Austin Rodriquez 2/18/2023

```
if(!require("vcd")){
  install.packages("vcd")
}
```

```
## Loading required package: vcd
```

```
## Loading required package: grid
```

```
if(!require("e1071")){
  install.packages("vcd")
}
```

```
## Loading required package: e1071
```

3:

The linear models for classification work by finding a linear combination of factors to separate observations into regions. These regions should only contain observations of one type.

4: The data used is “Prediction of Music genre” by vicsuperman. The data can be accessed at: https://www.kaggle.com/datasets/vicsuperman/prediction-of-music-genre (https://www.kaggle.com/datasets/vicsuperman/prediction-of-music-genre)

The data will first be read in.

```
df <- read.csv("music_genre.csv")
```

We will have to remove null data since trying to attach a mean or median value to things like duration and energy don’t apply well to music.

Also, the website preview reveals that some tempos are unknown and left as “?”. These can be removed since there will still be plenty of data left over to use.

```
df <- na.omit(df)
df <- subset(df, tempo != "?")
```

To preview the data, we will peek at it.

```
str(df)
```

```
## 'data.frame':    45020 obs. of  18 variables:
##  $ instance_id     : num  32894 46652 30097 62177 24907 ...
##  $ artist_name     : chr  "Röyksopp" "Thievery Corporation" "Dillon Francis" "Dubloadz" ...
##  $ track_name      : chr  "Röyksopp's Night Out" "The Shining Path" "Hurricane" "Nitro" ...
##  $ popularity      : num  27 31 28 34 32 46 43 39 22 30 ...
##  $ acousticness    : num  0.00468 0.0127 0.00306 0.0254 0.00465 0.0289 0.0297 0.00299 0.00934 0.855 ...
##  $ danceability    : num  0.652 0.622 0.62 0.774 0.638 0.572 0.809 0.509 0.578 0.607 ...
##  $ duration_ms     : num  -1 218293 215613 166875 222369 ...
##  $ energy          : num  0.941 0.89 0.755 0.7 0.587 0.803 0.706 0.921 0.731 0.158 ...
##  $ instrumentalness: num  7.92e-01 9.50e-01 1.18e-02 2.53e-03 9.09e-01 7.74e-06 9.03e-01 2.76e-04 1.12e-02 0.0
0 ...
##  $ key             : chr  "A#" "D" "G#" "C#" ...
##  $ liveness        : num  0.115 0.124 0.534 0.157 0.157 0.106 0.0635 0.178 0.111 0.106 ...
##  $ loudness        : num  -5.2 -7.04 -4.62 -4.5 -6.27 ...
##  $ mode            : chr  "Minor" "Minor" "Major" "Major" ...
##  $ speechiness     : num  0.0748 0.03 0.0345 0.239 0.0413 0.351 0.0484 0.268 0.173 0.0345 ...
##  $ tempo           : chr  "100.889" "115.00200000000001" "127.994" "128.014" ...
##  $ obtained_date   : chr  "4-Apr" "4-Apr" "4-Apr" "4-Apr" ...
##  $ valence         : num  0.759 0.531 0.333 0.27 0.323 0.23 0.761 0.273 0.203 0.307 ...
##  $ music_genre     : chr  "Electronic" "Electronic" "Electronic" "Electronic" ...
```

Columns to change to factors: key, music genre, mode Columns to change to number: tempo

```
df$key <- factor(df$key)
df$music_genre <- factor(df$music_genre)
df$mode <- factor(df$mode)
df$tempo <- as.numeric(df$tempo)
```

a: The data is now ready to be divided up.

```
set.seed(123)
i <- sample(1:nrow(df), nrow(df) * .80, replace = FALSE)
train <- df[i,]
test <- df[-i,]
```

