650 4  
## 9 A#/Bb Adagio -9.089 0.445 0.990000 5.38e-01 0.192 4  
## 10 C Presto -7.280 0.397 0.556000 0.00e+00 0.307 1

As you may notice, the new cluster variable was generated from dissimilarity Matrix / Gower’s Distance from previous script.

names(dd)

## [1] "key" "tempo\_cat" "loudness" "energy"   
## [5] "acousticness" "instrumentalness" "valence" "c2"

attach(dd)

## The following object is masked \_by\_ .GlobalEnv:  
##   
## c2

Calculate the test value of variable Xnum for all modalities of factor P

ValorTestXnum <- function(Xnum,P){  
 # Freq dis of factors  
 nk <- as.vector(table(P));   
 n <- sum(nk);  
   
 # Averages vs groups  
 xk <- tapply(Xnum,P,mean);  
   
 # Test values  
 txk <- (xk-mean(Xnum))/(sd(Xnum)\*sqrt((n-nk)/(n\*nk)));  
   
 # P-values  
 pxk <- pt(txk,n-1,lower.tail=F);  
 for(c in 1:length(levels(as.factor(P)))){if (pxk[c]>0.5){pxk[c]<-1-pxk[c]}}  
 return (pxk)  
}  
  
ValorTestXquali <- function(P,Xquali){  
 taula <- table(P,Xquali);  
 n <- sum(taula);   
 pk <- apply(taula,1,sum)/n;  
 pj <- apply(taula,2,sum)/n;  
 pf <- taula/(n\*pk);  
 pjm <- matrix(data=pj,nrow=dim(pf)[1],ncol=dim(pf)[2], byrow=TRUE);   
 dpf <- pf - pjm;   
 dvt <- sqrt(((1-pk)/(n\*pk))%\*%t(pj\*(1-pj)));  
   
 # And there are divisions equal to 0 woman NA and it doesn't work  
 zkj <- dpf  
 zkj[dpf!=0]<-dpf[dpf!=0]/dvt[dpf!=0];   
 pzkj <- pnorm(zkj,lower.tail=F);  
 for(c in 1:length(levels(as.factor(P)))){for (s in 1:length(levels(Xquali))){if (pzkj[c,s]> 0.5){pzkj[c,s]<-1- pzkj[c,s]}}}  
 return (list(rowpf=pf,vtest=zkj,pval=pzkj))  
}

# dades contain the dataset  
dades <- dd  
  
K<-dim(dades)[2]  
par(ask=TRUE)

Using the added column ‘cluster’ which we’ll continue use here as c2

# P must contain the class variable  
# c2 <- dd$cluster  
P <- c2  
  
nc <- length(levels(factor(P)))  
nc

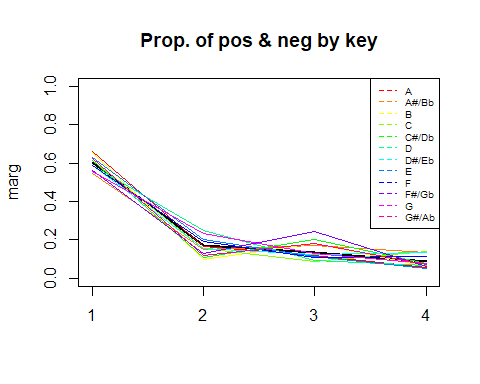
## [1] 4

pvalk <- matrix(data=0,nrow=nc,ncol=K, dimnames=list(levels(P),names(dades)))  
nameP <- "Class"  
n <- dim(dades)[1]

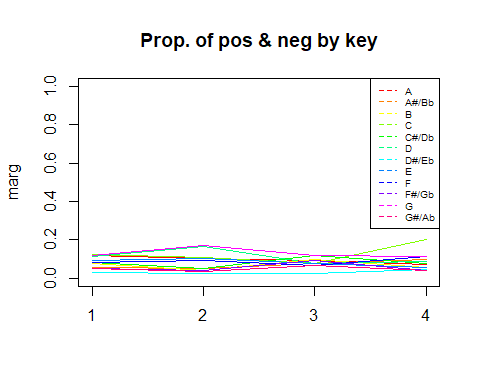
### Loop through the variable dataset to profile and visualize each

for(k in 1:K){  
 if (is.numeric(dades[,k])){   
 print(paste("Analysis by classes of the Variable:", names(dades)[k]))  
   
 boxplot(dades[,k]~P, main=paste("Boxplot of", names(dades)[k], "vs", nameP ), horizontal=TRUE)  
   
 barplot(tapply(dades[[k]], P, mean),main=paste("Means of", names(dades)[k], "by", nameP ))  
 abline(h=mean(dades[[k]]))  
 legend(0,mean(dades[[k]]),"global mean",bty="n")  
 print("Statistics per groups:")  
 for(s in levels(as.factor(P))) {print(summary(dades[P==s,k]))}  
 o<-oneway.test(dades[,k]~P)  
 print(paste("p-value ANOVA:", o$p.value))  
 kw<-kruskal.test(dades[,k]~P)  
 print(paste("p-value Kruskal-Wallis:", kw$p.value))  
 pvalk[,k]<-ValorTestXnum(dades[,k], P)  
 print("p-values ValorsTest: ")  
 print(pvalk[,k])   
 }else{  
 if(class(dd[,k])=="Date"){  
 print(summary(dd[,k]))  
 print(sd(dd[,k]))  
   
 # Decide breaks: weeks, months, quarters...  
 hist(dd[,k],breaks="weeks")  
   
 }else{  
 # Qualitatives  
 print(paste("Variable", names(dades)[k]))  
 table<-table(P,dades[,k])  
  
 rowperc<-prop.table(table,1)  
   
 colperc<-prop.table(table,2)  
   
