INFO 523 - Association Rules

Noah Giebink and Sebastian Deimen

March 3, 2020

Choosing Variables

```
# select relevant
spot <- spot %>% select(track.popularity,
                        danceability, energy, key, loudness,
                        speechiness, acousticness, instrumentalness,
                        liveness, valence, tempo, duration_min,
                        happiness, median_age, percent_urban,
                        percent_internet_users, density_sqkm,
                        freedom, gdp)

spot_cor <- cor(spot) # make correlation matrix
corrplot(spot_cor, method = 'number', number.digits = 2,
         number.cex = 3/5) # find correlations by visualizing corrplot
```

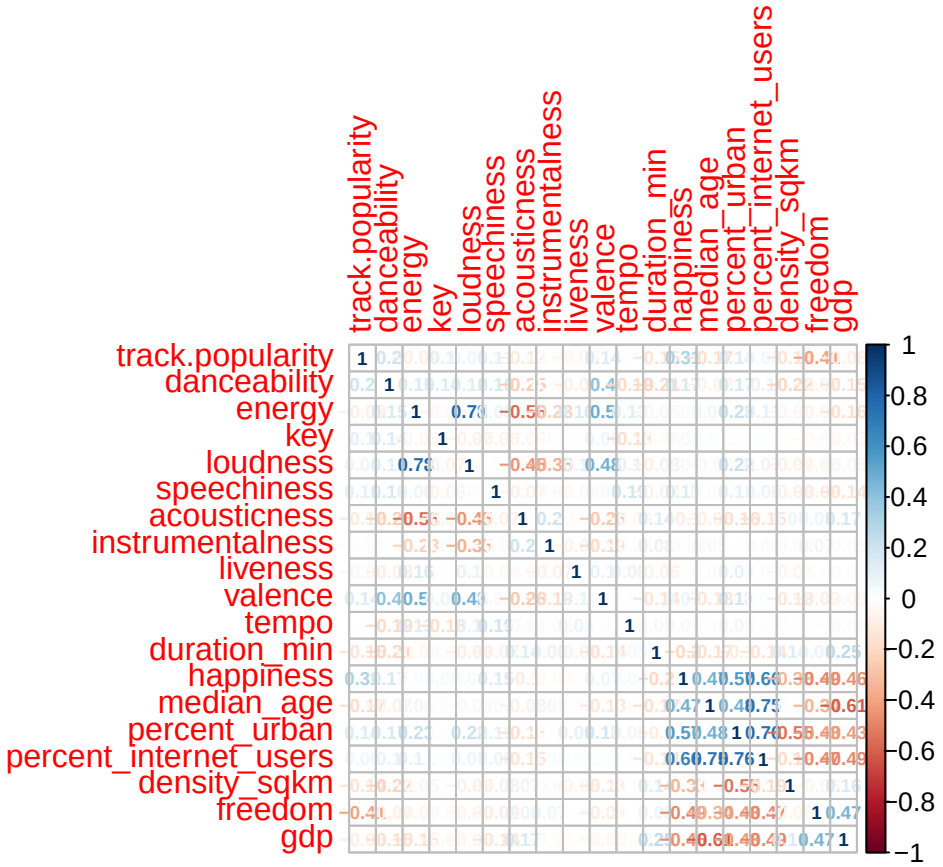

Figure 1. Correlation plot to weed out highly correlated variables.

We set an arbitrary correlation coefficient threshold of +/-0.75 to weed out correlated variables. As a result, we removed energy, median_age, and percent_urban.

Discretize variables

Track information variables

track.name, track.popularity

Audio metrics

danceability, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration_min,

Sociopolitical variables

happiness, percent_internet_users, density_sqkm, freedom, gdp

```
# select remaining variables
spot2 <- spot %>% select(track.popularity,
                         danceability, loudness,
                         speechiness, acousticness, instrumentalness,
                         liveness, valence, tempo, duration_min,
                         happiness,
                         percent_internet_users, density_sqkm,
                         freedom, gdp)
```

We discretized track.popularity separately because the vast majority of tracks in each country's Top 50 playlist were popular, as you would predict for Top 50 tracks! However, a small number of tracks with lower popularity do exist. We expect this to happen when songs in a country's Top 50 are popular locally, but obscure globally.

```
spot3 <- spot2 %>% select(-track.popularity)
pop <- spot2 %>% select(track.popularity)

# Discretize variables
  # function to discretize variables
disc <- function(x){
  cut(x, breaks = 4,
      labels = c('low', 'med-low', 'med-high', 'high'))}

  # apply disc fun to all dbl vars except track popularity
spot3_disc <- mutate_all(spot3, funs(disc))
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

