Stat 21 Final Project - Final Draft

Rodas Jateno, Sannan Dhillon, Amy Cho

12/17/2021

Data Cleaning

Before the entire process of data cleaning starts, we are first going to download the dataset and put it in a variable. The dataset in question is pubicly availbe from a link so we will read the CSV off the link directly.

```
billboard <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   url = col_character(),
##   week_id = col_character(),
##   week_position = col_double(),
##   song = col_character(),
##   performer = col_character(),
##   song_id = col_character(),
##   instance = col_double(),
##   previous_week_position = col_double(),
##   peak_position = col_double(),
##   weeks_on_chart = col_double()
## )
```

Another data we were looking at was the audio features data which is a subset of the billboard data that includes all the songs that appear on Spotify along with their spotify features. We decided to take out songs that have NA because we couldn't find data information for these songs.

```
audio_features = read.csv("audio_features.csv")
# remove NAs
audio_features_final <- na.omit(audio_features)

#select only important variables in our billboard data
b1 = billboard %>% mutate(date=mdy(week_id), year=year(date))

#order b1 by least recent to most recent
b1_ordered <- b1 %>% arrange(ymd(b1$date))
```

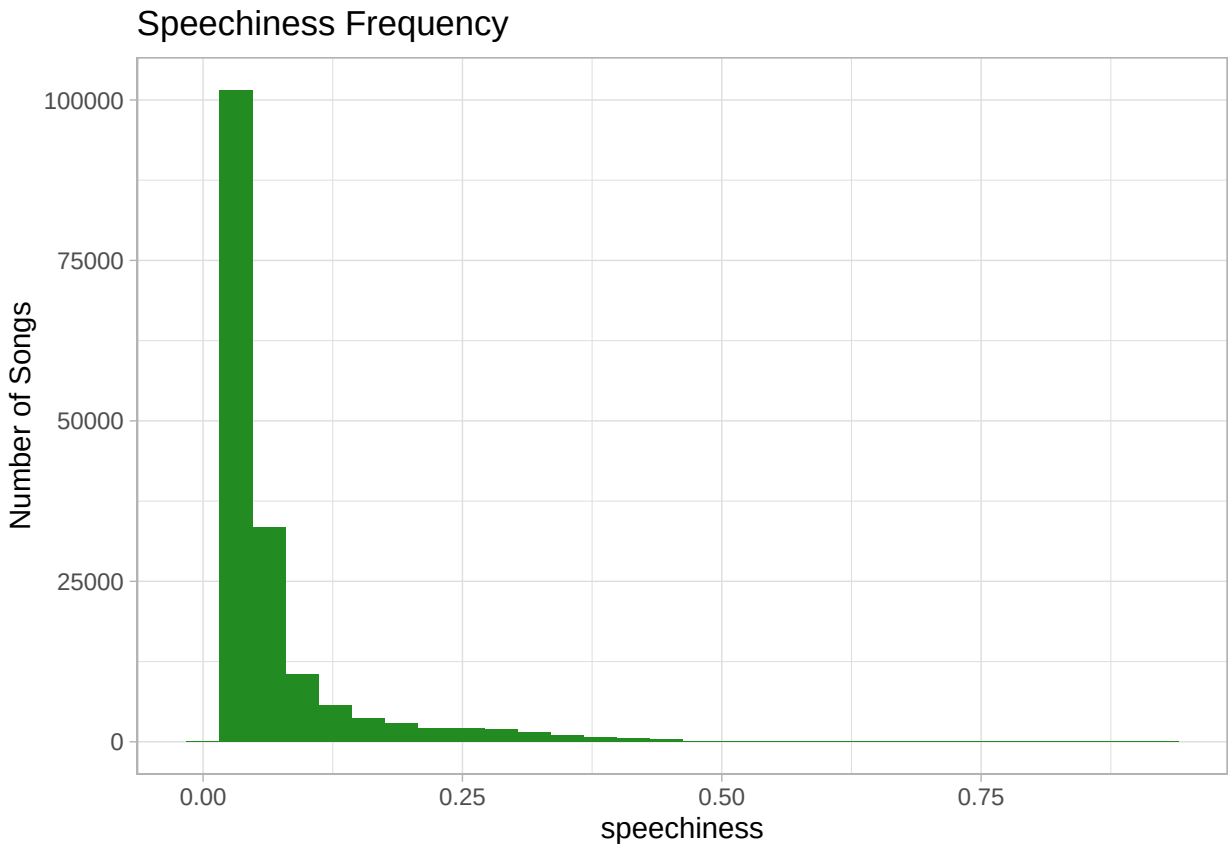
After we do that we have to find a way to merge the new dataset we made with the original billboard dataset because that dataset contains variables that we will need for our model.

```
#select only important variables in our audio_features data

f1 = audio_features_final %>% select(spotify_genre, danceability, energy, key, loudness, mode, speechine

#join the billboard_ordered and audio_features dataset
joined = b1_ordered %>% right_join(f1, by="song_id")

#added in_billboard = 1 because all songs appeared in billboard
final_data <- mutate(joined, in_billboard = 1)

#final_data shows a clean datatset that only includes important predictors
```