 # Observe why the variable is true or false. It identifies the type of logic  
 # This one has no levels, therefore, coercion was prevented  
 dades[,k]<-as.factor(dades[,k])  
   
 marg <- table(as.factor(P))/n  
 print(append("Categories=",levels(as.factor(dades[,k]))))  
   
 # From next plots, select one of them according to your practical case  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }  
   
 # With legend  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }  
 legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)  
   
 # Conditioned to classes  
 print(append("Categories=",levels(dades[,k])))  
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }  
   
 # With legend  
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }  
 legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)  
   
 # With abcisses axis variable  
 marg <-table(dades[,k])/n  
 print(append("Categories=",levels(dades[,k])))  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 paleta<-rainbow(length(levels(as.factor(P))))  
 for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c]) }  
   
 # With legend  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c])}  
 legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)  
   
 # Conditioned to column  
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 paleta<-rainbow(length(levels(as.factor(P))))  
 for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c]) }  
   
 # With legend  
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c])}  
 legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)  
   
 table<-table(dades[,k],P)  
 print("Cross Table:")  
 print(table)  
 print("Conditional distributions column :")  
 print(colperc)  
   
 # Stacked bar charts   
 paleta<-rainbow(length(levels(dades[,k])))  
 barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )  
   
 barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )  
 legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)  
   
 # Attached bar charts  
 barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta )  
   
 barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta)  
 legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)  
   
 print("Square Chi test: ")  
 print(chisq.test(dades[,k], as.factor(P)))  
   
 print("Values test:")  
 print( ValorTestXquali(P,dades[,k]))  
 # Calculate the pvalues of quali  
 }  
 }  
}

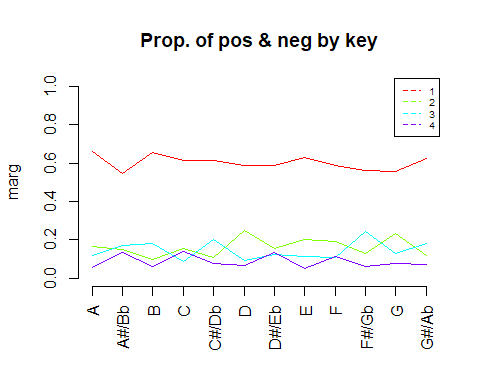
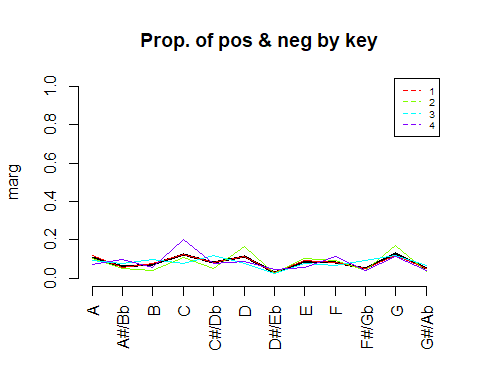
## [1] "Variable key"  
## [1] "Categories=" "A" "A#/Bb" "B" "C"   
## [6] "C#/Db" "D" "D#/Eb" "E" "F"   
## [11] "F#/Gb" "G" "G#/Ab"



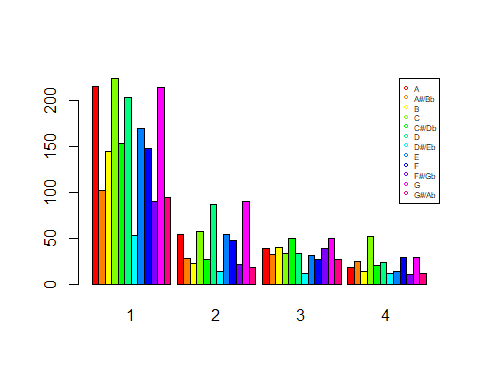
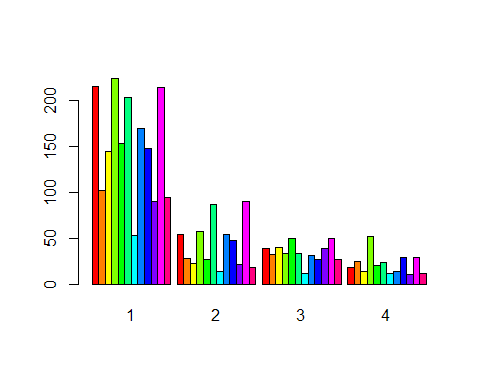
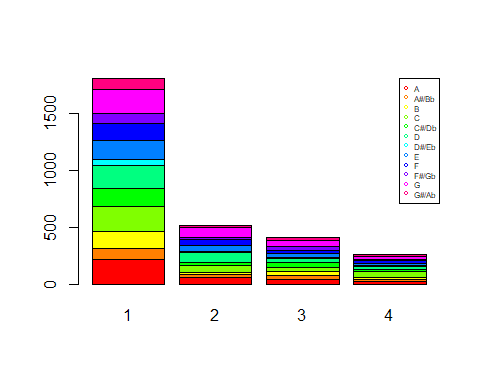
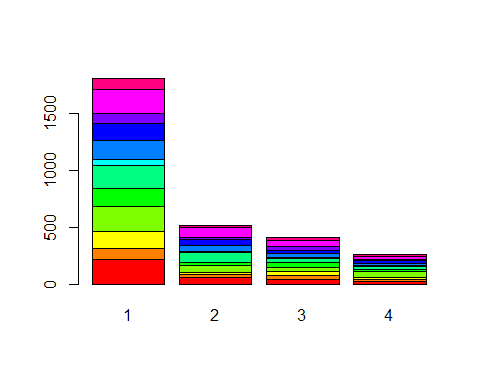
## [1] "Categories=" "A" "A#/Bb" "B" "C"   
## [6] "C#/Db" "D" "D#/Eb" "E" "F"   
## [11] "F#/Gb" "G" "G#/Ab"