```r
hist(pop$track.popularity)
```

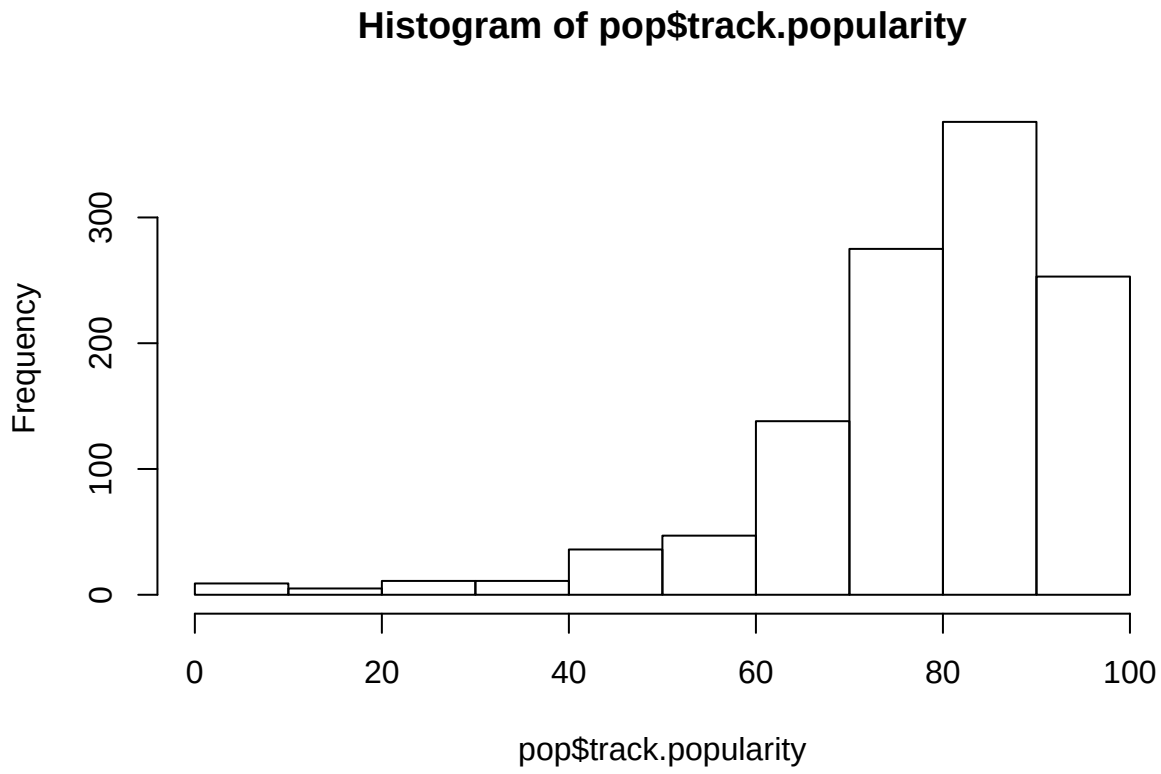

Figure 2. Distribution of track.popularity values for songs in Top 50 playlists for various countries. Values are left-skewed.

We are interested in discovering what social factors in individual countries are associated with preferences for local hits, which may show a less globalized music taste, as opposed to global hits. To compare these two categories, we separated track.popularity intro two bins: low and high, separated by the median track.popularity.

```r
track.popularity <- mutate(pop, track.popularity =
                             if_else(track.popularity > median(pop$track.popularity),
                                                        'high','low'))
```

Finally, we bound the two discretized data sets together.

```r
spot_disc <- cbind(spot3_disc, track.popularity)
spot_disc$track.popularity <- factor(track.popularity$track.popularity) # make factor
```

Plot distribution of levels for each variable

```r
spot_long <- pivot_longer(spot_disc, cols = colnames(spot_disc),
                          names_to = 'variable', values_to = 'level')
ggplot(spot_long, aes(level))+
  geom_bar()+
  facet_wrap(~variable)+
  theme(axis.text.x = element_text(angle = 90))
```

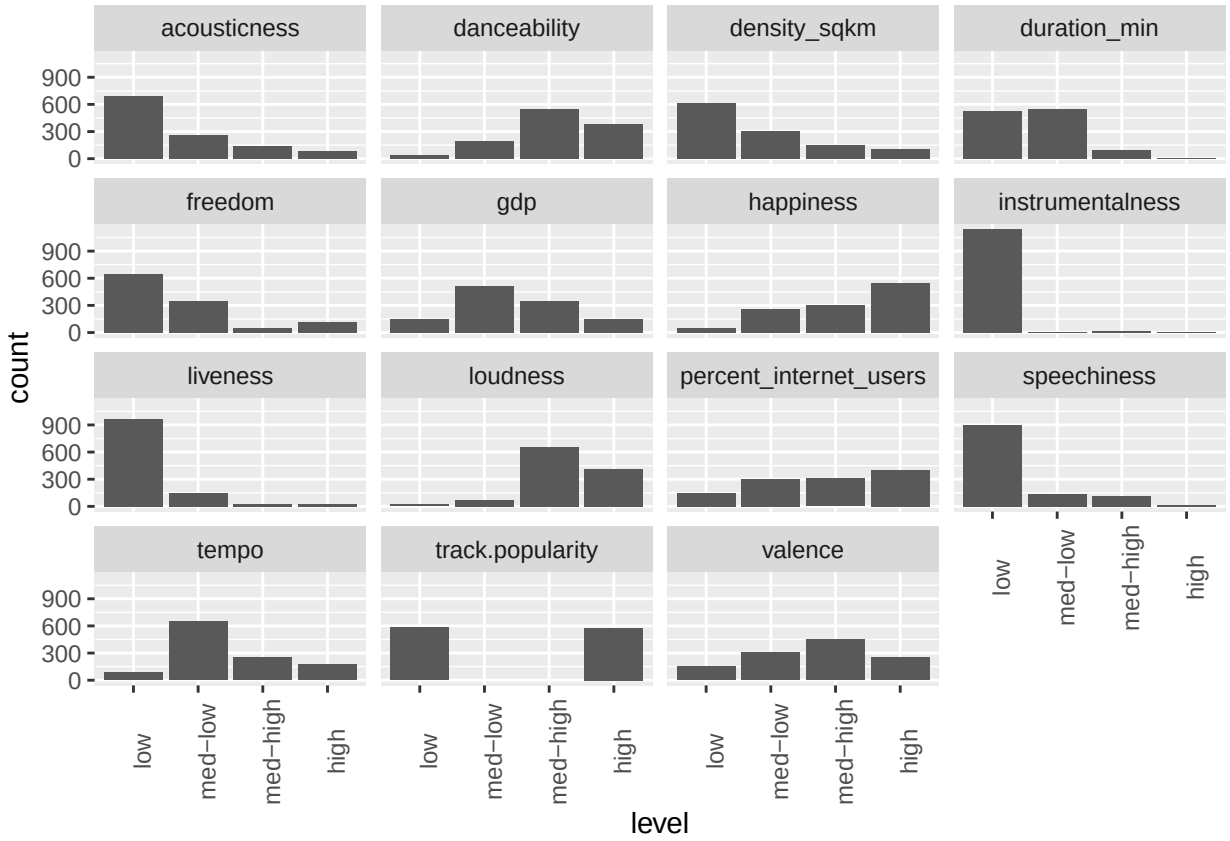

Figure 3. Distribution of discretized levels across all variables.

Most variables have a decent spread of values after discretization, except for instrumentalness, liveness, and speechiness. Since we think this is due to their irrelevance to the type of tracks found in the Top 50, we chose to omit these variables from association rule mining.

Our remaining variables are the following:

```
# The remaining dataset
spot_disc <- select(spot_disc, -instrumentalness, -liveness, -speechiness)
variable.names(spot_disc)
```

```
##  [1] "danceability"           "loudness"
##  [3] "acousticness"           "valence"
##  [5] "tempo"                  "duration_min"
##  [7] "happiness"              "percent_internet_users"
##  [9] "density_sqkm"           "freedom"
## [11] "gdp"                    "track.popularity"
```

Make transactional database

```
# make transactional dataset
spot_trans <- as(spot_disc, 'transactions')
inspect(spot_trans[1])
```

```
##     items                              transactionID
## [1] {danceability=high,
##      loudness=high,
```

```
##       acousticness=med-low,
##       valence=med-high,
##       tempo=med-low,
##       duration_min=med-low,
##       happiness=med-low,
##       percent_internet_users=med-high,
##       density_sqkm=med-low,
##       freedom=med-low,
##       gdp=med-high,
##       track.popularity=high}                        1
```

Plot frequent itemsets

```
itemFrequencyPlot(spot_trans, support = 0.2, cex.names = 0.8)
```

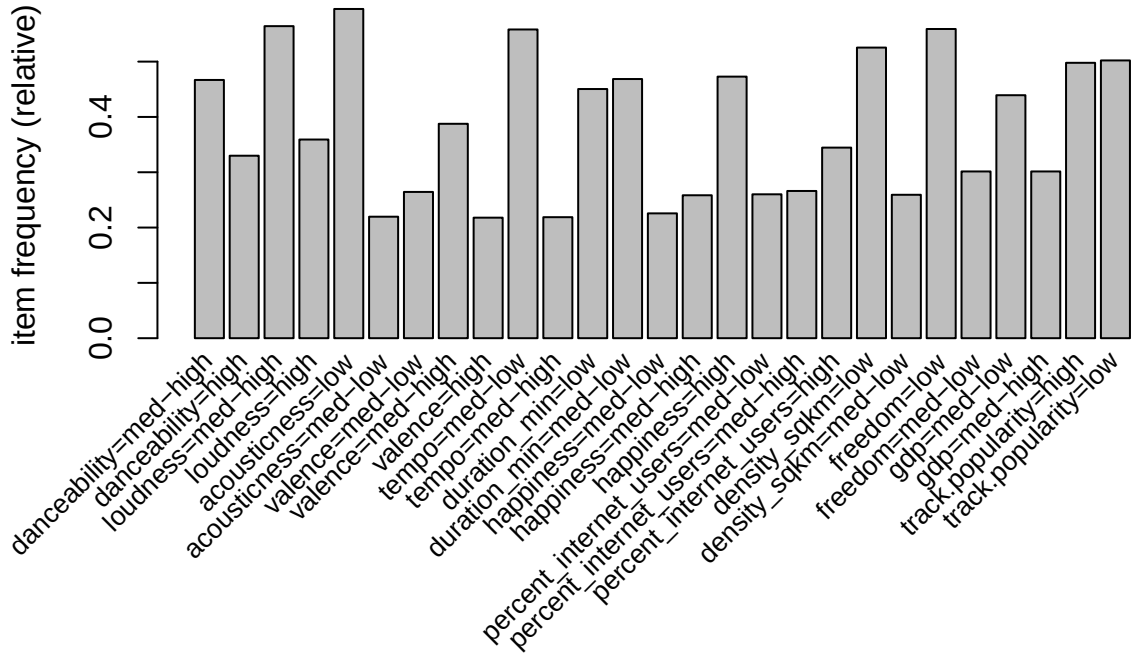

Figure 4. Most frequent itemsets (support above 0.2).

Mine and Inspect Frequent Itemsets

```
# frequent sets
sets <- apriori(spot_trans, parameter = list(support = 0.05, target = 'frequent itemsets'))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5    0.05      1
```

```
##  maxlen            target   ext
##      10 frequent itemsets FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 58
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[46 item(s), 1161 transaction(s)] done [0.00s].
## sorting and recoding items ... [41 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.03s].
## writing ... [3836 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
  # just closed
closed = sets[is.closed(sets)]
summary(closed)
```

```
## set of 3140 itemsets
##
## most frequent items:
##           freedom=low      acousticness=low      density_sqkm=low
##                  1070                   928                   835
##         tempo=med-low track.popularity=high               (Other)
##                   817                   793                  7034
##
## element (itemset/transaction) length distribution:sizes
##    1    2    3    4    5    6    7
##   36  337 1033 1126  489  114    5
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   3.655   4.000   7.000
##
## summary of quality measures:
##     support            count
##  Min.   :0.05082   Min.   : 59
##  1st Qu.:0.05857   1st Qu.: 68
##  Median :0.07149   Median : 83
##  Mean   :0.08871   Mean   :103
##  3rd Qu.:0.09819   3rd Qu.:114
##  Max.   :0.59518   Max.   :691
##
## includes transaction ID lists: FALSE
##
## mining info:
##        data ntransactions support confidence
##  spot_trans          1161    0.05          1
```