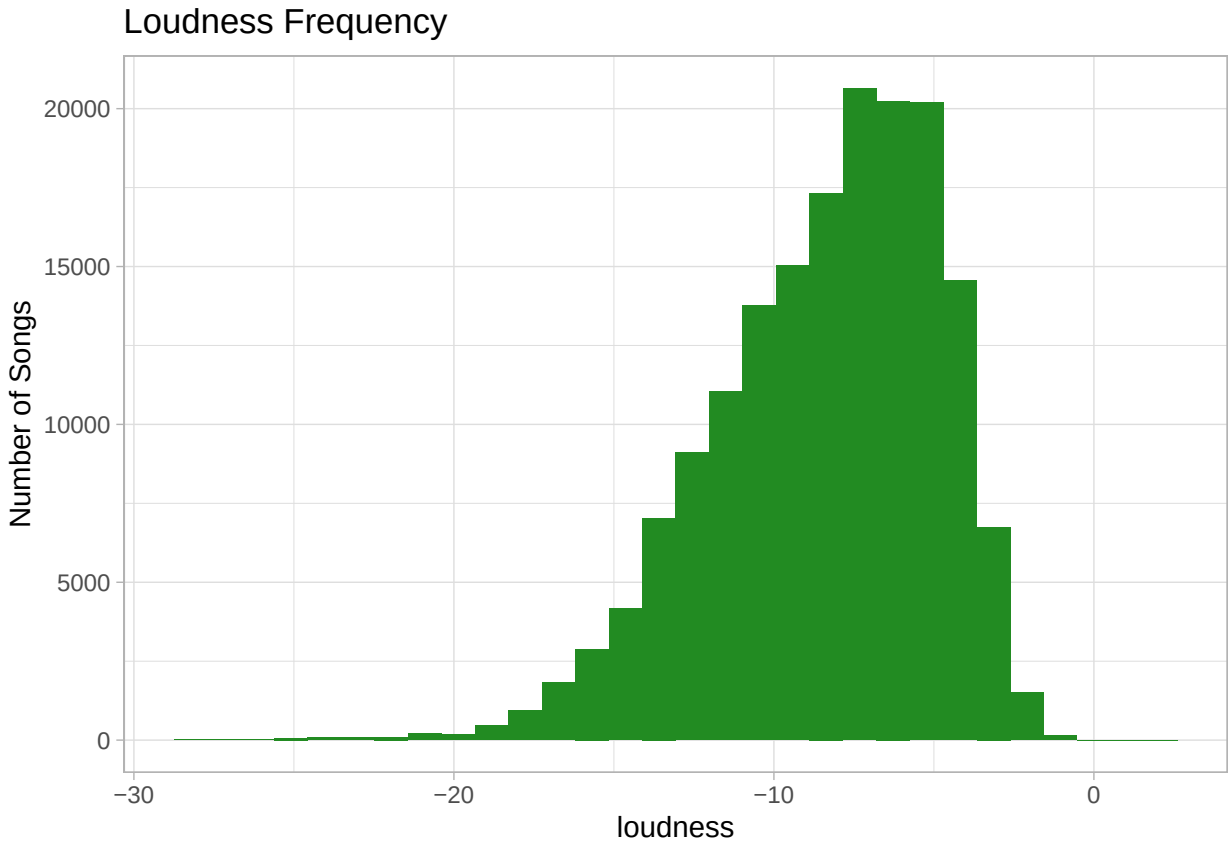
Exploring the Data

After we have accomplished the merge, we can now look at interesting EDAs that come up.

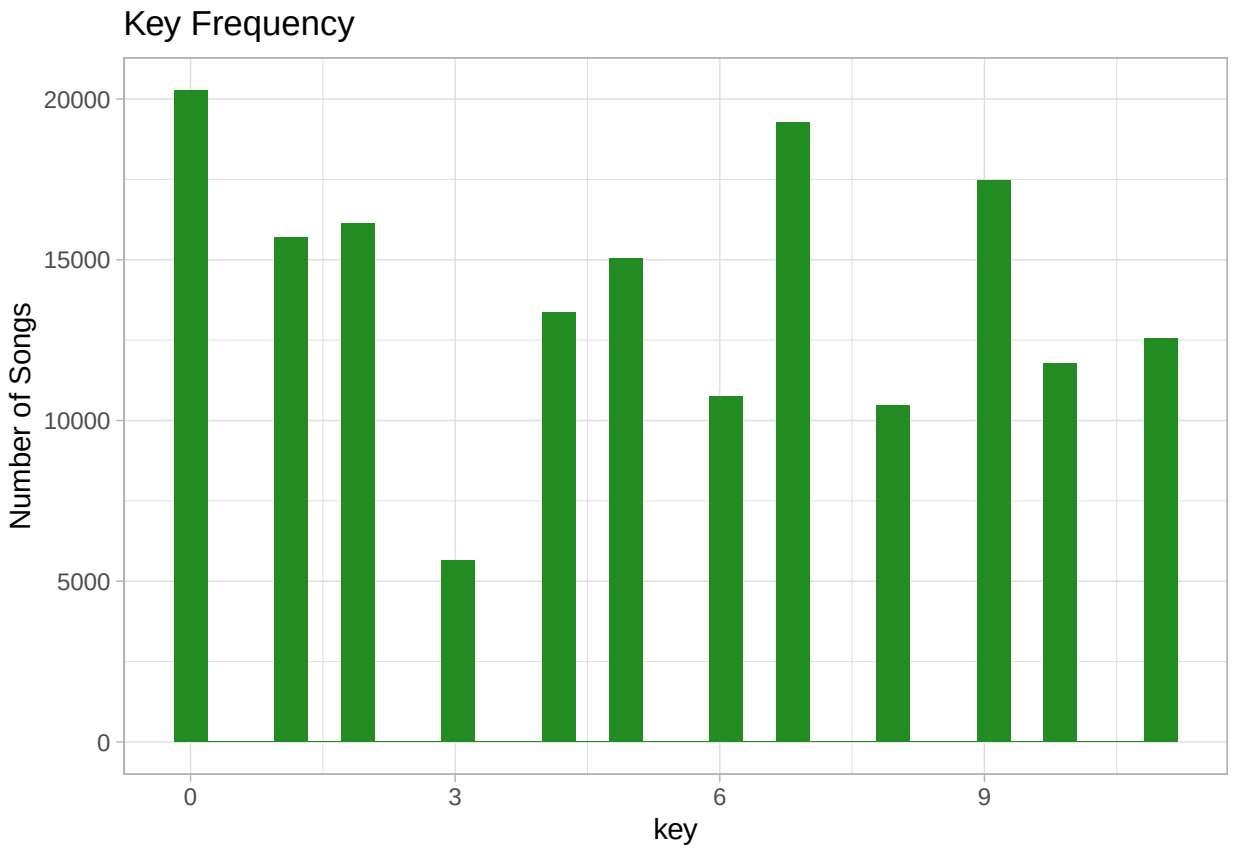
```
ggplot(final_data) +
  aes(x = speechiness) +
  geom_histogram(bins = 30L, fill = "#228B22") +
  labs(y = "Number of Songs", title = "Speechiness Frequency") +
  theme_light()
```