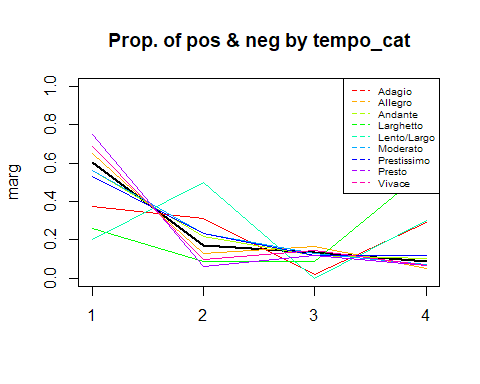
## [1] "Categories=" "A" "A#/Bb" "B" "C"   
## [6] "C#/Db" "D" "D#/Eb" "E" "F"   
## [11] "F#/Gb" "G" "G#/Ab"



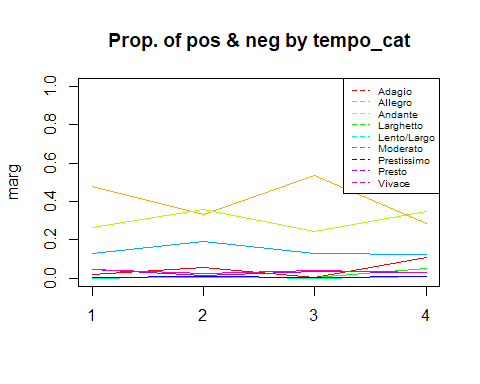
## [1] "Cross Table:"  
## P  
## 1 2 3 4  
## A 215 54 39 18  
## A#/Bb 102 28 32 25  
## B 145 22 40 14  
## C 224 57 33 52  
## C#/Db 153 27 50 20  
## D 203 87 33 23  
## D#/Eb 53 14 11 12  
## E 170 54 31 14  
## F 148 48 27 29  
## F#/Gb 90 21 39 10  
## G 214 90 50 29  
## G#/Ab 94 18 27 11  
## [1] "Conditional distributions column :"  
##   
## P A A#/Bb B C C#/Db D  
## 1 0.65950920 0.54545455 0.65610860 0.61202186 0.61200000 0.58670520  
## 2 0.16564417 0.14973262 0.09954751 0.15573770 0.10800000 0.25144509  
## 3 0.11963190 0.17112299 0.18099548 0.09016393 0.20000000 0.09537572  
## 4 0.05521472 0.13368984 0.06334842 0.14207650 0.08000000 0.06647399  
##   
## P D#/Eb E F F#/Gb G G#/Ab  
## 1 0.58888889 0.63197026 0.58730159 0.56250000 0.55874674 0.62666667  
## 2 0.15555556 0.20074349 0.19047619 0.13125000 0.23498695 0.12000000  
## 3 0.12222222 0.11524164 0.10714286 0.24375000 0.13054830 0.18000000  
## 4 0.13333333 0.05204461 0.11507937 0.06250000 0.07571802 0.07333333