```
  # just max
max = sets[is.maximal(sets)]
summary(max)
```

```
## set of 1247 itemsets
```

```
##
## most frequent items:
##       freedom=low  acousticness=low  density_sqkm=low     tempo=med-low
##               428               410               365               345
## loudness=med-high           (Other)
##               312              3174
##
## element (itemset/transaction) length distribution:sizes
##   1   2   3   4   5   6   7
##   2  46 331 495 276  92   5
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   4.000   4.037   5.000   7.000
##
## summary of quality measures:
##     support            count
##  Min.   :0.05082   Min.   : 59.00
##  1st Qu.:0.05340   1st Qu.: 62.00
##  Median :0.05685   Median : 66.00
##  Mean   :0.05821   Mean   : 67.58
##  3rd Qu.:0.06202   3rd Qu.: 72.00
##  Max.   :0.08699   Max.   :101.00
##
## includes transaction ID lists: FALSE
##
## mining info:
##        data ntransactions support confidence
##  spot_trans          1161    0.05          1
```

```
# inspect top 10 frequent itemsets
inspect(head(closed, n=20, by='support'))
```

```
##      items                                       support   count
## [1]  {acousticness=low}                          0.5951766 691
## [2]  {loudness=med-high}                         0.5641688 655
## [3]  {freedom=low}                               0.5590009 649
## [4]  {tempo=med-low}                             0.5581395 648
## [5]  {density_sqkm=low}                          0.5254091 610
## [6]  {track.popularity=low}                      0.5021533 583
## [7]  {track.popularity=high}                     0.4978467 578
## [8]  {happiness=high}                            0.4728682 549
## [9]  {duration_min=med-low}                      0.4685616 544
## [10] {danceability=med-high}                     0.4668389 542
## [11] {duration_min=low}                          0.4504737 523
## [12] {gdp=med-low}                               0.4392765 510
## [13] {valence=med-high}                          0.3875969 450
## [14] {happiness=high,freedom=low}                0.3867356 449
## [15] {loudness=high}                             0.3591731 417
## [16] {acousticness=low,tempo=med-low}            0.3574505 415
## [17] {percent_internet_users=high,freedom=low}   0.3445306 400
## [18] {freedom=low,gdp=med-low}                   0.3445306 400
## [19] {density_sqkm=low,freedom=low}              0.3445306 400
## [20] {acousticness=low,freedom=low}              0.3436693 399
```

Mining for Rules

How are social factors associated with global patterns in music taste?

```
# conservative rules set (high min confidence)
rules <- apriori(spot_trans, parameter = list(support = 0.05,
                                              confidence = 0.7,
                                              target = 'rules'))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.05      1
##  maxlen target   ext
##      10  rules FALSE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 58
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[46 item(s), 1161 transaction(s)] done [0.00s].
## sorting and recoding items ... [41 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 done [0.02s].
## writing ... [3474 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules)
```

```
## set of 3474 rules
##
## rule length distribution (lhs + rhs):sizes
##    2    3    4    5    6    7
##   19  374 1293 1327  436   25
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   4.000   5.000   4.536   5.000   7.000
##
## summary of quality measures:
##     support          confidence          lift            count
##  Min.   :0.05082   Min.   :0.7000   Min.   : 1.176   Min.   : 59.00
##  1st Qu.:0.05512   1st Qu.:0.7398   1st Qu.: 1.411   1st Qu.: 64.00
##  Median :0.06460   Median :0.7977   Median : 1.601   Median : 75.00
##  Mean   :0.07453   Mean   :0.8364   Mean   : 1.780   Mean   : 86.53
##  3rd Qu.:0.08269   3rd Qu.:0.9869   3rd Qu.: 1.898   3rd Qu.: 96.00
##  Max.   :0.38674   Max.   :1.0000   Max.   :10.366   Max.   :449.00
##
## mining info:
##        data ntransactions support confidence
##  spot_trans          1161    0.05        0.7
```

A general look at our rules:

```
# inspect returns table with columns lhs 'lefthand side', rhs 'rhs'
inspect(head(rules, n=10, decreasing = TRUE, by = "lift"))
```

```
##      lhs                                  rhs                    support confidence      lift count
## [1]  {happiness=med-low,
##       percent_internet_users=med-high,
##       gdp=med-low}                     => {freedom=high}      0.05167959  1.0000000 10.366071    60
## [2]  {happiness=med-low,
##       percent_internet_users=med-high,
##       density_sqkm=low}                => {freedom=high}      0.05167959  1.0000000 10.366071    60
## [3]  {happiness=med-low,
##       percent_internet_users=med-high,
##       density_sqkm=low,
##       gdp=med-low}                     => {freedom=high}      0.05167959  1.0000000 10.366071    60
## [4]  {duration_min=med-low,
##       happiness=med-low,
##       density_sqkm=med-low,
##       track.popularity=low}            => {gdp=high}          0.05254091  0.8356164  6.382570    61
## [5]  {happiness=med-low,
##       density_sqkm=med-low,
##       track.popularity=low}            => {gdp=high}          0.07062877  0.7809524  5.965038    82
## [6]  {duration_min=med-low,
##       happiness=med-low,
##       density_sqkm=med-low}            => {gdp=high}          0.06029285  0.7142857  5.455827    70
## [7]  {loudness=med-high,
##       happiness=med-low,
##       density_sqkm=med-low}            => {gdp=high}          0.05340224  0.7126437  5.443285    62
## [8]  {freedom=high}                   => {happiness=med-low} 0.09646856  1.0000000  4.431298   112
## [9]  {percent_internet_users=med-high,
##       freedom=high}                    => {happiness=med-low} 0.05167959  1.0000000  4.431298    60
## [10] {freedom=high,
##       gdp=med-low}                     => {happiness=med-low} 0.05167959  1.0000000  4.431298    60
```

To examine how social factors are associated with global music tastes, we distinguished social variables from those dealing with the sonic qualities of music.

```
# get frequent colnames
music <- colnames(spot_trans[,1:24])
social <- colnames(spot_trans[,25:ncol(spot_trans)])
```

Here is an example where we associated social factors with music qualities.