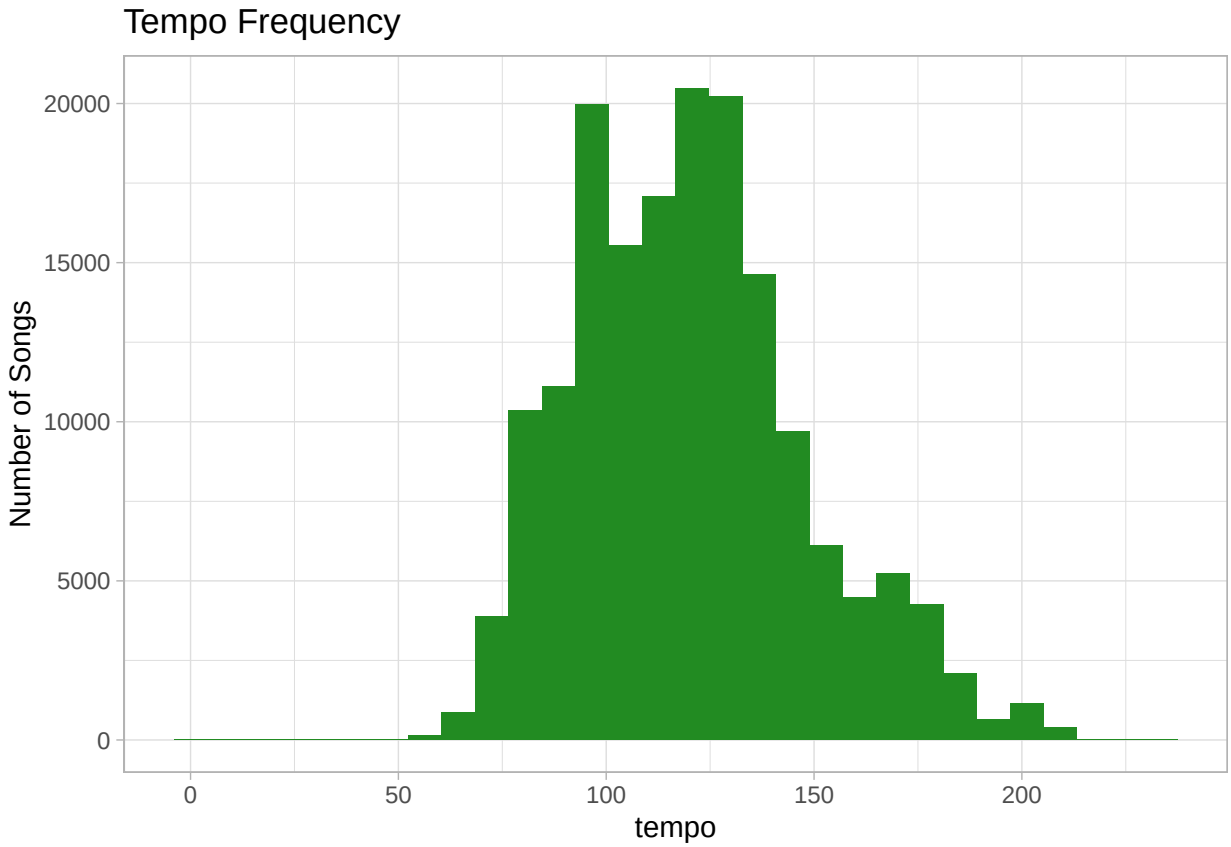
```
ggplot(final_data) +
  aes(x = loudness) +
  geom_histogram(bins = 30L, fill = "#228B22") +
  labs(y = "Number of Songs", title = "Loudness Frequency") +
  theme_light()
```




```
ggplot(final_data) +
  aes(x = key) +
  geom_histogram(bins = 30L, fill = "#228B22") +
  labs(y = "Number of Songs", title = "Key Frequency") +
  theme_light()
```