b: The five basic R functions will be used to explore the training data.

```
names(train)
```

```
##  [1] "instance_id"      "artist_name"      "track_name"       "popularity"
##  [5] "acousticness"     "danceability"     "duration_ms"      "energy"
##  [9] "instrumentalness" "key"              "liveness"         "loudness"
## [13] "mode"             "speechiness"      "tempo"            "obtained_date"
## [17] "valence"          "music_genre"
```

```
dim(train)
```

```
## [1] 36016    18
```

```
summary(train)
```

```
##   instance_id    artist_name         track_name          popularity
##  Min.   :20002   Length:36016       Length:36016       Min.   : 0.00
##  1st Qu.:38074   Class :character   Class :character   1st Qu.:34.00
##  Median :55864   Mode  :character   Mode  :character   Median :45.00
##  Mean   :55899                                         Mean   :44.33
##  3rd Qu.:73884                                         3rd Qu.:56.00
##  Max.   :91758                                         Max.   :99.00
##
##   acousticness     danceability     duration_ms          energy
##  Min.   :0.0000   Min.   :0.0596   Min.   :     -1   Min.   :0.000795
##  1st Qu.:0.0198   1st Qu.:0.4430   1st Qu.: 174690   1st Qu.:0.433000
##  Median :0.1430   Median :0.5690   Median : 219120   Median :0.645000
##  Mean   :0.3056   Mean   :0.5588   Mean   : 220578   Mean   :0.600872
##  3rd Qu.:0.5500   3rd Qu.:0.6870   3rd Qu.: 268161   3rd Qu.:0.817000
##  Max.   :0.9960   Max.   :0.9860   Max.   :4497994   Max.   :0.999000
##
##  instrumentalness        key           liveness          loudness
##  Min.   :0.000000   G      : 4116   Min.   :0.00967   Min.   :-47.046
##  1st Qu.:0.000000   C      : 3990   1st Qu.:0.09680   1st Qu.:-10.836
##  Median :0.000159   C#     : 3842   Median :0.12600   Median : -7.270
##  Mean   :0.181304   D      : 3776   Mean   :0.19346   Mean   : -9.118
##  3rd Qu.:0.151000   A      : 3509   3rd Qu.:0.24300   3rd Qu.: -5.166
##  Max.   :0.996000   F      : 3107   Max.   :1.00000   Max.   :  3.744
##                     (Other):13676
##     mode        speechiness          tempo        obtained_date
##  Major:23092   Min.   :0.02230   Min.   : 34.35   Length:36016
##  Minor:12924   1st Qu.:0.03610   1st Qu.: 94.97   Class :character
##                Median :0.04890   Median :119.92   Mode  :character
##                Mean   :0.09380   Mean   :120.03
##                3rd Qu.:0.09932   3rd Qu.:140.57
##                Max.   :0.94200   Max.   :220.04
##
##     valence            music_genre
##  Min.   :0.0000   Rock       : 3687
##  1st Qu.:0.2590   Alternative: 3640
##  Median :0.4510   Hip-Hop    : 3609
##  Mean   :0.4579   Jazz       : 3607
##  3rd Qu.:0.6500   Blues      : 3603
##  Max.   :0.9920   Rap        : 3587
##                   (Other)    :14283
```

```
str(train)
```

```
## 'data.frame':    36016 obs. of  18 variables:
##  $ instance_id     : num  79146 20775 51166 78444 58918 ...
##  $ artist_name     : chr  "The Prodigy" "Bo Diddley" "Steely Dan" "César Franck" ...
##  $ track_name      : chr  "No Good (Start The Dance)" "Long Distance Call" "I Got The News" "Violin Sonata in
A Major, FWV 8: Allegro" ...
##  $ popularity      : num  45 27 38 27 36 27 45 40 44 44 ...
##  $ acousticness    : num  6.07e-03 6.53e-02 4.04e-03 9.65e-01 1.59e-03 6.85e-01 9.09e-01 7.57e-03 1.05e-03 5.5
6e-05 ...
##  $ danceability    : num  0.645 0.47 0.62 0.166 0.699 0.473 0.539 0.56 0.498 0.353 ...
##  $ duration_ms     : num  379907 -1 307107 479640 244800 ...
##  $ energy          : num  0.996 0.417 0.471 0.129 0.974 0.397 0.182 0.621 0.981 0.772 ...
##  $ instrumentalness: num  7.28e-01 4.29e-05 1.71e-02 2.49e-01 7.41e-01 1.67e-05 0.00 5.14e-02 8.08e-05 2.55e-0
2 ...
##  $ key             : Factor w/ 12 levels "A","A#","B","C",..: 9 8 4 10 9 4 8 6 11 6 ...
##  $ liveness        : num  0.128 0.0959 0.0336 0.0658 0.108 0.527 0.137 0.271 0.081 0.233 ...
##  $ loudness        : num  -4.34 -11.81 -14.35 -20.72 -4.4 ...
##  $ mode            : Factor w/ 2 levels "Major","Minor": 2 1 1 1 1 1 1 1 1 1 ...
##  $ speechiness     : num  0.0558 0.0415 0.0545 0.0614 0.0688 0.313 0.0433 0.0336 0.122 0.0294 ...
##  $ tempo           : num  145 171.1 121.3 73.7 150 ...
##  $ obtained_date   : chr  "4-Apr" "4-Apr" "4-Apr" "4-Apr" ...
##  $ valence         : num  0.169 0.733 0.847 0.0386 0.453 0.649 0.565 0.231 0.355 0.584 ...
##  $ music_genre     : Factor w/ 10 levels "Alternative",..: 6 3 3 4 6 4 8 3 1 1 ...
```

```
head(train)
```

```
##       instance_id       artist_name
## 3337        79146       The Prodigy
## 33320       20775        Bo Diddley
## 33075       51166        Steely Dan
## 41696       78444      César Franck
## 3088        58918              Jauz
## 43251       81712 Leonard Bernstein
##                                                 track_name popularity
## 3337                                No Good (Start The Dance)         45
## 33320                                      Long Distance Call         27
## 33075                                          I Got The News         38
## 41696                 Violin Sonata in A Major, FWV 8: Allegro         27
## 3088                                             Pure Evil         36
## 43251 West Side Story (Original Broadway Cast): Act I: America         27
##       acousticness danceability duration_ms energy instrumentalness key
## 3337       0.00607        0.645      379907  0.996         7.28e-01   F
## 33320      0.06530        0.470          -1  0.417         4.29e-05   E
## 33075      0.00404        0.620      307107  0.471         1.71e-02   C
## 41696      0.96500        0.166      479640  0.129         2.49e-01  F#
## 3088       0.00159        0.699      244800  0.974         7.41e-01   F
## 43251      0.68500        0.473      272493  0.397         1.67e-05   C
##       liveness loudness  mode speechiness   tempo obtained_date valence
## 3337    0.1280   -4.339 Minor      0.0558 145.002         4-Apr  0.1690
## 33320   0.0959  -11.815 Major      0.0415 171.114         4-Apr  0.7330
## 33075   0.0336  -14.346 Major      0.0545 121.254         4-Apr  0.8470
## 41696   0.0658  -20.724 Major      0.0614  73.716         4-Apr  0.0386
## 3088    0.1080   -4.399 Major      0.0688 150.016         4-Apr  0.4530
## 43251   0.5270  -15.271 Major      0.3130 162.058         3-Apr  0.6490
##       music_genre
## 3337   Electronic
## 33320       Blues
## 33075       Blues
## 41696   Classical
## 3088   Electronic
## 43251   Classical
```

There are about 36000 entries in the training data. The data reveals various songs and information ranging from its name to popularity are included

In order to create a binary target to build models around, popularity will be converted to a factor. Since the 3rd quarterly value for popularity is 56, we will round up and say that a song is considered “popular” if it is at least ranked 60.