```
inspect(head(n = 20, subset(rules, subset=(lhs %in% social & rhs %in% music))))
```

```
##      lhs                                  rhs                       support confidence     lift cou
## [1]  {gdp=low}                        => {loudness=med-high}     0.09302326  0.7248322 1.284779   1
## [2]  {gdp=low,
##       track.popularity=low}            => {duration_min=low}      0.05857020  0.7010309 1.556208
## [3]  {duration_min=low,
##       gdp=low}                         => {loudness=med-high}     0.06201550  0.7741935 1.372273
## [4]  {duration_min=low,
##       gdp=low}                         => {acousticness=low}      0.05857020  0.7311828 1.228514
## [5]  {happiness=high,
##       gdp=low}                         => {loudness=med-high}     0.09302326  0.7248322 1.284779   1
## [6]  {gdp=low,
##       track.popularity=low}            => {loudness=med-high}     0.06804479  0.8144330 1.443598
```

```
## [7]  {freedom=low,
##       gdp=low}                          => {loudness=med-high}     0.09302326  0.7248322 1.284779   10
## [8]  {acousticness=low,
##       gdp=low}                          => {loudness=med-high}     0.06373816  0.7184466 1.273460    7
## [9]  {happiness=high,
##       density_sqkm=med-high}            => {loudness=med-high}     0.06287683  0.7300000 1.293939    7
## [10] {tempo=high,
##       gdp=med-high}                     => {danceability=med-high} 0.06029285  0.7608696 1.629833    7
## [11] {tempo=high,
##       track.popularity=high}            => {duration_min=med-low}  0.06804479  0.7181818 1.532737    7
## [12] {valence=high,
##       gdp=med-high}                     => {acousticness=low}      0.07321275  0.7083333 1.190123    8
## [13] {valence=high,
##       gdp=med-low}                      => {duration_min=low}      0.06029285  0.7368421 1.635705    7
## [14] {valence=high,
##       gdp=med-low}                      => {acousticness=low}      0.05770887  0.7052632 1.184965    6
## [15] {valence=high,
##       happiness=high}                   => {acousticness=low}      0.07321275  0.7391304 1.241867    8
## [16] {valence=high,
##       track.popularity=high}            => {acousticness=low}      0.10508183  0.8079470 1.357491   11
## [17] {valence=high,
##       density_sqkm=low}                 => {acousticness=low}      0.10249785  0.7083333 1.190123   11
## [18] {valence=high,
##       freedom=low}                      => {acousticness=low}      0.08268734  0.7164179 1.203707    9
## [19] {tempo=med-high,
##       percent_internet_users=med-high} => {duration_min=low}      0.05254091  0.7721519 1.714089    6
## [20] {tempo=med-high,
##       track.popularity=high}            => {danceability=med-high} 0.06632214  0.8555556 1.832657    7
```

To tease apart *only* the affect of social factors, we needed to specify the lhs to not include music variables. Here, we sort by lift to get a sense of whether any of the rules contain associations that co-occur more often than is likely by chance.

```
inspect(head(n = 20, by = 'lift',
             subset(rules, subset=(lhs %in% social &
                                   !(lhs %in% music) &
                                   rhs %in% music))))
```

```
##      lhs                                  rhs                       support confidence     lift coun
## [1]  {percent_internet_users=med-high,
##       freedom=low,
##       track.popularity=low}            => {duration_min=low}     0.06804479  0.7669903 1.702630    7
## [2]  {percent_internet_users=med-high,
##       gdp=med-low,
##       track.popularity=low}            => {duration_min=low}     0.06201550  0.7578947 1.682439    7
## [3]  {percent_internet_users=med-high,
##       density_sqkm=low,
##       gdp=med-low,
##       track.popularity=low}            => {duration_min=low}     0.06201550  0.7578947 1.682439    7
## [4]  {density_sqkm=med-low,
##       freedom=low,
##       track.popularity=low}            => {loudness=med-high}    0.06201550  0.9230769 1.636171    7
## [5]  {happiness=high,
##       density_sqkm=med-low,
##       freedom=low,
```

```
##       track.popularity=low}            => {loudness=med-high}    0.06201550  0.9230769 1.636171    71
## [6]  {percent_internet_users=med-high,
##       freedom=low,
##       gdp=med-low}                     => {duration_min=low}     0.06287683  0.7300000 1.620516    73
## [7]  {percent_internet_users=med-high,
##       density_sqkm=low,
##       freedom=low,
##       gdp=med-low}                     => {duration_min=low}     0.06287683  0.7300000 1.620516    73
## [8]  {percent_internet_users=med-high,
##       density_sqkm=low,
##       track.popularity=low}            => {duration_min=low}     0.07149009  0.7280702 1.616232    83
## [9]  {happiness=med-low,
##       gdp=high,
##       track.popularity=low}            => {duration_min=med-low} 0.05254091  0.7439024 1.587630    61
## [10] {density_sqkm=med-low,
##       gdp=high,
##       track.popularity=low}            => {duration_min=med-low} 0.05254091  0.7439024 1.587630    61
## [11] {happiness=med-low,
##       density_sqkm=med-low,
##       gdp=high,
##       track.popularity=low}            => {duration_min=med-low} 0.05254091  0.7439024 1.587630    61
## [12] {gdp=low,
##       track.popularity=low}            => {duration_min=low}     0.05857020  0.7010309 1.556208    68
## [13] {happiness=high,
##       gdp=low,
##       track.popularity=low}            => {duration_min=low}     0.05857020  0.7010309 1.556208    68
## [14] {freedom=low,
##       gdp=low,
##       track.popularity=low}            => {duration_min=low}     0.05857020  0.7010309 1.556208    68
## [15] {happiness=high,
##       freedom=low,
##       gdp=low,
##       track.popularity=low}            => {duration_min=low}     0.05857020  0.7010309 1.556208    68
## [16] {percent_internet_users=med-high,
##       gdp=med-low}                     => {duration_min=low}     0.09646856  0.7000000 1.553920   111
## [17] {percent_internet_users=med-high,
##       density_sqkm=low,
##       gdp=med-low}                     => {duration_min=low}     0.09646856  0.7000000 1.553920   111
## [18] {density_sqkm=med-low,
##       freedom=low}                     => {loudness=med-high}    0.07407407  0.8686869 1.539764    86
## [19] {happiness=high,
##       density_sqkm=med-low,
##       freedom=low}                     => {loudness=med-high}    0.07407407  0.8686869 1.539764    86
## [20] {happiness=high,
##       density_sqkm=med-low,
##       track.popularity=low}            => {loudness=med-high}    0.06890612  0.8602151 1.524748    80
```

Finally, our goal could be phrased in terms of a conditional probability: what kind of musical qualities do we see listeners engaging with, conditioned on social factors in their country? Therefore, we are most interested in rules with high confidence.