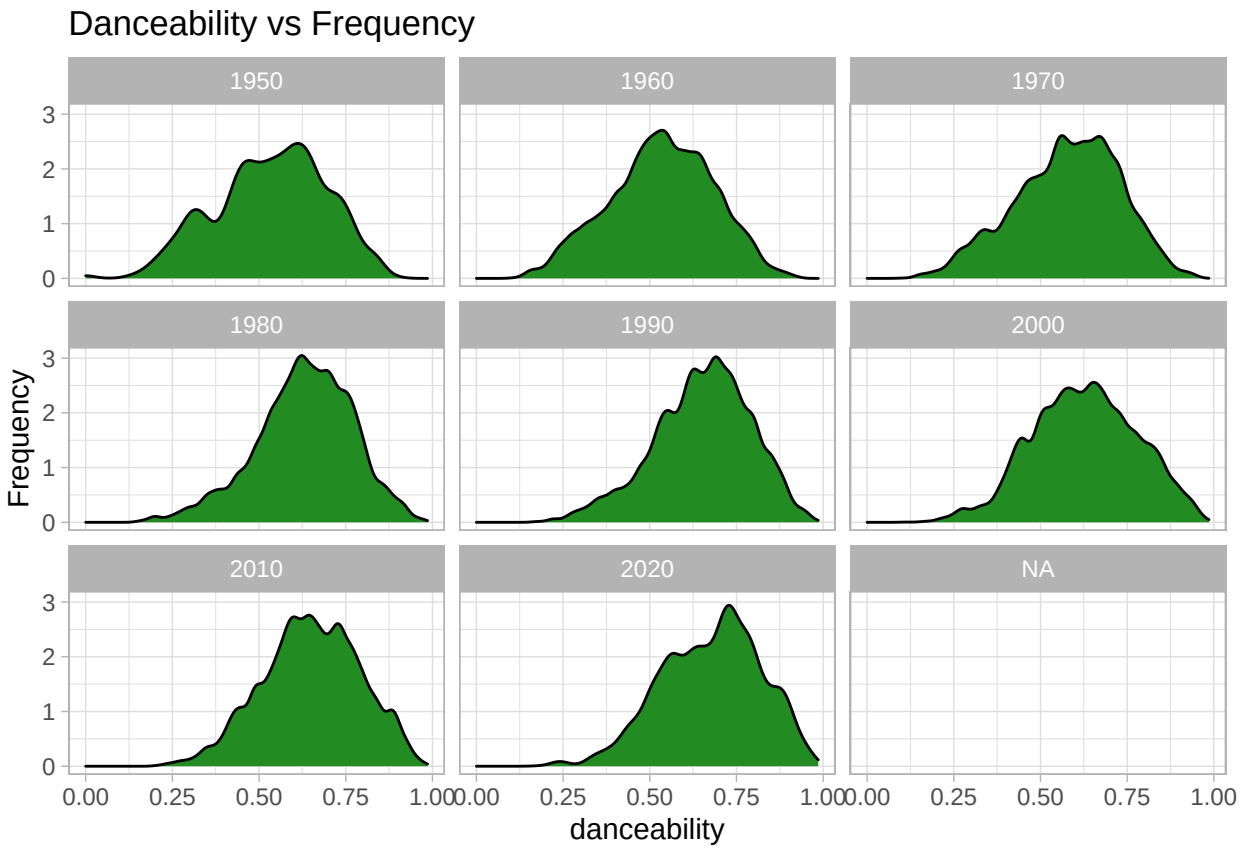



```
ggplot(final_data) +
  aes(x = tempo) +
  geom_histogram(bins = 30L, fill = "#228B22") +
  labs(y = "Number of Songs", title = "Tempo Frequency") +
  theme_light()
```