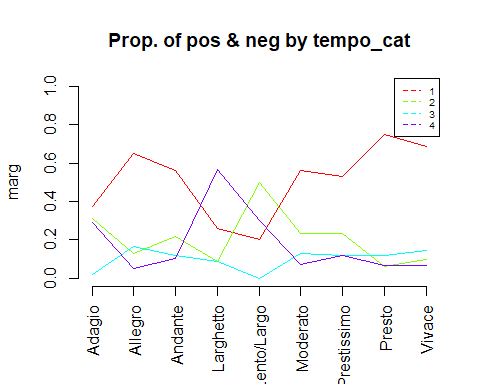
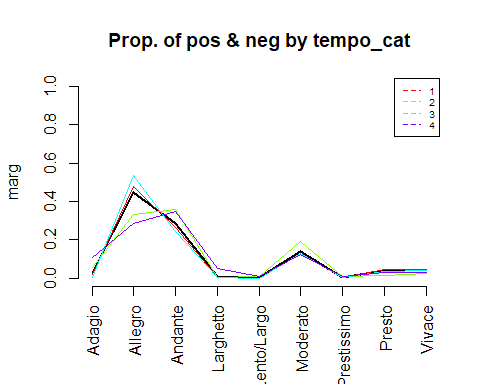
## [1] "Square Chi test: "  
##   
## Pearson's Chi-squared test  
##   
## data: dades[, k] and as.factor(P)  
## X-squared = 123.42, df = 33, p-value = 2.278e-12  
##   
## [1] "Values test:"  
## $rowpf  
## Xquali  
## P A A#/Bb B C C#/Db D  
## 1 0.11871894 0.05632247 0.08006626 0.12368857 0.08448371 0.11209277  
## 2 0.10384615 0.05384615 0.04230769 0.10961538 0.05192308 0.16730769  
## 3 0.09466019 0.07766990 0.09708738 0.08009709 0.12135922 0.08009709  
## 4 0.07003891 0.09727626 0.05447471 0.20233463 0.07782101 0.08949416  
## Xquali  
## P D#/Eb E F F#/Gb G G#/Ab  
## 1 0.02926560 0.09387079 0.08172281 0.04969630 0.11816676 0.05190502  
## 2 0.02692308 0.10384615 0.09230769 0.04038462 0.17307692 0.03461538  
## 3 0.02669903 0.07524272 0.06553398 0.09466019 0.12135922 0.06553398  
## 4 0.04669261 0.05447471 0.11284047 0.03891051 0.11284047 0.04280156  
##   
## $vtest  
## Xquali  
## P A A#/Bb B C C#/Db D  
## 1 2.1833580 -1.6806648 1.6560074 0.3487553 0.2813547 -0.6857766  
## 2 -0.3884735 -0.8804718 -3.0107809 -0.9490559 -2.8503149 4.0809225  
## 3 -0.9835300 1.3863437 1.9593320 -2.7979770 3.0067089 -2.4107350  
## 4 -2.0808727 2.4231997 -1.2317269 4.1151902 -0.3343750 -1.3562058  
## Xquali  
## P D#/Eb E F F#/Gb G G#/Ab  
## 1 -0.2910152 0.9946919 -0.5549345 -1.0941535 -1.9242803 0.5908575  
## 2 -0.4523833 1.2447475 0.7511549 -1.4453244 3.4127967 -1.7704200  
## 3 -0.4228850 -1.1033048 -1.4548325 4.0194035 -0.4130479 1.5576255  
## 4 1.6405594 -2.0651043 1.7431277 -1.0761351 -0.7448420 -0.5537402  
##   
## $pval  
## Xquali  
## P A A#/Bb B C C#/Db  
## 1 1.450473e-02 4.641403e-02 4.886016e-02 3.636365e-01 3.892192e-01  
## 2 3.488328e-01 1.893019e-01 1.302884e-03 1.712961e-01 2.183798e-03  
## 3 1.626733e-01 8.282099e-02 2.503696e-02 2.571189e-03 1.320462e-03  
## 4 1.872278e-02 7.692234e-03 1.090256e-01 1.934301e-05 3.690483e-01  
## Xquali  
## P D D#/Eb E F F#/Gb  
## 1 2.464270e-01 3.855199e-01 1.599431e-01 2.894697e-01 1.369438e-01  
## 2 2.242866e-05 3.254964e-01 1.066123e-01 2.262797e-01 7.418340e-02  
## 3 7.960205e-03 3.361896e-01 1.349474e-01 7.285781e-02 2.917283e-05  
## 4 8.751685e-02 5.044445e-02 1.945657e-02 4.065566e-02 1.409334e-01  
## Xquali  
## P G G#/Ab  
## 1 2.715973e-02 2.773079e-01  
## 2 3.214994e-04 3.832860e-02  
## 3 3.397857e-01 5.966102e-02  
## 4 2.281836e-01 2.898783e-01  
##   
## [1] "Variable tempo\_cat"  
## [1] "Categories=" "Adagio" "Allegro" "Andante" "Larghetto"   
## [6] "Lento/Largo" "Moderato" "Prestissimo" "Presto" "Vivace"



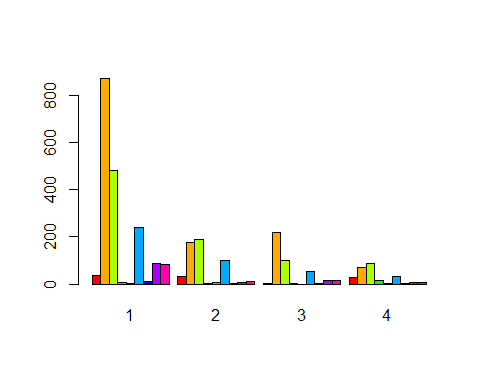
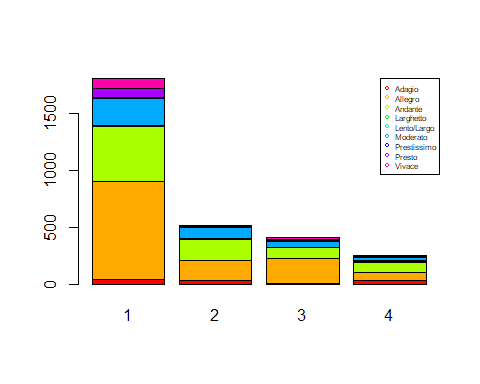
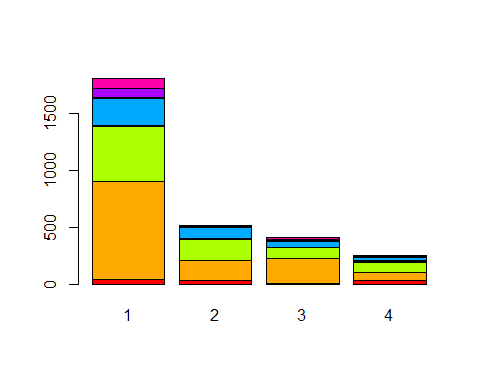
## [1] "Categories=" "Adagio" "Allegro" "Andante" "Larghetto"   
## [6] "Lento/Largo" "Moderato" "Prestissimo" "Presto" "Vivace"



## [1] "Categories=" "Adagio" "Allegro" "Andante" "Larghetto"   
## [6] "Lento/Largo" "Moderato" "Prestissimo" "Presto" "Vivace"