```
inspect(head(n = 20, by = 'confidence',
             subset(rules, subset=(lhs %in% social &
                                   !(lhs %in% music) &
                                   rhs %in% music))))
```

```
##      lhs                                  rhs                        support confidence     lift coun
## [1]  {density_sqkm=med-low,
##       freedom=low,
##       track.popularity=low}           => {loudness=med-high}    0.06201550  0.9230769 1.636171    7
## [2]  {happiness=high,
##       density_sqkm=med-low,
##       freedom=low,
##       track.popularity=low}           => {loudness=med-high}    0.06201550  0.9230769 1.636171    7
## [3]  {density_sqkm=med-low,
##       freedom=low}                    => {loudness=med-high}    0.07407407  0.8686869 1.539764    8
## [4]  {happiness=high,
##       density_sqkm=med-low,
##       freedom=low}                    => {loudness=med-high}    0.07407407  0.8686869 1.539764    8
## [5]  {happiness=high,
##       density_sqkm=med-low,
##       track.popularity=low}           => {loudness=med-high}    0.06890612  0.8602151 1.524748    8
## [6]  {gdp=low,
##       track.popularity=low}           => {loudness=med-high}    0.06804479  0.8144330 1.443598    7
## [7]  {happiness=high,
##       gdp=low,
##       track.popularity=low}           => {loudness=med-high}    0.06804479  0.8144330 1.443598    7
## [8]  {freedom=low,
##       gdp=low,
##       track.popularity=low}           => {loudness=med-high}    0.06804479  0.8144330 1.443598    7
## [9]  {happiness=high,
##       freedom=low,
##       gdp=low,
##       track.popularity=low}           => {loudness=med-high}    0.06804479  0.8144330 1.443598    7
## [10] {happiness=high,
##       freedom=low,
##       track.popularity=low}           => {loudness=med-high}    0.12575366  0.7724868 1.369248   14
## [11] {happiness=high,
##       percent_internet_users=med-high} => {loudness=med-high}    0.06546081  0.7676768 1.360722    7
## [12] {happiness=high,
##       percent_internet_users=med-high,
##       freedom=low}                    => {loudness=med-high}    0.06546081  0.7676768 1.360722    7
## [13] {percent_internet_users=med-high,
##       freedom=low,
##       track.popularity=low}           => {duration_min=low}     0.06804479  0.7669903 1.702630    7
## [14] {percent_internet_users=med-high,
##       gdp=med-low,
##       track.popularity=low}           => {duration_min=low}     0.06201550  0.7578947 1.682439    7
## [15] {percent_internet_users=med-high,
##       density_sqkm=low,
##       gdp=med-low,
##       track.popularity=low}           => {duration_min=low}     0.06201550  0.7578947 1.682439    7
## [16] {happiness=med-low,
##       gdp=high,
##       track.popularity=low}           => {duration_min=med-low} 0.05254091  0.7439024 1.587630    6
## [17] {density_sqkm=med-low,
##       gdp=high,
##       track.popularity=low}           => {duration_min=med-low} 0.05254091  0.7439024 1.587630    6
## [18] {happiness=med-low,
##       density_sqkm=med-low,
```

```
##       gdp=high,
##       track.popularity=low}            => {duration_min=med-low} 0.05254091  0.7439024 1.587630    6
## [19] {density_sqkm=med-low,
##       track.popularity=low}            => {loudness=med-high}    0.12489233  0.7323232 1.298057   14
## [20] {happiness=high,
##       density_sqkm=med-high}           => {loudness=med-high}    0.06287683  0.7300000 1.293939    7
```

```r
# save subset
rules_sub <- subset(rules, subset = lhs %in% social & !(lhs %in% music) & rhs %in% music)

# explore rules associated with valence only; take from less conservative set
rules_val <- subset(rules, subset = lhs %in% social & !(lhs %in% music) & rhs %pin% 'valence=')

# explore rules associated with track.popularity only
rules_pop <- subset(rules, subset = lhs %in% social & !(lhs %in% music) & rhs %pin% 'track.pop')

# what music taste is happiness associated with?
rules_happy <- subset(rules, subset = (lhs %pin% 'hap' & size(lhs)<2) & rhs %in% music & !(rhs %in% soci
```

Plot