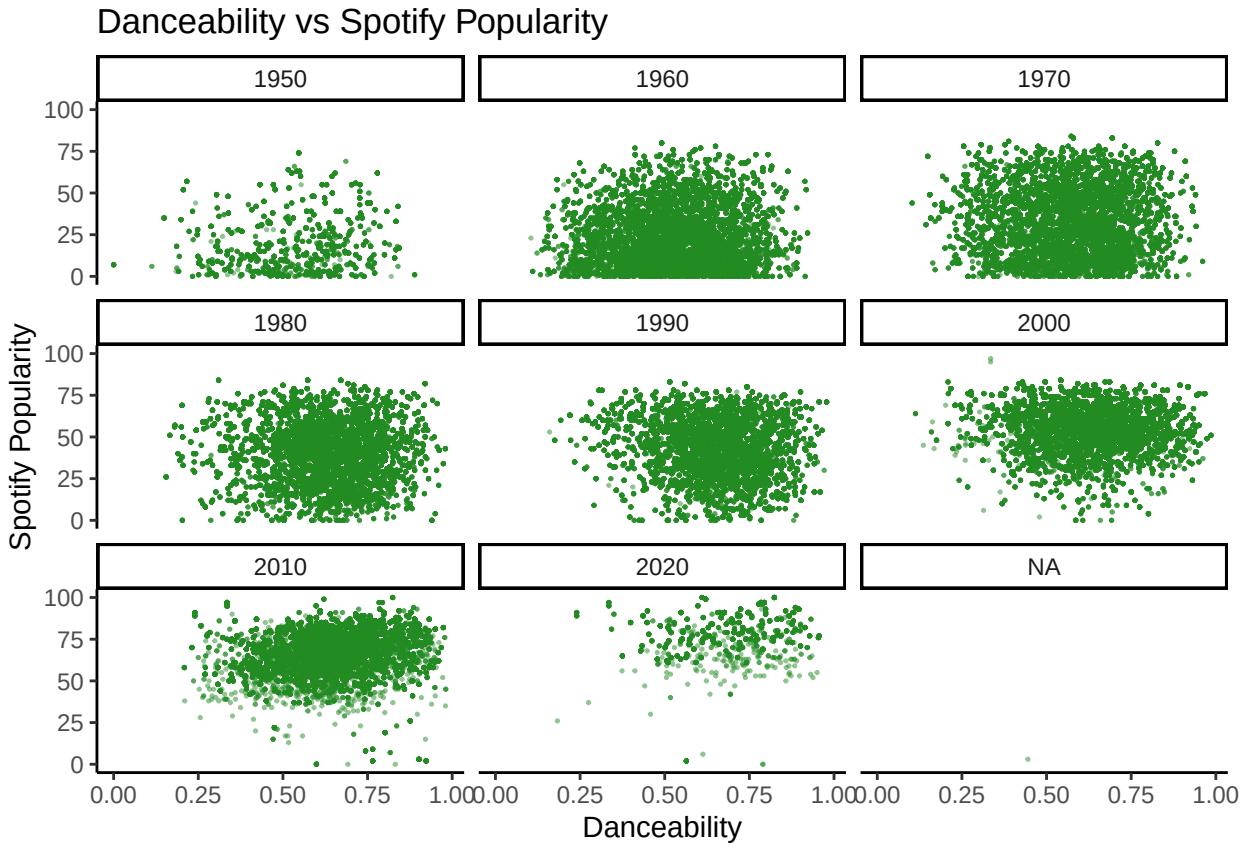



```
final_data = final_data %>%
  mutate(year = format(date,"%Y")) %>%
  mutate(decade = floor((strtoi(year))/10)*10) %>%
      group_by(decade)

ggplot(final_data) +
  aes(x = danceability) +
  geom_density(adjust = 1L, fill = "#228B22") +
  labs(y = "Frequency", title = "Danceability vs Frequency") +
  theme_light() +
  facet_wrap(vars(decade))
```

```
## Warning: Groups with fewer than two data points have been dropped.

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```




```
ggplot(final_data) +
  aes(x = danceability, y = spotify_track_popularity) +
  geom_point(shape = "circle", alpha = 0.5, size = 0.25, colour = "#228B22") +
  labs(
    x = "Danceability",
    y = "Spotify Popularity",
    title = "Danceability vs Spotify Popularity"
  ) + facet_wrap(vars(decade)) +
  theme_classic()
```




```
billboard_songs = read.csv("final_data.csv")
random_songs = read.csv("random_songs_playlist5001.csv")
random_songs = subset(random_songs, select = -c(spotify_track_explicit))
random_songs = random_songs %>%
  mutate(song_id = paste(track_name, artist, sep=""))
random_songs = random_songs %>%
  filter(!song_id %in% billboard_songs$song_id)
random_songs = random_songs %>%
  rename(spotify_track_explicit = explicit) %>%
  mutate(in_billboard = 0)
billboard_songs = billboard_songs %>%
  select(c(4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20))
random_songs = random_songs %>%
  select(c(22, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 19, 21, 23))
```

```
colnames(random_songs)
```

```
##  [1] "song_id"                 "danceability"
##  [3] "energy"                  "key"
##  [5] "loudness"                "mode"
##  [7] "speechiness"             "acousticness"
##  [9] "instrumentalness"        "liveness"
## [11] "valence"                 "tempo"
## [13] "time_signature"          "spotify_track_popularity"
## [15] "spotify_track_explicit"  "in_billboard"
```

```r
colnames(billboard_songs)
```

```
##  [1] "song_id"                  "danceability"
##  [3] "energy"                   "key"
##  [5] "loudness"                 "mode"
##  [7] "speechiness"              "acousticness"
##  [9] "instrumentalness"         "liveness"
## [11] "valence"                  "tempo"
## [13] "time_signature"           "spotify_track_popularity"
## [15] "spotify_track_explicit"   "in_billboard"
```

```r
all_songs = rbind(random_songs, billboard_songs)
all_songs = unique(all_songs)
```

Modelling and Selection

Now that we have seen all the interesting trends in the data we can move on to performing the predictor

```r
#using forward backward selection

#base model

base_lm = lm(formula = in_billboard ~ 1, data = all_songs)

#full model

full_lm = lm(formula = in_billboard ~ danceability+energy+key+loudness+mode+speechiness +acousticness+i

#forward selection
forward_lm = stats::step(base_lm, scope=list(lower=base_lm, upper=full_lm), direction="forward")
```

```
## Start:  AIC=-32027.52
## in_billboard ~ 1
##
##                            Df Sum of Sq    RSS      AIC
## + spotify_track_explicit    3    3615.8    0.0 -1154697
## + loudness                  1     403.9 3211.9   -34298
## + acousticness              1     190.1 3425.7   -33062
## + danceability              1     125.9 3489.9   -32706
## + energy                    1     120.7 3495.1   -32677
## + speechiness               1     106.0 3509.8   -32596
## + mode                      1     104.9 3510.9   -32590
## + valence                   1      97.9 3517.9   -32552
## + time_signature            1      17.7 3598.1   -32120
## + spotify_track_popularity  1      16.7 3599.1   -32114
## + instrumentalness          1       3.6 3612.2   -32045
## + tempo                     1       3.3 3612.5   -32043
## <none>                                  3615.8   -32028
```

