Final Project

Majorz

```
library(tidyverse)
library(tidymodels)
library(glmnet)
library(Stat2Data)
library(ggcorrplot)
spotify <- read_csv("data/tf_mini.csv")
```

```
spotify_mode <- spotify |>
  mutate(new_mode = if_else(mode == "major", 1, 0),
         new_mode = as.numeric(new_mode))

spotify_mode |> drop_na(new_mode)
```

```
# A tibble: 50,704 x 31
   track_id      durat~1 relea~2 us_po~3 acous~4 beat_~5 bounc~6 dance~7 dyn_r~8
   <chr>           <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
 1 t_a540e552-1~    110.    1950   100.    0.458   0.519   0.505   0.400    7.51
 2 t_67965da0-1~    188.    1950   100.    0.916   0.419   0.546   0.491    9.10
 3 t_0614ecd3-a~    161.    1951    99.6   0.813   0.426   0.508   0.492    8.37
 4 t_070a63a0-7~    175.    1951    99.7   0.397   0.401   0.360   0.552    5.97
 5 t_d6990e17-9~    370.    1951   100.    0.729   0.371   0.335   0.483    5.80
 6 t_fcb90952-0~    178.    1951   100.    0.186   0.549   0.579   0.744    8.67
 7 t_20675f8a-3~    166.    1952   100.    0.519   0.592   0.640   0.741    9.53
 8 t_7577ca53-5~    198.    1952    99.5   0.787   0.472   0.448   0.427    6.91
 9 t_8a461a4e-6~    215.    1954   100.    0.155   0.526   0.566   0.523    8.63
10 t_ae523005-8~    281.    1954    97.4   0.941   0.233   0.209   0.242    4.83
# ... with 50,694 more rows, 22 more variables: energy <dbl>, flatness <dbl>,
#   instrumentalness <dbl>, key <dbl>, liveness <dbl>, loudness <dbl>,
#   mechanism <dbl>, mode <chr>, organism <dbl>, speechiness <dbl>,
#   tempo <dbl>, time_signature <dbl>, valence <dbl>, acoustic_vector_0 <dbl>,
#   acoustic_vector_1 <dbl>, acoustic_vector_2 <dbl>, acoustic_vector_3 <dbl>,
#   acoustic_vector_4 <dbl>, acoustic_vector_5 <dbl>, acoustic_vector_6 <dbl>,
#   acoustic_vector_7 <dbl>, new_mode <dbl>, and abbreviated variable names ...
```

```r
glm_all_mode <- glm(new_mode ~ us_popularity_estimate + duration + release_year + acoustic
     beat_strength + bounciness + danceability + dyn_range_mean + energy +
     flatness + instrumentalness + key + liveness + loudness + mechanism +
       organism + speechiness + tempo + time_signature + valence,
     data = spotify_mode,
     family = "binomial")
summary(glm_all_mode)
```

```
Call:
glm(formula = new_mode ~ us_popularity_estimate + duration +
    release_year + acousticness + beat_strength + bounciness +
    danceability + dyn_range_mean + energy + flatness + instrumentalness +
    key + liveness + loudness + mechanism + organism + speechiness +
    tempo + time_signature + valence, family = "binomial", data = spotify_mode)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3569  -1.2543   0.7625   0.9493   1.8185

Coefficients:
                         Estimate Std. Error z value Pr(>|z|)
(Intercept)            32.2683808  2.3096693  13.971  < 2e-16 ***
us_popularity_estimate -0.0112941  0.0085642  -1.319 0.187249
duration               -0.0008868  0.0001370  -6.472 9.68e-11 ***
release_year           -0.0145826  0.0010562 -13.807  < 2e-16 ***
acousticness            0.4800550  0.1339125   3.585 0.000337 ***
beat_strength           2.3227249  0.3798220   6.115 9.64e-10 ***
bounciness             -4.2116774  0.5087117  -8.279  < 2e-16 ***
danceability            0.2508033  0.1611182   1.557 0.119556
dyn_range_mean          0.1188409  0.0200062   5.940 2.85e-09 ***
energy                 -0.5804580  0.1072094  -5.414 6.15e-08 ***
flatness                0.7082200  0.3348900   2.115 0.034448 *
instrumentalness       -0.3421403  0.0522757  -6.545 5.95e-11 ***
key                    -0.0930592  0.0026793 -34.733  < 2e-16 ***
liveness                0.3261005  0.0588139   5.545 2.95e-08 ***
loudness                0.0223914  0.0043966   5.093 3.53e-07 ***
```

```
mechanism              -0.8263282  0.2122943  -3.892 9.93e-05 ***
organism               -0.3927748  0.3168700  -1.240 0.215144    
speechiness            -1.0627013  0.0967583 -10.983  < 2e-16 ***
tempo                   0.0027563  0.0004504   6.120 9.37e-10 ***
time_signature         -0.2081995  0.0260103  -8.005 1.20e-15 ***
valence                 0.5394631  0.0506272  10.656  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 66141  on 50703  degrees of freedom
Residual deviance: 63327  on 50683  degrees of freedom
AIC: 63369

Number of Fisher Scoring iterations: 4
```