```r
# grouped plot
plot(rules_sub, method = 'grouped')
```

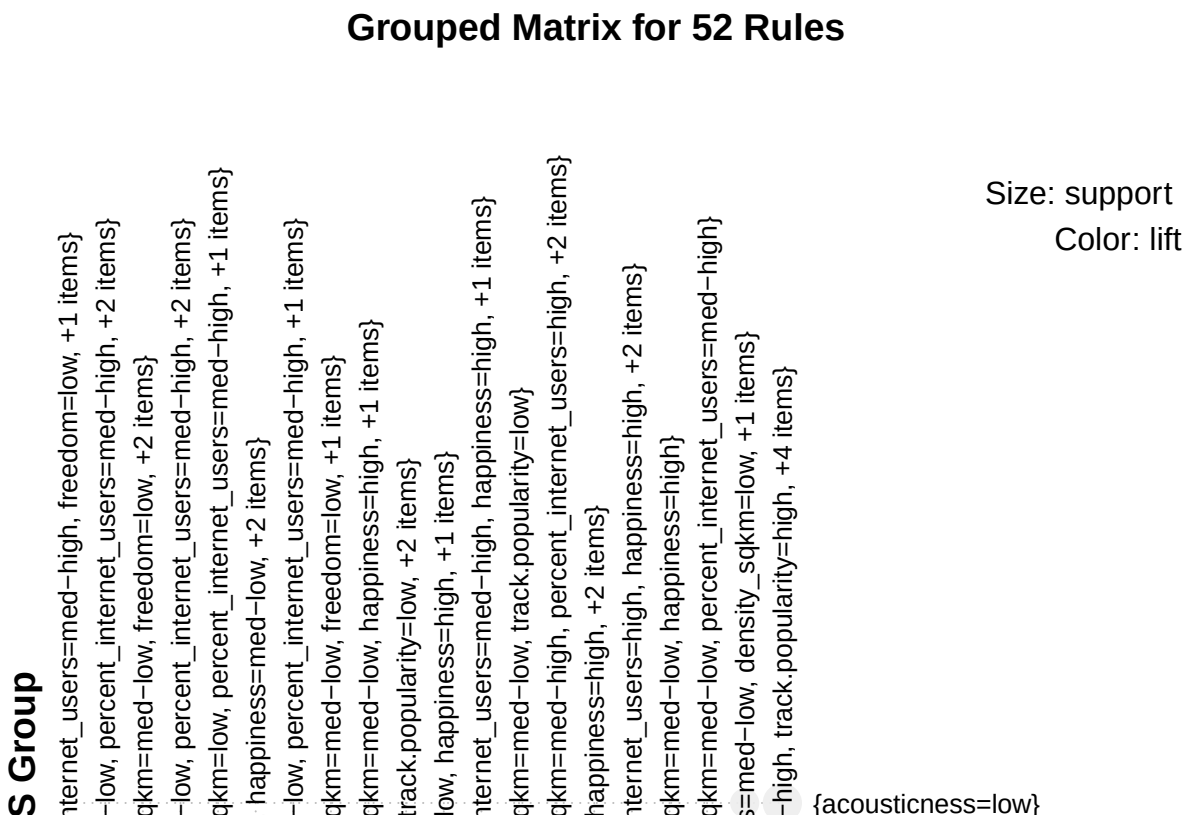

Figure 5. A sample of all rules generated.

```
plot(rules_pop, method = 'grouped')
```

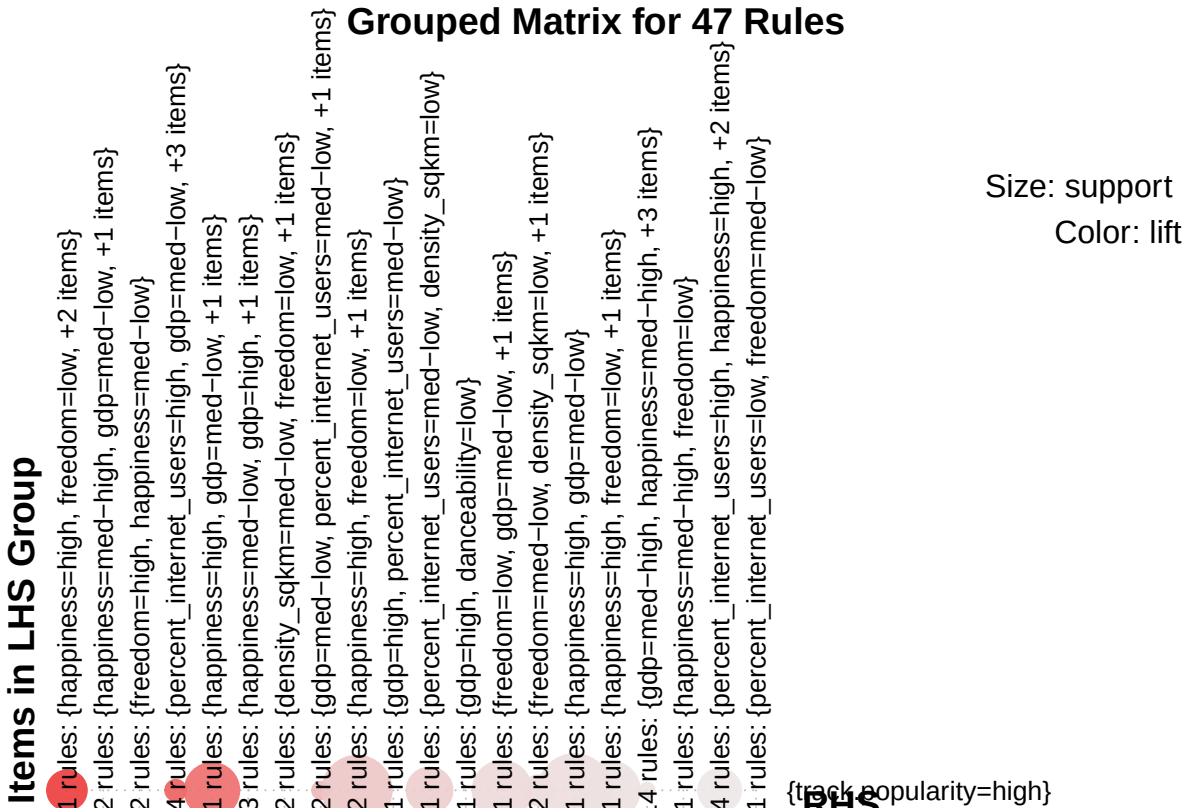

Figure 6. Rules specific to track.popularity.

```
plot(rules_val, method = 'grouped')
```

```
## Error in plot.rules(rules_val, method = "grouped"): x contains 0 rules!
```

We know there are no interesting associations with valence because the query produces **no rules** (and throws an informative error)! Similarly, we lack rules for citizens' self-reported happiness.