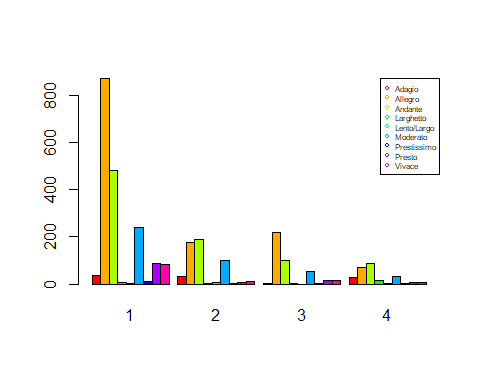


## [1] "Cross Table:"  
## P  
## 1 2 3 4  
## Adagio 36 30 2 28  
## Allegro 869 174 220 72  
## Andante 482 187 101 88  
## Larghetto 6 2 2 13  
## Lento/Largo 2 5 0 3  
## Moderato 238 99 54 31  
## Prestissimo 9 4 2 2  
## Presto 87 7 14 8  
## Vivace 82 12 17 8  
## [1] "Conditional distributions column :"  
##   
## P Adagio Allegro Andante Larghetto Lento/Largo Moderato  
## 1 0.37500000 0.65093633 0.56177156 0.26086957 0.20000000 0.56398104  
## 2 0.31250000 0.13033708 0.21794872 0.08695652 0.50000000 0.23459716  
## 3 0.02083333 0.16479401 0.11771562 0.08695652 0.00000000 0.12796209  
## 4 0.29166667 0.05393258 0.10256410 0.56521739 0.30000000 0.07345972  
##   
## P Prestissimo Presto Vivace  
## 1 0.52941176 0.75000000 0.68907563  
## 2 0.23529412 0.06034483 0.10084034  
## 3 0.11764706 0.12068966 0.14285714  
## 4 0.11764706 0.06896552 0.06722689

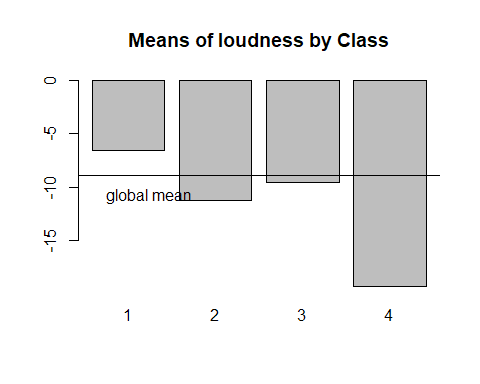
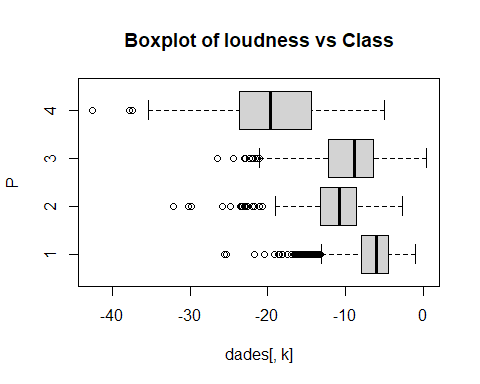


## [1] "Square Chi test: "

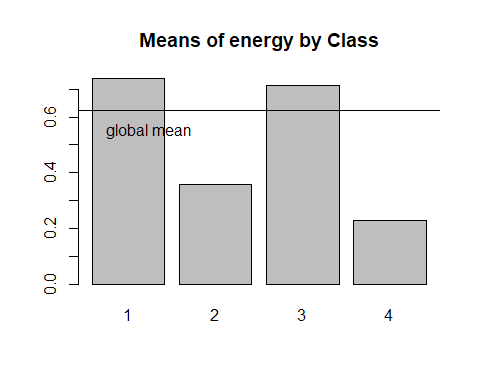
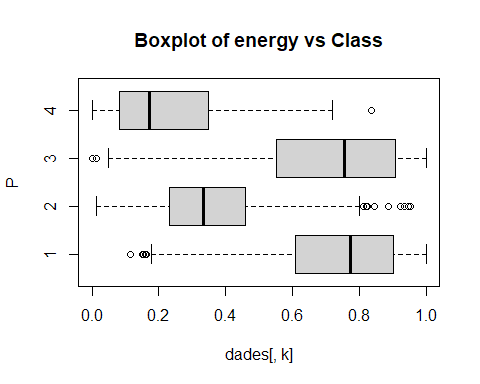
## Warning in chisq.test(dades[, k], as.factor(P)): Chi-squared approximation may  
## be incorrect