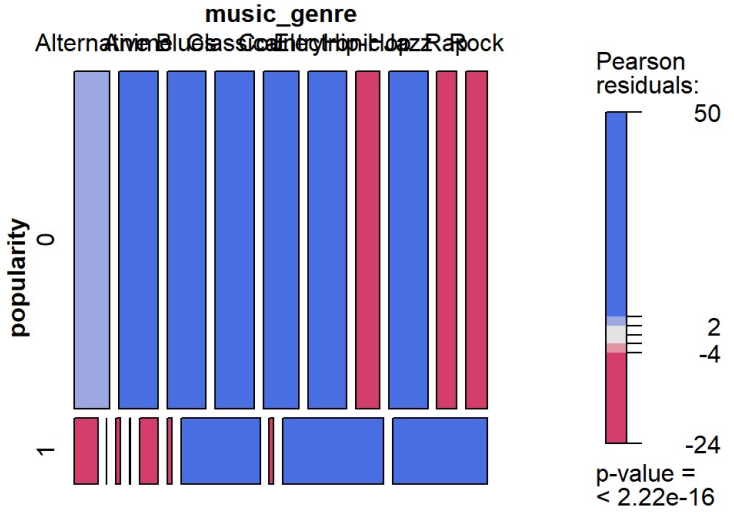
```
train$popularity <- as.factor(ifelse (train$popularity >= 60, 1, 0))

test$popularity <- as.factor(ifelse (test$popularity >= 60, 1, 0))
```

```
levels(train$popularity)
```

```
## [1] "0" "1"
```

b: Now we can make some graphs to see how some factors affect the popularity.

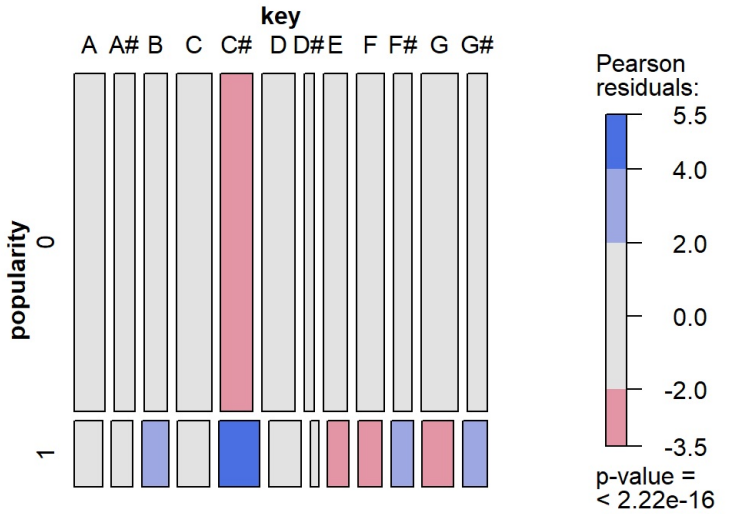
```
library(vcd)
mosaic(table(train[,c(4,18)]), shade = TRUE, legend = TRUE)
```



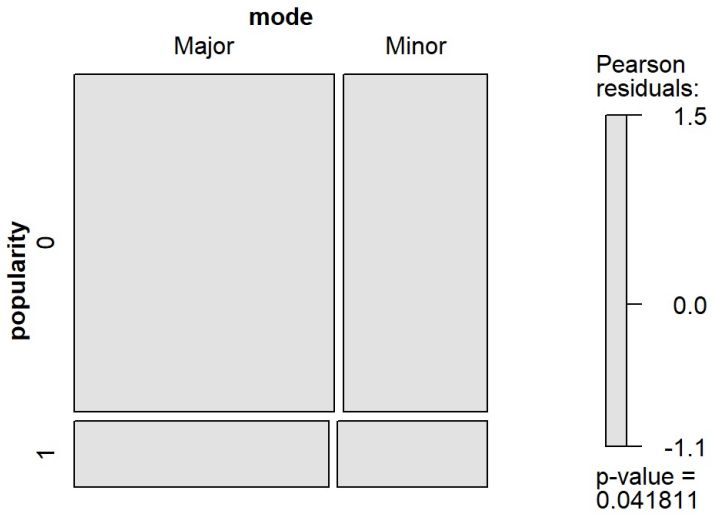

```
levels(train$music_genre)
```

```
##  [1] "Alternative" "Anime"       "Blues"       "Classical"   "Country"
##  [6] "Electronic"  "Hip-Hop"     "Jazz"        "Rap"         "Rock"
```