```
## + liveness                  1       0.3 3615.5   -32027
## + key                       1       0.0 3615.8   -32026
##
## Step:  AIC=-1154697
## in_billboard ~ spotify_track_explicit
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                            Df  Sum of Sq        RSS      AIC
## + liveness                  1 2.5476e-26 1.4143e-22 -1154698
## + tempo                     1 1.7672e-26 1.4144e-22 -1154697
## <none>                                   1.4145e-22 -1154697
## + danceability              1 5.5387e-27 1.4145e-22 -1154696
## + loudness                  1 4.2330e-27 1.4145e-22 -1154695
## + acousticness              1 3.4975e-27 1.4145e-22 -1154695
## + mode                      1 2.8490e-27 1.4145e-22 -1154695
## + valence                   1 1.8287e-27 1.4145e-22 -1154695
## + spotify_track_popularity  1 1.4575e-27 1.4145e-22 -1154695
## + energy                    1 1.0830e-27 1.4145e-22 -1154695
## + speechiness               1 3.7210e-28 1.4145e-22 -1154695
## + key                       1 2.7850e-28 1.4145e-22 -1154695
## + instrumentalness          1 3.6800e-29 1.4145e-22 -1154695
## + time_signature            1 1.1000e-30 1.4145e-22 -1154695
##
## Step:  AIC=-1154698
## in_billboard ~ spotify_track_explicit + liveness
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                            Df  Sum of Sq        RSS      AIC
## + tempo                     1 1.7085e-26 1.4141e-22 -1154699
## <none>                                   1.4143e-22 -1154698
## + danceability              1 9.2048e-27 1.4142e-22 -1154698
## + acousticness              1 3.9869e-27 1.4142e-22 -1154697
## + loudness                  1 3.4858e-27 1.4142e-22 -1154697
## + mode                      1 2.8780e-27 1.4142e-22 -1154697
## + valence                   1 2.1499e-27 1.4143e-22 -1154697
## + spotify_track_popularity  1 6.8560e-28 1.4143e-22 -1154696
## + key                       1 3.2300e-28 1.4143e-22 -1154696
## + energy                    1 2.2070e-28 1.4143e-22 -1154696
## + speechiness               1 4.4300e-29 1.4143e-22 -1154696
## + instrumentalness          1 1.6100e-29 1.4143e-22 -1154696
## + time_signature            1 1.0400e-29 1.4143e-22 -1154696
##
## Step:  AIC=-1154699
## in_billboard ~ spotify_track_explicit + liveness + tempo
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                            Df  Sum of Sq        RSS      AIC
## <none>                                   1.4141e-22 -1154699
## + danceability              1 1.3735e-26 1.4140e-22 -1154698
```

```
## + valence                   1 3.1030e-27 1.4141e-22 -1154697
## + mode                      1 3.0499e-27 1.4141e-22 -1154697
## + acousticness              1 2.4899e-27 1.4141e-22 -1154697
## + loudness                  1 2.3207e-27 1.4141e-22 -1154697
## + spotify_track_popularity  1 6.9910e-28 1.4141e-22 -1154697
## + key                       1 3.0530e-28 1.4141e-22 -1154697
## + instrumentalness          1 3.8600e-29 1.4141e-22 -1154697
## + energy                    1 3.1200e-29 1.4141e-22 -1154697
## + time_signature            1 1.7100e-29 1.4141e-22 -1154697
## + speechiness               1 1.3200e-29 1.4141e-22 -1154697
```

```
#backward selection
backward_lm = stats::step(full_lm, scope=list(lower=base_lm, upper=full_lm), direction="backward")
```

```
## Start:  AIC=-1172691
## in_billboard ~ danceability + energy + key + loudness + mode +
##     speechiness + acousticness + instrumentalness + liveness +
##     valence + tempo + time_signature + spotify_track_popularity +
##     spotify_track_explicit
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                            Df Sum of Sq    RSS      AIC
## - acousticness              1       0.0    0.0 -1172732
## - energy                    1       0.0    0.0 -1172723
## - valence                   1       0.0    0.0 -1172711
## - loudness                  1       0.0    0.0 -1172703
## <none>                                     0.0 -1172691
## - liveness                  1       0.0    0.0 -1172691
## - danceability              1       0.0    0.0 -1172685
## - tempo                     1       0.0    0.0 -1172675
## - mode                      1       0.0    0.0 -1172672
## - spotify_track_popularity  1       0.0    0.0 -1172668
## - time_signature            1       0.0    0.0 -1172664
## - instrumentalness          1       0.0    0.0 -1172655
## - key                       1       0.0    0.0 -1172632
## - speechiness               1       0.0    0.0 -1172609
## - spotify_track_explicit    3    2731.7 2731.7   -37382
##
## Step:  AIC=-1172724
## in_billboard ~ danceability + energy + key + loudness + mode +
##     speechiness + instrumentalness + liveness + valence + tempo +
##     time_signature + spotify_track_popularity + spotify_track_explicit
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                            Df Sum of Sq    RSS      AIC
## - energy                    1       0.0    0.0 -1172745
## - time_signature            1       0.0    0.0 -1172734
## - spotify_track_popularity  1       0.0    0.0 -1172725
## <none>                                     0.0 -1172724
## - instrumentalness          1       0.0    0.0 -1172724
```