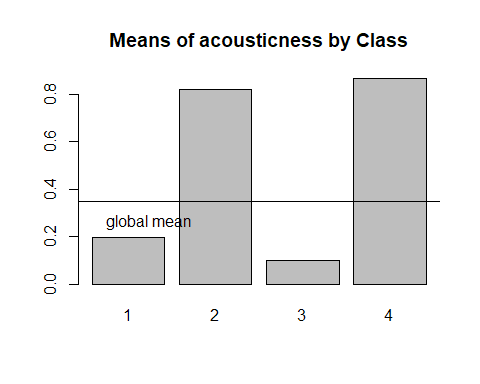
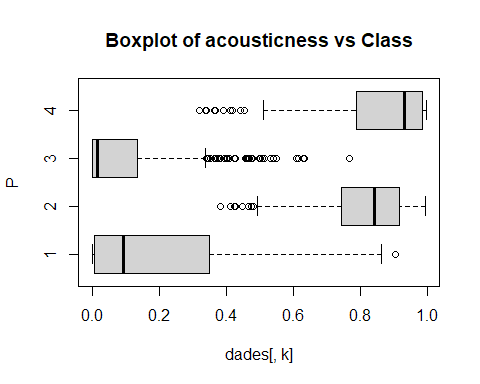
##   
## Pearson's Chi-squared test  
##   
## data: dades[, k] and as.factor(P)  
## X-squared = 251.71, df = 24, p-value < 2.2e-16  
##   
## [1] "Values test:"  
## $rowpf  
## Xquali  
## P Adagio Allegro Andante Larghetto Lento/Largo Moderato  
## 1 0.019878520 0.479845389 0.266151298 0.003313087 0.001104362 0.131419105  
## 2 0.057692308 0.334615385 0.359615385 0.003846154 0.009615385 0.190384615  
## 3 0.004854369 0.533980583 0.245145631 0.004854369 0.000000000 0.131067961  
## 4 0.110671937 0.284584980 0.347826087 0.051383399 0.011857708 0.122529644  
## Xquali  
## P Prestissimo Presto Vivace  
## 1 0.004969630 0.048039757 0.045278851  
## 2 0.007692308 0.013461538 0.023076923  
## 3 0.004854369 0.033980583 0.041262136  
## 4 0.007905138 0.031620553 0.031620553  
##   
## $vtest  
## Xquali  
## P Adagio Allegro Andante Larghetto Lento/Largo Moderato  
## 1 -4.6737107 4.6629839 -3.0281184 -3.3831231 -2.6202248 -1.8353169  
## 2 3.6532953 -5.6008431 4.0635126 -1.1009342 2.7301585 3.5714794  
## 3 -3.3741422 3.8866492 -1.9936381 -0.7067939 -1.2648189 -0.6148688  
## 4 7.4218023 -5.3849933 2.2594034 8.3242419 2.4555619 -0.8756680  
## Xquali  
## P Prestissimo Presto Vivace  
## 1 -0.6347418 3.2694277 1.9261213  
## 2 0.6739330 -3.2839107 -2.1375659  
## 3 -0.2385596 -0.5367328 0.1726234  
## 4 0.4937210 -0.6115821 -0.6893693  
##   
## $pval  
## Xquali  
## P Adagio Allegro Andante Larghetto Lento/Largo  
## 1 1.479030e-06 1.558286e-06 1.230408e-03 3.583325e-04 4.393591e-03  
## 2 1.294481e-04 1.066559e-08 2.416986e-05 1.354627e-01 3.165194e-03  
## 3 3.702306e-04 5.081873e-05 2.309581e-02 2.398473e-01 1.029681e-01  
## 4 5.776857e-14 3.622368e-08 1.192915e-02 4.243505e-17 7.033227e-03  
## Xquali  
## P Moderato Prestissimo Presto Vivace  
## 1 3.322937e-02 2.627984e-01 5.388263e-04 2.704462e-02  
## 2 1.774852e-04 2.501769e-01 5.118868e-04 1.627599e-02  
## 3 2.693207e-01 4.057236e-01 2.957261e-01 4.314737e-01  
## 4 1.906053e-01 3.107516e-01 2.704071e-01 2.452954e-01  
##   
## [1] "Analysis by classes of the Variable: loudness"



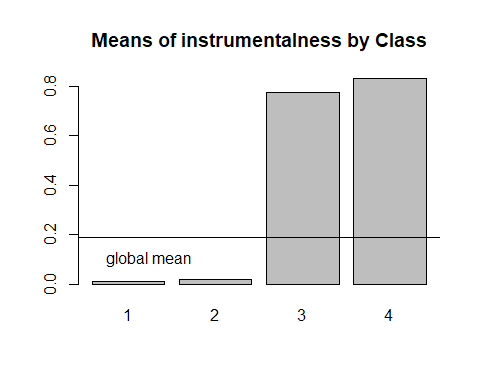
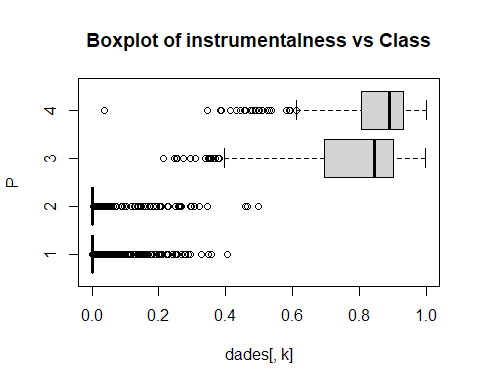
## [1] "Statistics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -25.641 -7.987 -6.035 -6.583 -4.519 -0.958   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -32.108 -13.261 -10.819 -11.236 -8.603 -2.689   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -26.527 -12.243 -8.899 -9.548 -6.371 0.377   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -42.631 -23.609 -19.674 -19.321 -14.454 -5.044   
## [1] "p-value ANOVA: 2.17489695592098e-173"  
## [1] "p-value Kruskal-Wallis: 2.45213703529884e-239"  
## [1] "p-values ValorsTest: "  
## [1] 3.451881e-166 0.000000e+00 3.369412e-03 0.000000e+00  
## [1] "Analysis by classes of the Variable: energy"