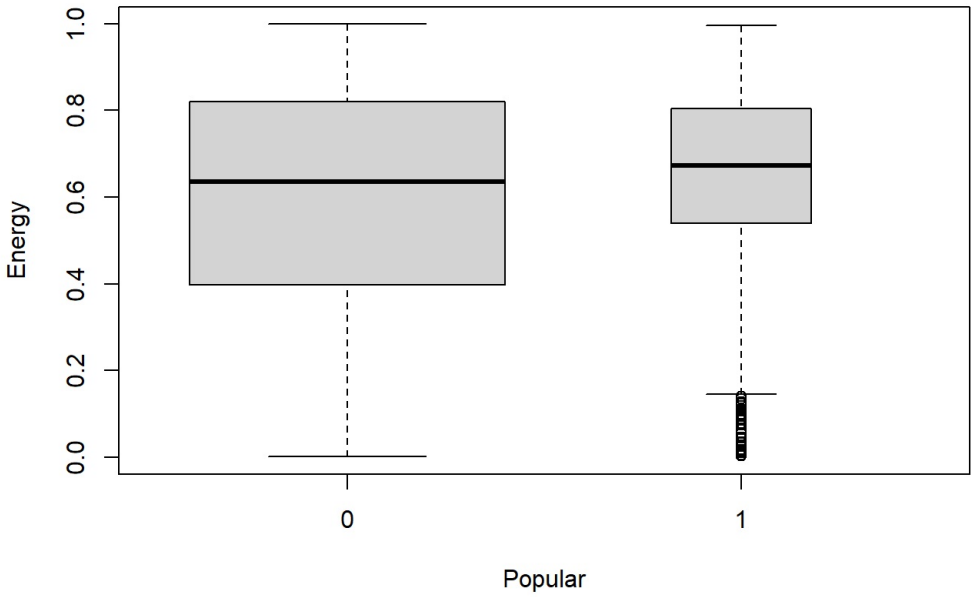
```
mosaic(table(train[,c(4,10)]), shade = TRUE, legend = TRUE)
```



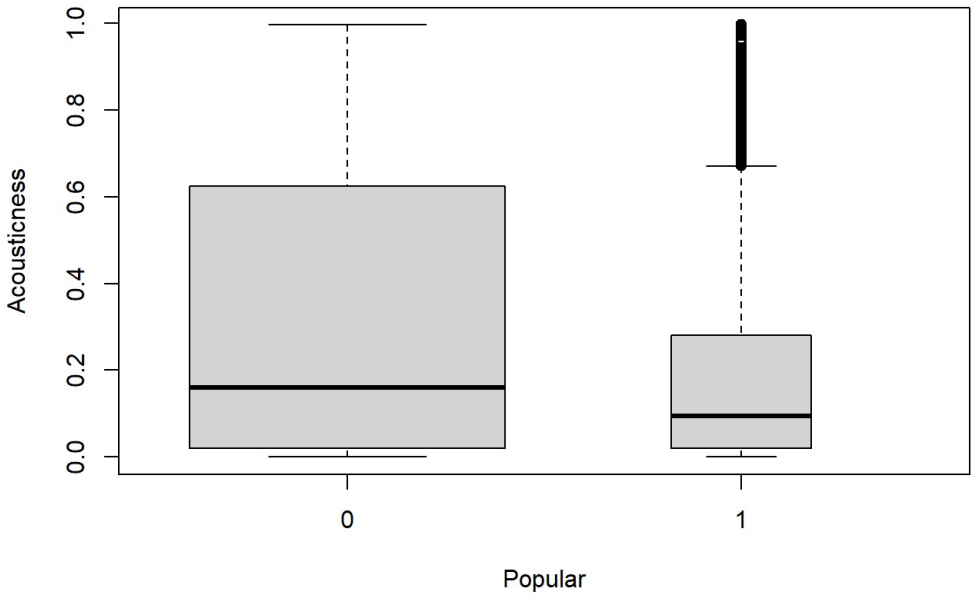

```
mosaic(table(train[,c(4,13)]), shade = TRUE, legend = TRUE)
```



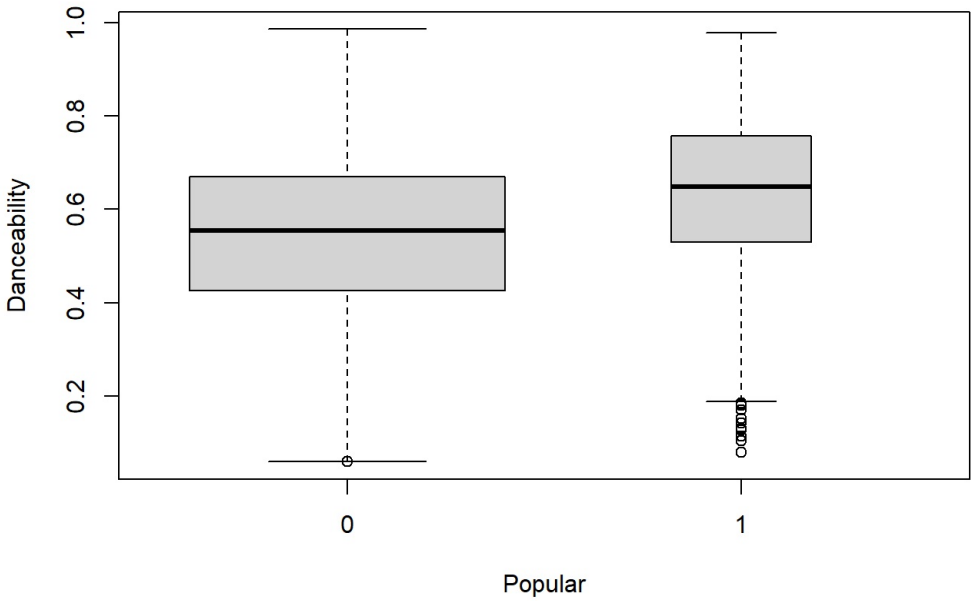

```
plot(train$popularity, train$energy, ylab = "Energy", xlab = "Popular", varwidth = TRUE)
```