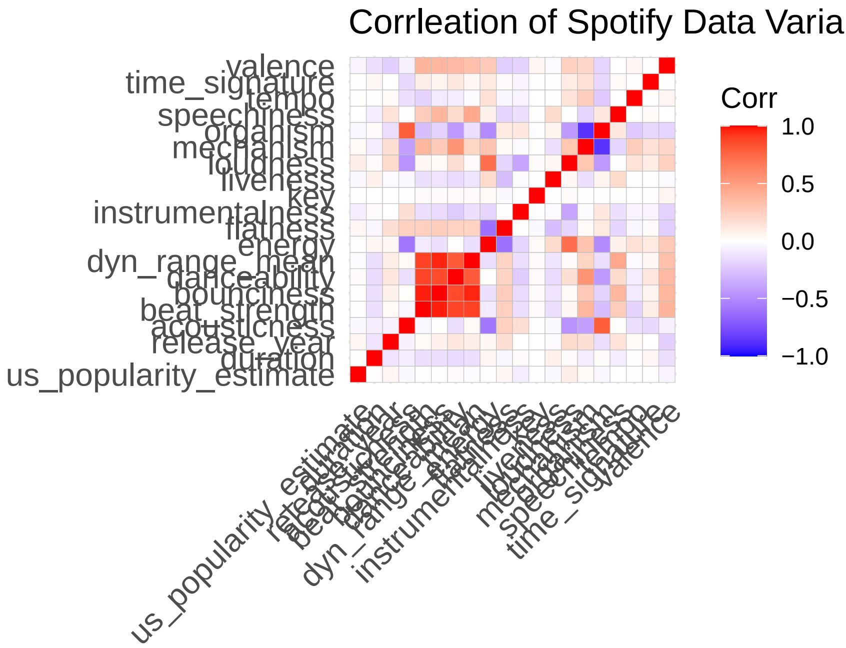
As demonstrated by the regression model above, there are many predictors that are statistically signficiant (significance level of 0.05). However, it is critical to improve this baseline model in the following ways:

1) Confrirm that there are not instances of multicollinearity (or model overfitting)
2) Ensure that the variables included are meaningfully contributing to the model
3) Optimize the model and determine if interacitons or changes are appropriate

```r
spotify_cor <- spotify_mode|>
  select(us_popularity_estimate, duration, release_year, acousticness,
     beat_strength, bounciness, danceability, dyn_range_mean, energy,
     flatness,instrumentalness, key, liveness, loudness, mechanism,
       organism, speechiness, tempo, time_signature, valence)

cor_spotify <- cor(spotify_cor)

ggcorrplot(cor_spotify)+
  labs(title = "Corrleation of Spotify Data Variables")
```



#Source http://www.sthda.com/english/wiki/ggcorrplot-visualization-of-a-correlation-matrix-using-ggplot2#:~:text=The%20easiest%20way%20to%20visualize,ggcorr()%20in%20ggally%20package

Examining the correlation plot above, it appears there are variables that have a high positive correlation with each other. This causes great concern with multicollinearity as the model may be overfitted. For example,

- beat_strength is highly correlated with
  - dyn_range_mean
  - danceability
  - bounciness

Therefore, to prevent overfitting in our regression model, the following variables should be removed:

1) beat_strength

2) dyn_range_mean

3) danceability

4) bounciness

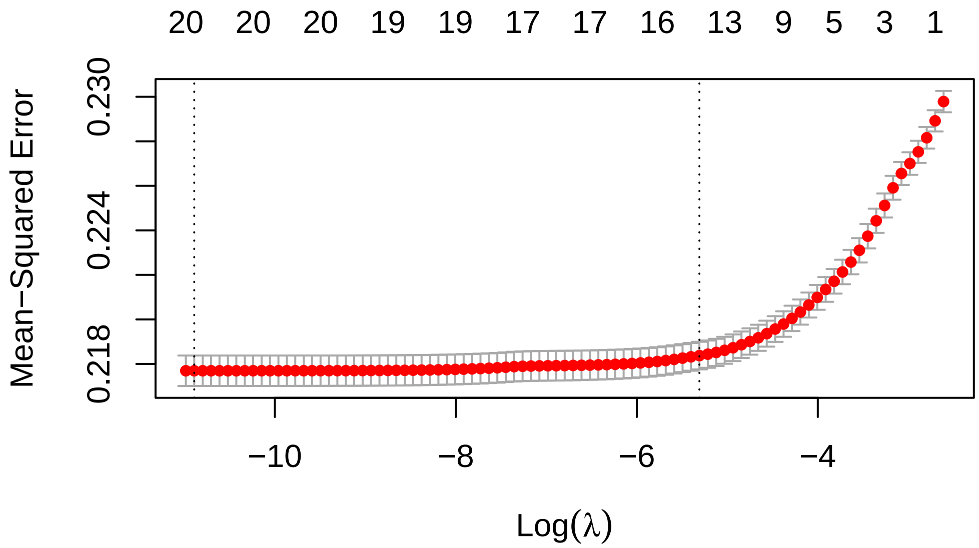
In addition to removing variables due to extremely high correlations, it is also important to select variables that make an impact on the model. For example, some variables may be replicated or not meaningful by nature to the outcome of interest; therefore, removal is essential. In this analysis, we decided to use a LASSO model to select variables that are essential to the model.

```
y <- spotify_mode$new_mode
x <- model.matrix(new_mode ~ us_popularity_estimate + duration + release_year +
                acousticness + beat_strength + bounciness + danceability +
                  dyn_range_mean + energy + flatness + instrumentalness + key +
                  liveness + loudness + mechanism + organism + speechiness +
                  tempo + time_signature + valence,
                  data = spotify_mode, family = "binomial")
lasso_sc <- cv.glmnet(x, y, alpha = 1)
best_lambda <- lasso_sc$lambda.min
lasso_final <- glmnet(x, y, alpha = 1, lambda = best_lambda)
lasso_final$beta
```

```
21 x 1 sparse Matrix of class "dgCMatrix"
                                  s0
(Intercept)             .
us_popularity_estimate -0.0023679722
duration               -0.0001815326
release_year           -0.0027552856
acousticness            0.0812649420
beat_strength           0.4395871374
bounciness             -0.8228417626
danceability            0.0540884815
dyn_range_mean          0.0228099401
energy                 -0.1261583504
flatness                0.1277800576
instrumentalness       -0.0765283096
key                    -0.0204720118
liveness                0.0689359419
loudness                0.0046767651
mechanism              -0.1463426787
organism               -0.0386700035
speechiness            -0.2428939108
tempo                   0.0005691694
time_signature         -0.0409648214
valence                 0.1198943315
```

LASSO kept all of the predictors.

```r
plot(lasso_sc)
```



not sure if this is needed or not^

Introduction and Data

Methodology

Evaluating assumptions:

figure out to make this smaller or how to get charts to show^

There had to be less data points for some of the predictors because there was only so many different values and enough of them to be able to get the empirical logits. For example, with key there is only 12 unique values, but not all of them had enough values to be calculated, so we did 10 groups. I eliminated the titles to make the plots more clear and because they were repetitive. In summary, we concluded that linearity is met for __________ because there is no major pattern in empirical logits. Linearity was not met for ______ because ______.

```r
glm_aug <- glm_aug |>
  mutate(prob = exp(.fitted)/(1 + exp(.fitted)),
         pred_mode = ifelse(prob > 0.5, "Major", "Minor")) |>
  select(.fitted, prob, pred_mode, new_mode)

table(glm_aug$pred_mode, glm_aug$new_mode)
```

Using our logistic regression model as a classifier for any infection by using a threshold of 0.5 predicted probability, we are able to calculate the following values:

Prevalence:

Sensitivity:

Specificity:

Positive predicted value:

Negative predicted value:

This implies that _____

```r
glm_aug |>
  roc_auc(truth = as.factor(new_mode),
          prob,
          event_level = "second") |>
  autoplot()

glm_aug |>
  roc_auc(truth = as.factor(new_mode),
          prob,
          event_level = "second")
```

Results

HOW to pick which predictors are the best???

One predictor that makes sense to interpret is key because key has changes in whole numbers while many of the other predictors are within tenths of differences of each other amongst observations. Holding all other predictors constant, for every one (unit) increase in key, we expect the log-odds of a song being major rather than minor to increase by approximately 0.0931. So, when holding all other predictors constant, we for every one number increase in key (find what this means), the odds of the patient getting any infection is predicted to be multiplied by $e^{0.0931} = 1.0976$. For an example, while holding all other predictors constant, the relative odds of a song being major rather than minor comparing a song with key 10 vs a song with key 2 is $e^{8*0.0931}$ is 2.106.

to be continued

Discussion