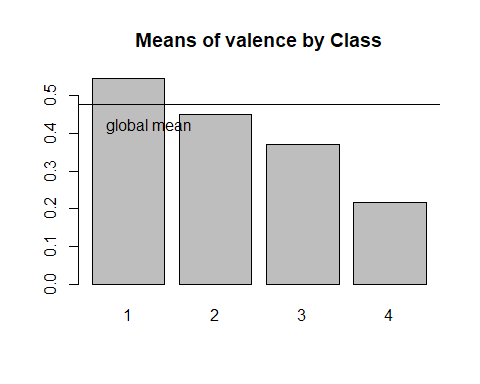
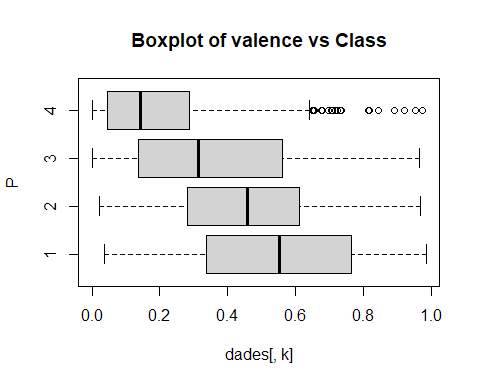
## [1] "Statistics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.1140 0.6070 0.7710 0.7381 0.9000 0.9990   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0125 0.2315 0.3325 0.3562 0.4590 0.9530   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000242 0.551500 0.755500 0.714677 0.906000 0.999000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00108 0.08330 0.17200 0.22836 0.34800 0.83600   
## [1] "p-value ANOVA: 1.24124380786274e-274"  
## [1] "p-value Kruskal-Wallis: 1.11360835084654e-278"  
## [1] "p-values ValorsTest: "  
## [1] 5.555962e-161 0.000000e+00 1.102107e-13 0.000000e+00  
## [1] "Analysis by classes of the Variable: acousticness"



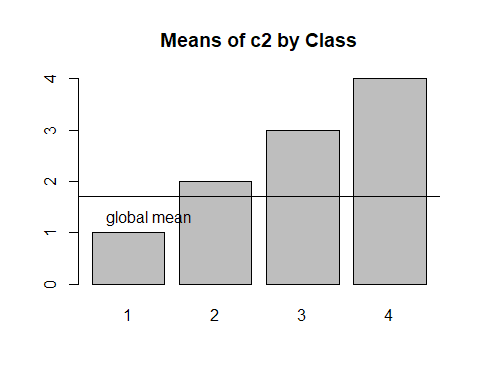
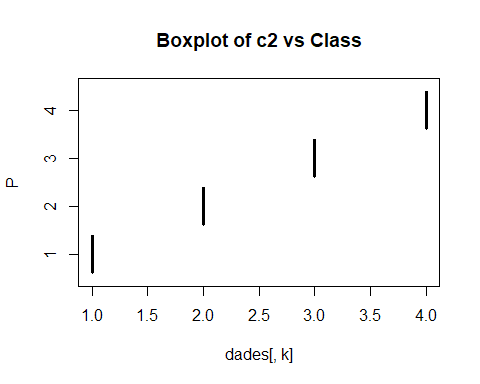
## [1] "Statistics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000001 0.006440 0.093600 0.197365 0.348500 0.905000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3830 0.7438 0.8410 0.8189 0.9170 0.9950   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000011 0.0003835 0.0147000 0.0970514 0.1352500 0.7660000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3190 0.7870 0.9310 0.8646 0.9850 0.9960   
## [1] "p-value ANOVA: 0"  
## [1] "p-value Kruskal-Wallis: 0"  
## [1] "p-values ValorsTest: "  
## [1] 0.000000e+00 6.393416e-212 0.000000e+00 2.450066e-123  
## [1] "Analysis by classes of the Variable: instrumentalness"



## [1] "Statistics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000000 0.0000000 0.0000049 0.0121457 0.0006955 0.4060000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000000 0.000000 0.000006 0.019713 0.001240 0.498000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.2130 0.6940 0.8440 0.7773 0.9010 0.9990   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0379 0.8050 0.8900 0.8312 0.9330 1.0000   
## [1] "p-value ANOVA: 0"  
## [1] "p-value Kruskal-Wallis: 0"  
## [1] "p-values ValorsTest: "  
## [1] 0.000000e+00 0.000000e+00 1.726791e-258 6.493344e-192  
## [1] "Analysis by classes of the Variable: valence"



## [1] "Statistics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0366 0.3370 0.5530 0.5458 0.7635 0.9850   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0219 0.2830 0.4585 0.4514 0.6085 0.9680   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.1378 0.3130 0.3702 0.5613 0.9640   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0454 0.1430 0.2167 0.2860 0.9730   
## [1] "p-value ANOVA: 6.3023780909863e-92"  
## [1] "p-value Kruskal-Wallis: 4.24544276547456e-94"  
## [1] "p-values ValorsTest: "  
## [1] 7.842021e-65 7.847820e-03 0.000000e+00 0.000000e+00  
## [1] "Analysis by classes of the Variable: c2"



## [1] "Statistics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1 1 1 1 1 1   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2 2 2 2 2 2   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3 3 3 3 3 3   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4 4 4 4 4 4   
## [1] "p-value ANOVA: NaN"  
## [1] "p-value Kruskal-Wallis: 0"  
## [1] "p-values ValorsTest: "  
## [1] 0.000000e+00 8.169678e-14 4.324152e-157 1.458199e-264

## Descriptions for every variable

### What patterns can we see for each class

* Class 1 (black) - Most loudest and energetic
* Class 2 (red) - Most acoustic
* Class 3 (green) - Most energetic and instrumental
* Class 4 (blue) - Most acoustic and instrumental

Happiest to saddest songs - Class 1 > 2 > 3 > 4

# Descriptors of the most significant classes. Add infoboxes  
for (c in 1:length(levels(as.factor(P)))) {  
 if(!is.na(levels(as.factor(P))[c])){  
 print(paste("P.values per class:",levels(as.factor(P))[c]));  
 print(sort(pvalk[c,]), digits=3)   
 }  
}