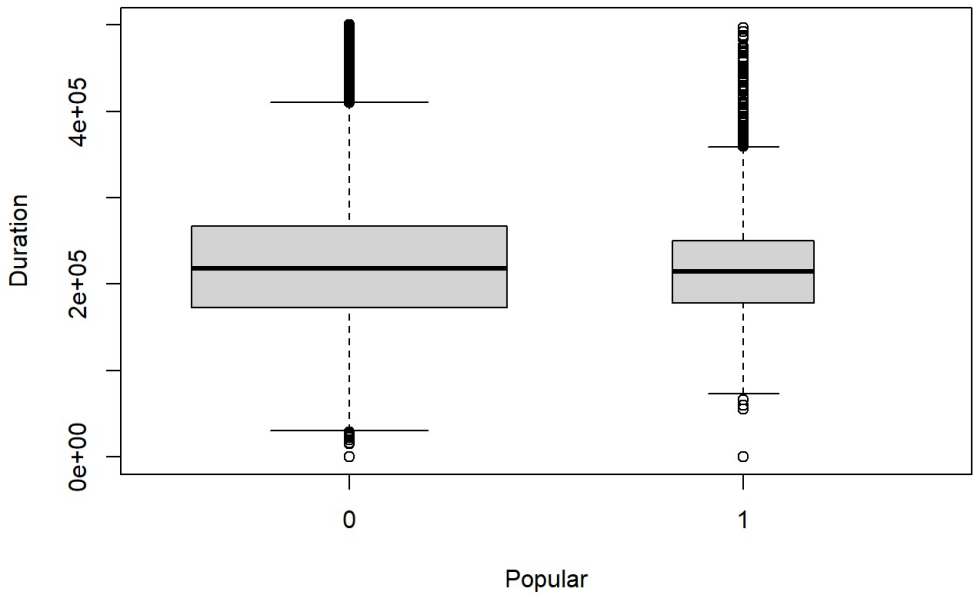

```
plot(train$popularity, train$acousticness, ylab = "Acousticness", xlab = "Popular", varwidth = TRUE)
```



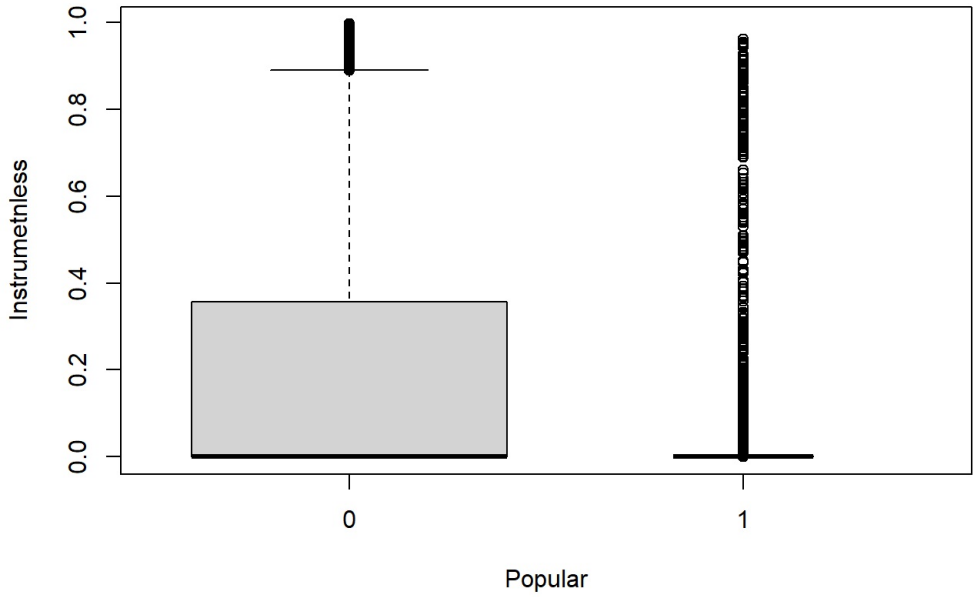

```
plot(train$popularity, train$danceability, ylab = "Danceability", xlab = "Popular", varwidth = TRUE)
```