```
## - loudness                  1       0.0    0.0 -1172722
## - liveness                  1       0.0    0.0 -1172719
## - speechiness               1       0.0    0.0 -1172717
## - tempo                     1       0.0    0.0 -1172717
## - danceability              1       0.0    0.0 -1172687
## - key                       1       0.0    0.0 -1172637
## - valence                   1       0.0    0.0 -1172620
## - mode                      1       0.0    0.0 -1172611
## - spotify_track_explicit    3    2745.8 2745.8   -37285
##
## Step:  AIC=-1172738
## in_billboard ~ danceability + key + loudness + mode + speechiness +
##     instrumentalness + liveness + valence + tempo + time_signature +
##     spotify_track_popularity + spotify_track_explicit
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                            Df Sum of Sq    RSS      AIC
## - speechiness               1       0.0    0.0 -1172740
## - tempo                     1       0.0    0.0 -1172739
## <none>                                     0.0 -1172738
## - key                       1       0.0    0.0 -1172729
## - spotify_track_popularity  1       0.0    0.0 -1172722
## - instrumentalness          1       0.0    0.0 -1172711
## - danceability              1       0.0    0.0 -1172699
## - time_signature            1       0.0    0.0 -1172699
## - mode                      1       0.0    0.0 -1172668
## - liveness                  1       0.0    0.0 -1172658
## - valence                   1       0.0    0.0 -1172608
## - loudness                  1       0.0    0.0 -1172594
## - spotify_track_explicit    3    2745.8 2745.8   -37287
##
## Step:  AIC=-1172732
## in_billboard ~ danceability + key + loudness + mode + instrumentalness +
##     liveness + valence + tempo + time_signature + spotify_track_popularity +
##     spotify_track_explicit
```

```
#forward-backward selection
both_lm = stats::step(base_lm, scope=list(lower=base_lm, upper=full_lm), direction="both")
```

```
## Start:  AIC=-32027.52
## in_billboard ~ 1
##
##                            Df Sum of Sq    RSS      AIC
## + spotify_track_explicit    3    3615.8    0.0 -1154697
## + loudness                  1     403.9 3211.9   -34298
## + acousticness              1     190.1 3425.7   -33062
## + danceability              1     125.9 3489.9   -32706
## + energy                    1     120.7 3495.1   -32677
## + speechiness               1     106.0 3509.8   -32596
## + mode                      1     104.9 3510.9   -32590
## + valence                   1      97.9 3517.9   -32552
## + time_signature            1      17.7 3598.1   -32120
```

```
## + spotify_track_popularity  1      16.7 3599.1   -32114
## + instrumentalness          1       3.6 3612.2   -32045
## + tempo                     1       3.3 3612.5   -32043
## <none>                                  3615.8   -32028
## + liveness                  1       0.3 3615.5   -32027
## + key                       1       0.0 3615.8   -32026
##
## Step:  AIC=-1154697
## in_billboard ~ spotify_track_explicit
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                             Df Sum of Sq    RSS      AIC
## + liveness                   1       0.0    0.0 -1154698
## + tempo                      1       0.0    0.0 -1154697
## <none>                                      0.0 -1154697
## + danceability               1       0.0    0.0 -1154696
## + loudness                   1       0.0    0.0 -1154695
## + acousticness               1       0.0    0.0 -1154695
## + mode                       1       0.0    0.0 -1154695
## + valence                    1       0.0    0.0 -1154695
## + spotify_track_popularity   1       0.0    0.0 -1154695
## + energy                     1       0.0    0.0 -1154695
## + speechiness                1       0.0    0.0 -1154695
## + key                        1       0.0    0.0 -1154695
## + instrumentalness           1       0.0    0.0 -1154695
## + time_signature             1       0.0    0.0 -1154695
## - spotify_track_explicit     3    3615.8 3615.8   -32028
##
## Step:  AIC=-1154698
## in_billboard ~ spotify_track_explicit + liveness
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
## Warning: attempting model selection on an essentially perfect fit is nonsense
```

```
##                             Df Sum of Sq    RSS      AIC
## - liveness                   1       0.0    0.0 -1154704
## + tempo                      1       0.0    0.0 -1154699
## <none>                                      0.0 -1154698
## + danceability               1       0.0    0.0 -1154698
## + acousticness               1       0.0    0.0 -1154697
## + loudness                   1       0.0    0.0 -1154697
## + mode                       1       0.0    0.0 -1154697
## + valence                    1       0.0    0.0 -1154697
## + spotify_track_popularity   1       0.0    0.0 -1154696
## + key                        1       0.0    0.0 -1154696
## + energy                     1       0.0    0.0 -1154696
## + speechiness                1       0.0    0.0 -1154696
## + instrumentalness           1       0.0    0.0 -1154696
## + time_signature             1       0.0    0.0 -1154696
```

```
## - spotify_track_explicit    3    3615.5 3615.5   -32027
##
## Step:  AIC=-1154697
## in_billboard ~ spotify_track_explicit
```

We then proceed to picking the most important variables from the above selection. The predictors we select will be hard conded in the regression model we create in the later stage of the code. In the meantime, we are going to import a dataset that was generated by our Python code in the ipynb file called Random Songs Generator. This contains a lot of popular songs. We import that data and then

Training and Test Dataset

Then we use an algorithm that breaks our all_songs dataset into test and train dataset.