## [1] "P.values per class: 1"  
## key tempo\_cat acousticness instrumentalness   
## 0.00e+00 0.00e+00 0.00e+00 0.00e+00   
## c2 loudness energy valence   
## 0.00e+00 3.45e-166 5.56e-161 7.84e-65   
## [1] "P.values per class: 2"  
## key tempo\_cat loudness energy   
## 0.00e+00 0.00e+00 0.00e+00 0.00e+00   
## instrumentalness acousticness c2 valence   
## 0.00e+00 6.39e-212 8.17e-14 7.85e-03   
## [1] "P.values per class: 3"  
## key tempo\_cat acousticness valence   
## 0.00e+00 0.00e+00 0.00e+00 0.00e+00   
## instrumentalness c2 energy loudness   
## 1.73e-258 4.32e-157 1.10e-13 3.37e-03   
## [1] "P.values per class: 4"  
## key tempo\_cat loudness energy   
## 0.00e+00 0.00e+00 0.00e+00 0.00e+00   
## valence c2 instrumentalness acousticness   
## 0.00e+00 1.46e-264 6.49e-192 2.45e-123

Add the information of the modalities of qualitative to the list of pvalues and make global ordering

## Check song if fits profile

Based from the charts above, we describe the classes with these profiles:

* Class 1 - Mostly loudest
* Class 1 & 3 - Most energetic
* Class 2 & 4 - Most acoustic
* Class 3 & 4 - Most instrumental
* From Happiest to saddest Class 1, 2, 3, 4

We then picked the most popular song for every class and see if the profiling fits that song:

dd\_clust <- read.csv("data/cleaneddata-withtitles.csv") %>%  
 bind\_cols(tibble(cluster = c2))  
head(dd\_clust)

## popularity duration\_ms explicit danceability energy key loudness mode  
## 1 38 153039 False 0.522 0.854 A -5.274 minor  
## 2 20 212840 False 0.544 0.855 C -3.464 major  
## 3 71 235779 False 0.856 0.617 A#/Bb -5.308 minor  
## 4 12 142759 False 0.478 0.235 D -13.106 major  
## 5 9 292217 False 0.786 0.714 A -10.344 minor  
## 6 38 194506 False 0.650 0.771 D -6.901 major  
## speechiness acousticness instrumentalness liveness valence tempo  
## 1 0.0431 0.049800 0.000253 0.2650 0.699 136.087  
## 2 0.0282 0.003410 0.042800 0.2600 0.609 139.432  
## 3 0.0840 0.171000 0.000000 0.0857 0.759 136.081  
## 4 0.0278 0.917000 0.264000 0.0836 0.382 93.226  
## 5 0.0595 0.000726 0.908000 0.1010 0.269 119.956  
## 6 0.0269 0.001190 0.026100 0.1600 0.743 112.710  
## time\_signature track\_genre multiple\_artists tempo\_cat  
## 1 4 electronic FALSE Allegro  
## 2 4 power-pop FALSE Allegro  
## 3 4 k-pop FALSE Allegro  
## 4 3 honky-tonk FALSE Andante  
## 5 4 detroit-techno FALSE Moderato  
## 6 4 punk-rock FALSE Moderato  
## artists album\_name  
## 1 Babasónicos Infame  
## 2 Hoodoo Gurus Mars Needs Guitars!  
## 3 BTS Love Yourself 承 'Her'  
## 4 George Jones Mr. Country and Western  
## 5 Omar S Thank You for Letting Me Be Myself  
## 6 Patricio Rey y sus Redonditos de Ricota Último Bondi a Finisterre  
## track\_name cluster  
## 1 Once 1  
## 2 The Other Side Of Paradise - 2005 Remaster 1  
## 3 Go Go 1  
## 4 Your Tender Years 2  
## 5 Dumpster Graves 3  
## 6 Alien Duce 1

names(dd\_clust)

## [1] "popularity" "duration\_ms" "explicit" "danceability"   
## [5] "energy" "key" "loudness" "mode"   
## [9] "speechiness" "acousticness" "instrumentalness" "liveness"   
## [13] "valence" "tempo" "time\_signature" "track\_genre"   
## [17] "multiple\_artists" "tempo\_cat" "artists" "album\_name"   
## [21] "track\_name" "cluster"

Here’s the top 2 songs for every cluster. Try to listen to each song and check if it fits the profile we created.

pop\_per\_class <- dd\_clust %>%  
 select(popularity, track\_name, artists, track\_genre, cluster) %>%  
 arrange(desc(popularity)) %>%  
 group\_by(cluster) %>%  
 top\_n(6, popularity) %>%  
 arrange(cluster)  
pop\_per\_class

## # A tibble: 24 × 5  
## # Groups: cluster [4]  
## popularity track\_name artists track\_genre cluster  
## <int> <chr> <chr> <chr> <int>  
## 1 97 Tití Me Preguntó Bad Bunny latin 1  
## 2 93 Sweater Weather The Neighbourhood alt-rock 1  
## 3 92 I Wanna Be Yours Arctic Monkeys garage 1  
## 4 87 Daddy Issues The Neighbourhood alt-rock 1  
## 5 84 Jocelyn Flores XXXTENTACION emo 1  
## 6 83 Si Estuviésemos Juntos Bad Bunny latino 1  
## 7 77 Your Power Billie Eilish electro 2  
## 8 75 I Didn't Change My Number Billie Eilish electro 2  
## 9 75 I Drink Wine Adele british 2  
## 10 73 State Lines Novo Amor ambient 2  
## # … with 14 more rows

ggplot(pop\_per\_class) +  
 geom\_bar(aes(x = cluster, fill = track\_genre))