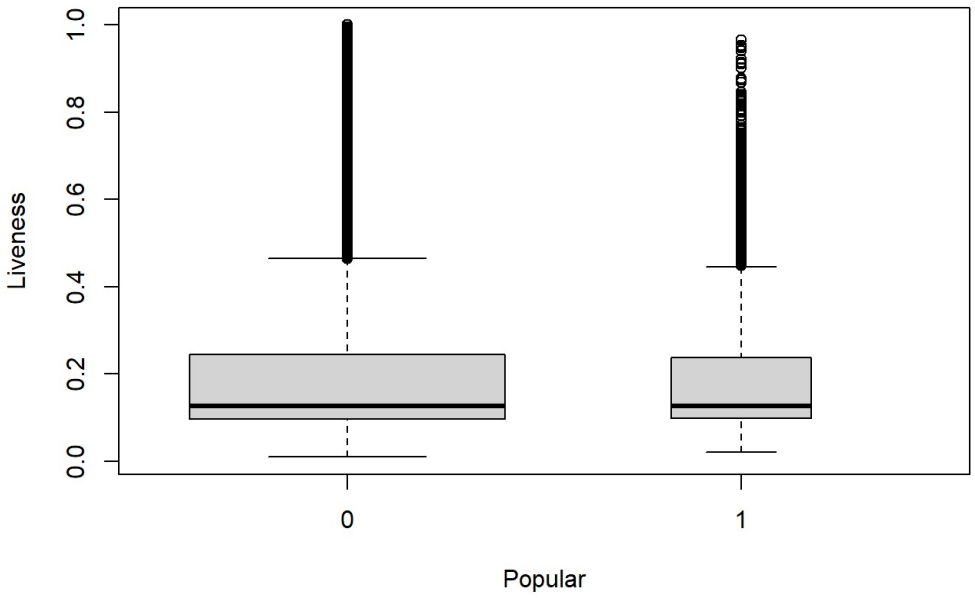

```
durationDF <- subset(train, duration_ms <= 500000)
plot(durationDF$popularity, durationDF$duration_ms, ylab = "Duration", xlab = "Popular", varwidth = TRUE)
```



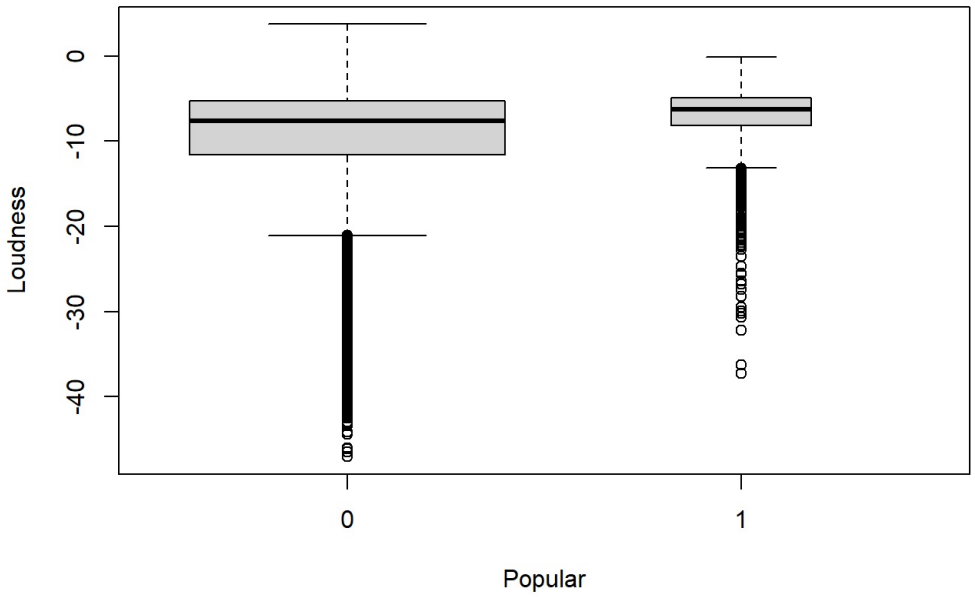

```
plot(train$popularity, train$instrumentalness, ylab = "Instrumetnless", xlab = "Popular", varwidth = TRUE)
```



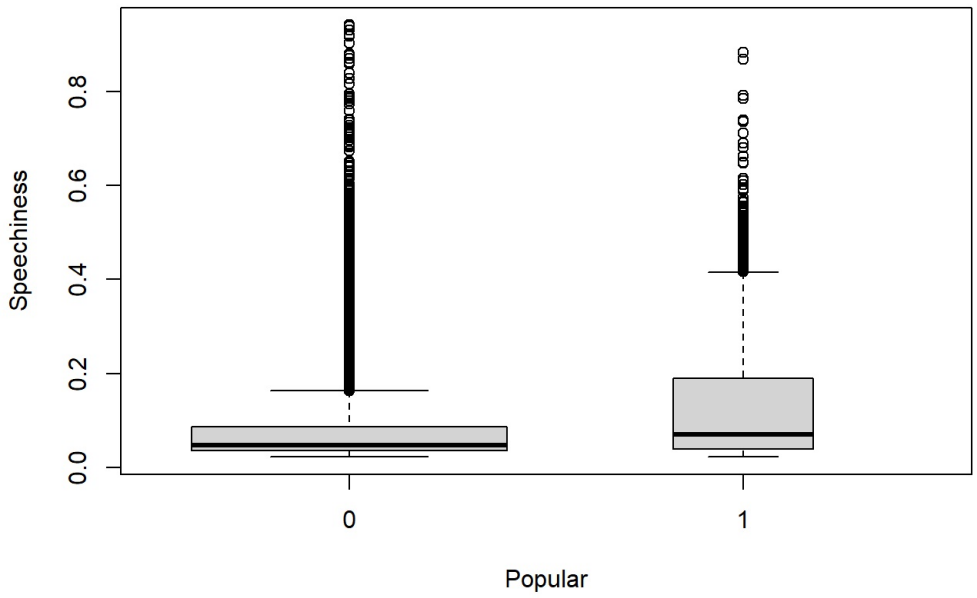

```
plot(train$popularity, train$liveness, ylab = "Liveness", xlab = "Popular", varwidth = TRUE)
```