```
plot(rules_happy, method = 'grouped')
```

```
## Error in plot.rules(rules_happy, method = "grouped"): x contains 0 rules!
```

```
rules_pop10 <- head(rules_pop, n = 10, by = 'confidence')
plot(rules_pop10, method = 'graph', engine='htmlwidget')
```

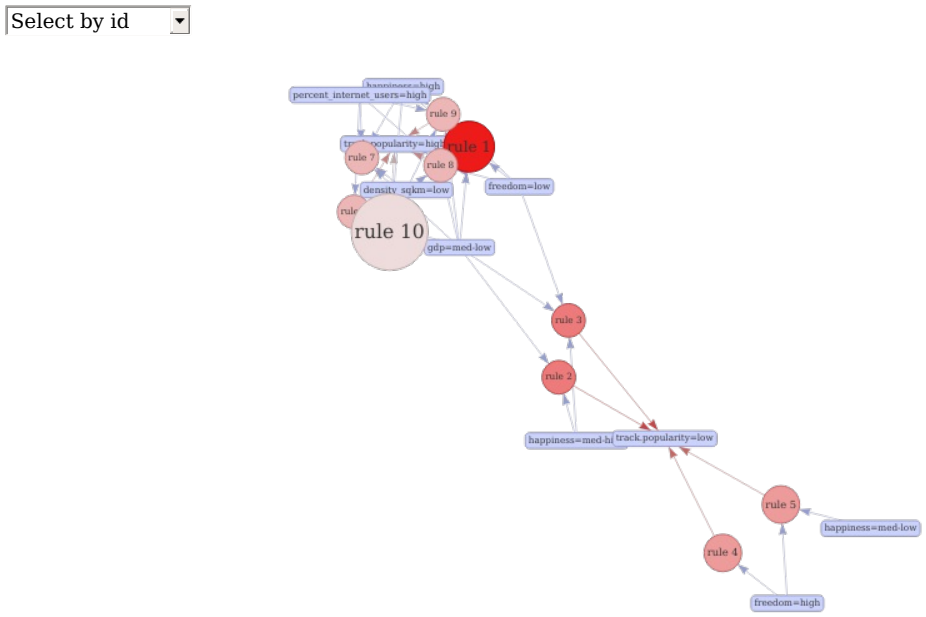

Figure 7. An interactive plot of the association rules for track.popularity.

Explanation of interesting rules

The top 10 rules dealing with track.popularity:

As outlined above, we chose these by systematically filtering a subset of rules and choosing those with highest confidence. We are especially interested in the top 2 rules, but consider aspects of the entire top 10 rules in our interpretation.

```
inspect(head(rules_pop, by = 'confidence',n=10))
```

```
##      lhs                                rhs                        support confidence     lift count
## [1]  {happiness=high,
##       density_sqkm=low,
##       freedom=low,
##       gdp=med-low}                   => {track.popularity=high} 0.11455642  0.8866667 1.781003   133
## [2]  {happiness=med-high,
##       gdp=med-low}                   => {track.popularity=low}  0.07579673  0.8800000 1.752453    88
## [3]  {happiness=med-high,
##       freedom=low,
##       gdp=med-low}                   => {track.popularity=low}  0.07579673  0.8800000 1.752453    88
## [4]  {freedom=high}                  => {track.popularity=low}  0.08440999  0.8750000 1.742496    98
## [5]  {happiness=med-low,
##       freedom=high}                  => {track.popularity=low}  0.08440999  0.8750000 1.742496    98
## [6]  {percent_internet_users=high,
##       density_sqkm=low,
##       gdp=med-low}                   => {track.popularity=high} 0.07407407  0.8600000 1.727439    86
## [7]  {happiness=high,
##       percent_internet_users=high,
##       density_sqkm=low,
##       gdp=med-low}                   => {track.popularity=high} 0.07407407  0.8600000 1.727439    86
```

```
## [8]  {percent_internet_users=high,
##       density_sqkm=low,
##       freedom=low,
##       gdp=med-low}                 => {track.popularity=high} 0.07407407  0.8600000 1.727439    86
## [9]  {happiness=high,
##       percent_internet_users=high,
##       density_sqkm=low,
##       freedom=low,
##       gdp=med-low}                 => {track.popularity=high} 0.07407407  0.8600000 1.727439    86
## [10] {happiness=high,
##       density_sqkm=low,
##       gdp=med-low}                 => {track.popularity=high} 0.14642550  0.8500000 1.707353   170
```

Tentative finding

Happy countries appear to enjoy the most globally popular music. Less happy countries enjoy more globally obscure music.

Among the top 10 rules, happiness=high was always associated with track.popularity=high. By contrast, several rules for track.popularity=low contained happiness=med-high, and in one case happiness=med-low.

Gdp=med-low and freedom=low were common in rules containing both track.popularity=high and track.popularity=low, making them unfit for teasing apart the two popularity levels on their own. In a future analysis, we could look for possible interactions between happiness and gdp or freedom.

Another interesting item in rules with track.popularity=high but not in track.popularity=low is percent_internet_users=high. This could align with our previously mentioned speculation that globally obscure tracks in a country's Top 50 may be a sign of reduced globalization in music tastes, because the internet is clearly a powerful force in globalization and mixing of ideas. It would be neccessary and interesting to assess the impact of internet access on music tastes in the future.

Mining rationale

We chose interesting rules with a mixture of subsetting to consider how social factors are associated with a country's listening preferences, omitting subsets which fail to produce rules (e.g. the case of valence, above), and selecting interesting rules by confidence, given that they also yield lift > 1. To compare associations across track.popularity=high and track.popularity=low, we chose rules with the highest confidence for each of these two levels on the rhs. We chose the rules with greatest confidence because we wanted to know what track.popularity would be, *conditioned* on social factors (i.e. to answer our question: how are social factors in individual countries associated with preferences for tracks with high or low global popularity?).