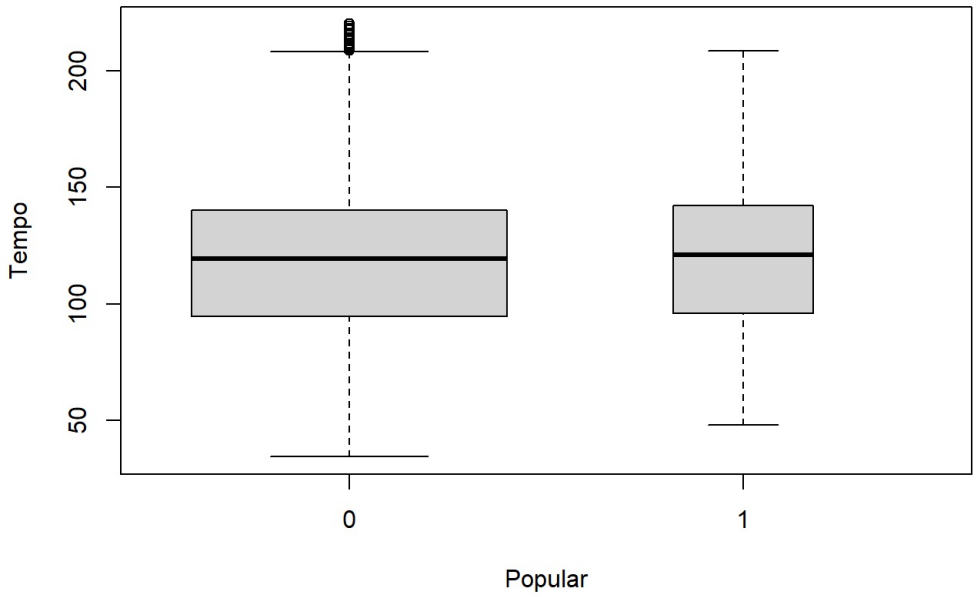

```
plot(train$popularity, train$loudness, ylab = "Loudness", xlab = "Popular", varwidth = TRUE)
```



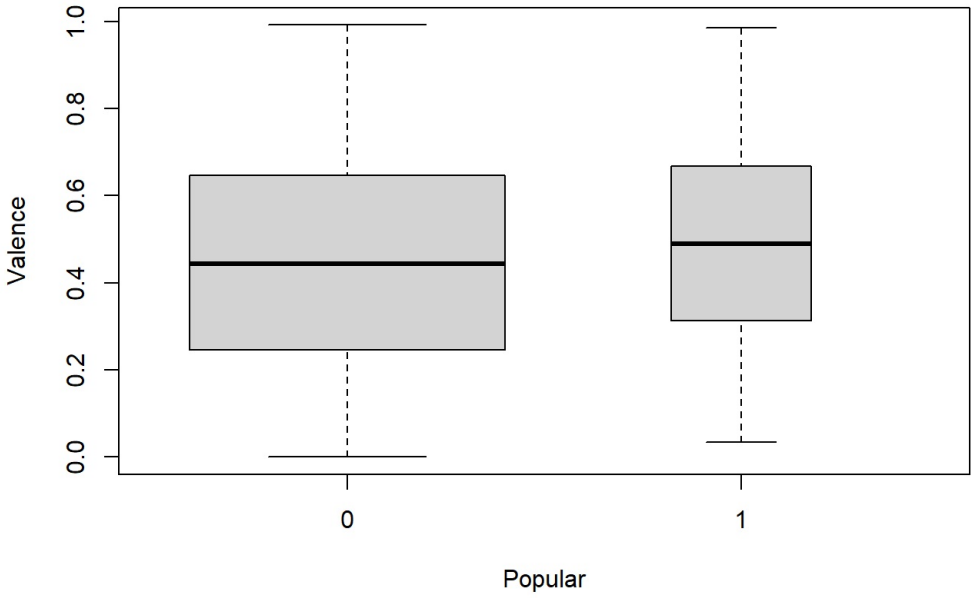

```
plot(train$popularity, train$speechiness, ylab = "Speechiness", xlab = "Popular", varwidth = TRUE)
```




```
plot(train$popularity, train$tempo, ylab = "Tempo", xlab = "Popular", varwidth = TRUE)
```




```
plot(train$popularity, train$valence, ylab = "Valence", xlab = "Popular", varwidth = TRUE)
```



Genre: It appears Rock, Rap, and Hip-Hop are the most popular. Alternate and country are the second most picked. The rest are about the same.

Key: Even though C# is pretty popular, it doesn't appear to have much influence popularity overall.

Mode: Has no influence.

Energy: It seem like higher energy very slightly increases popularity.

Acousticness: Lower acoustics have a small increase on popularity.

Danceability: Higher dancing shows a higher popularity.

Duration: Has no influence on popularity.

Instrumetnless: Instruments definitely make a song more popular.

Liveness: Has no influence.

Loudness: Louder has a small increase in popularity.

Speechiness: More speech has an increase on popularity.

Tempo: No effect.

Valence: No effect.

Big influence: Genre, Dance, Instrument less, Speech Small influence: Energy, Acoustics, Loudness

d: We will next build a logistic regression model using the factors that influenced the data the most.

```
glm1 <- glm(popularity~music_genre+danceability+instrumentalness+speechiness, data = train, family = binomial)
summary(glm1)
```

```
##
## Call:
## glm(formula = popularity ~ music_genre + danceability + instrumentalness +
##     speechiness, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.2972  -0.4620  -0.2198  -0.0545   3.6352
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -2.17732    0.08690 -25.055  < 2e-16 ***
## music_genreAnime      -4.59434    0.50320  -9.130  < 2e-16 ***
## music_genreBlues      -1.71564    0.12265 -13.989  < 2e-16 ***
## music_genreClassical  -2.39751    0.21949 -10.923  < 2e-16 ***
## music_genreCountry    -0.33823    0.07772  -4.352 1.35e-05 ***
## music_genreElectronic -1.46998    0.12470 -11.788  < 2e-16 ***
## music_genreHip-Hop     1.56705    0.06823  22.967  < 2e-16 ***
## music_genreJazz       -1.48633    0.12425 -11.963  < 2e-16 ***
## music_genreRap         2.00529    0.06646  30.174  < 2e-16 ***
## music_genreRock        1.83548    0.06170  29.748  < 2e-16 ***
## danceability           0.51722    0.12363   4.184 2.87e-05 ***
## instrumentalness      -1.00547    0.12989  -7.741 9.87e-15 ***
## speechiness           -0.94462    0.16387  -5.764 8.20e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 32259  on 36015  degrees of freedom
## Residual deviance: 22373  on 36003  degrees of freedom
## AIC: 22399
##
## Number of Fisher Scoring iterations: 9
```

Its good that the residual deviance is much lower than the null deviance. It would appear that each chosen factor was a good match since each got ***. The spread from min an max isn't half bad.

Many earlier assumptions proved correct. For example, Rap music has a logged 2 points increase in popularity, thus making it the most popular genre. Both lyric less and instrument less music have a -1 decline in popularity. Dacncibility didn't prove as influential as previously thought since there's only a .51 increase associated with it.

e: Now a Naive Bayes model will be made with the same data.

```
library(e1071)
nb1 <- naiveBayes(popularity~music_genre+danceability+instrumentalness+speechiness, data = train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.8350178 0.1649822
##
## Conditional probabilities:
##    music_genre
## Y   Alternative       Anime       Blues   Classical     Country  Electronic
##   0 0.106936224 0.118840194 0.117044623 0.117676398 0.107734256 0.115747822
##   1 0.071356446 0.000673174 0.013968361 0.004039044 0.056546617 0.014304948
##    music_genre
## Y       Hip-Hop        Jazz         Rap        Rock
##   0 0.073053136 0.117111126 0.059553102 0.066303119
##   1 0.237630427 0.014304948 0.302255133 0.284920902
##
##    danceability
## Y        [,1]      [,2]
##   0 0.5429642 0.1784203
##   1 0.6391185 0.1571334
##
##    instrumentalness
## Y         [,1]      [,2]
##   0 0.21165191 0.3439252
##   1 0.02770792 0.1236849
##
##    speechiness
## Y         [,1]       [,2]
##   0 0.08695412 0.09537119
##   1 0.12842068 0.12196696
```

As for the prior: The data showed that about .84 of the songs were unpopular and .16 was popular.

For genres: The popularity rate rock was .28, rap was .30, and hip-hop was .24. Meaning those 3 genres represented about .82 of all popular music, supporting the idea that they are the most popular.

For dance levels: The mean for popular songs was about .64 with an error of .17. This is notably better than the average unpopular songs dance level of .54 and with .17 error.

For Instrument less levels: The average unpopular song was about .21 (21%) without instruments with an a wide error of .34. Popular songs had only about .2 without instruments with a small error of .12. This supports the notion that music with instruments is more popular.

For speech level: It actually appears that speech didn't matter as much. The mean and errors are roughly the same, but still a bit higher with popular music.

f: Now for the fun part of comparing each model against the test data.

```
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, 2, 1)
acc1 <- mean(pred==as.integer(test$popularity))
print(paste("Logistic regression accuracy = ", acc1))
```

```
## [1] "Logistic regression accuracy =  0.833407374500222"
```

```
table(pred, as.integer(test$popularity))
```

```
##
## pred    1    2
##    1 7221 1211
##    2  289  283
```

```
pred2 <- predict(nb1, newdata = test, type = "class")
acc2 <-mean(pred2==test$popularity)
print(paste("Naive Bayes accuracy = ", acc2))
```

```
## [1] "Naive Bayes accuracy =  0.79775655264327"
```

```
table(pred2, test$popularity)
```

```
##
## pred2    0    1
##     0 6066  377
##     1 1444 1117
```

Logistic regression had the superior accuracy of .83. As for its errors, it seemed to underestimate how many songs were popular with 1211 false negative cases compared to 283 false positives.

Naive Bayes had less accuracy with about .80. Oddly enough, it had the opposite issue of overestimating a song popularity with 1444 false positives and only 377 false negatives.

Considering the data is very large, its natural that Naive Bayes doesn't perform as well as its specialized in smaller data sets. Logistic regression also performs well with liner data, as this is likely the case for the songs.

g: Logistic regression

Strengths: Works well for linearly separable classes and isn't very demanding computation wise.

Weaknesses: Its likely to under fit data that isn't liner or simple. Bad for more than 2 dimensions.

Naive Bayes

Strengths: Handles larger dimensions. It surprisingly easy to implement and interpret. Works well with small data sets.

Weaknesses: Will probably do worse that other classifiers if there is a large amount of data. Makes guesses on values in test sets even if they weren't in the training set. Has poor performance if the predictors aren't independent of each other.

h.