```
#logistic model
all_songs = all_songs %>%
mutate(
  in_billboard_factor = as.factor(in_billboard),
  spotify_track_explicit = ifelse(spotify_track_explicit == "True", 1, 0)
)

all_songs <- all_songs %>%
  mutate(id = row_number())
#Check IDs
head(all_songs$id)
```

```
## [1] 1 2 3 4 5 6
```

```
#Create training set
train <- all_songs %>%
  sample_frac(.70)
#Create test set
test  <- anti_join(all_songs, train, by = 'id')
```

Logistic Regression

```
spotify_glm <- logistic_reg(mode = "classification") %>%
set_engine("glm") %>%
fit(in_billboard_factor ~ danceability+energy+key+loudness+mode+speechiness +acousticness+valence+tempo

tidy(spotify_glm)
```

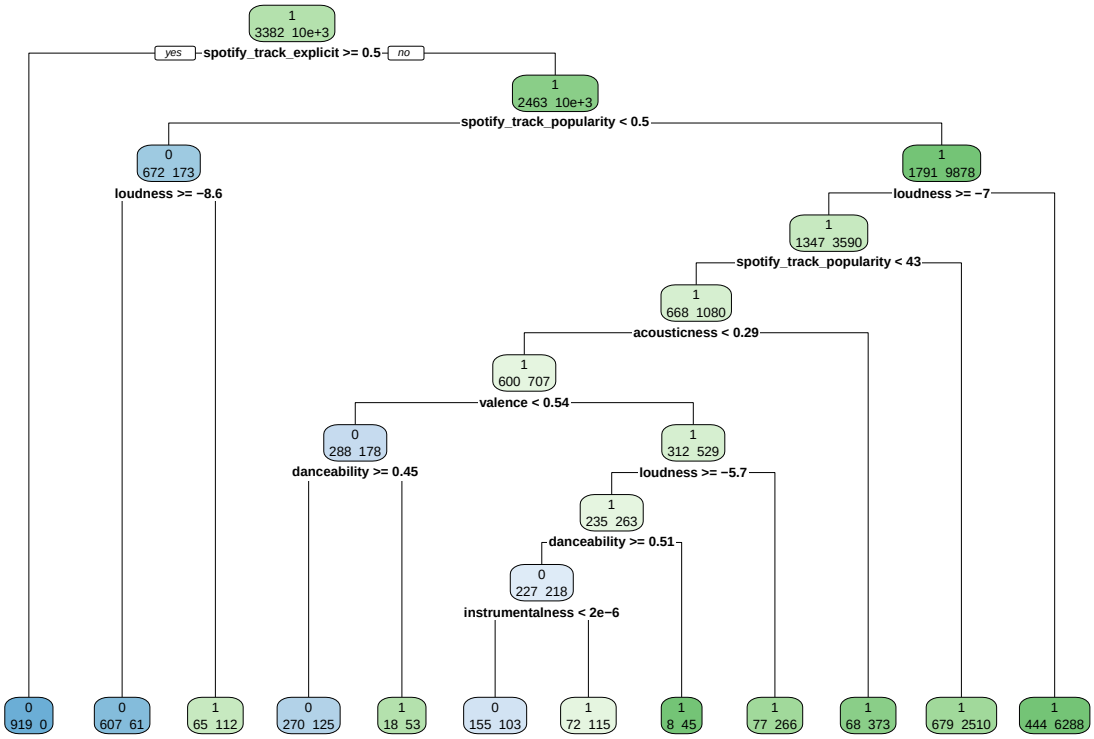
```
## # A tibble: 13 x 5
##    term                    estimate std.error statistic   p.value
##    <chr>                      <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)              -1.95      0.521      -3.74  1.83e-  4
##  2 danceability             -3.18      0.220     -14.5   2.05e- 47
##  3 energy                    0.776     0.238       3.26  1.10e-  3
##  4 key                       0.0104    0.00725     1.44  1.49e-  1
```

```
##  5 loudness                  -0.356     0.0141    -25.3   1.12e-140
##  6 mode                       0.624     0.0541     11.5   9.61e- 31
##  7 speechiness                0.653     0.345       1.89  5.88e-  2
##  8 acousticness               1.13      0.143       7.91  2.61e- 15
##  9 valence                    2.73      0.138      19.8   3.40e- 87
## 10 tempo                     -0.00204   0.00103    -1.99  4.69e-  2
## 11 time_signature            -0.248     0.113      -2.20  2.75e-  2
## 12 spotify_track_popularity   0.0317    0.00119    26.7   1.98e-157
## 13 spotify_track_explicit   -19.9     190.         -0.105 9.17e-  1
```

Classification Tree

```
#decision tree

spotify_tree <- decision_tree(mode = "classification") %>%
  set_engine("rpart", control = rpart.control(cp = 0.005)) %>%
  fit(in_billboard_factor ~ danceability+energy+key+loudness+mode+speechiness +acousticness+instrumental

library(rpart.plot)
rpart.plot(spotify_tree$fit, extra = 1)
```

```
## Warning: Cannot retrieve the data used to build the model (so cannot determine roundint and is.binary
## To silence this warning:
##     Call rpart.plot with roundint=FALSE,
##     or rebuild the rpart model with model=TRUE.
```



```
log_reg_predictions = augment(spotify_glm, new_data = test) %>%
  mutate(model = "logistic")

tree_predictions = augment(spotify_tree, new_data = test) %>%
  mutate(model = "classification tree")
log_reg_predictions
```

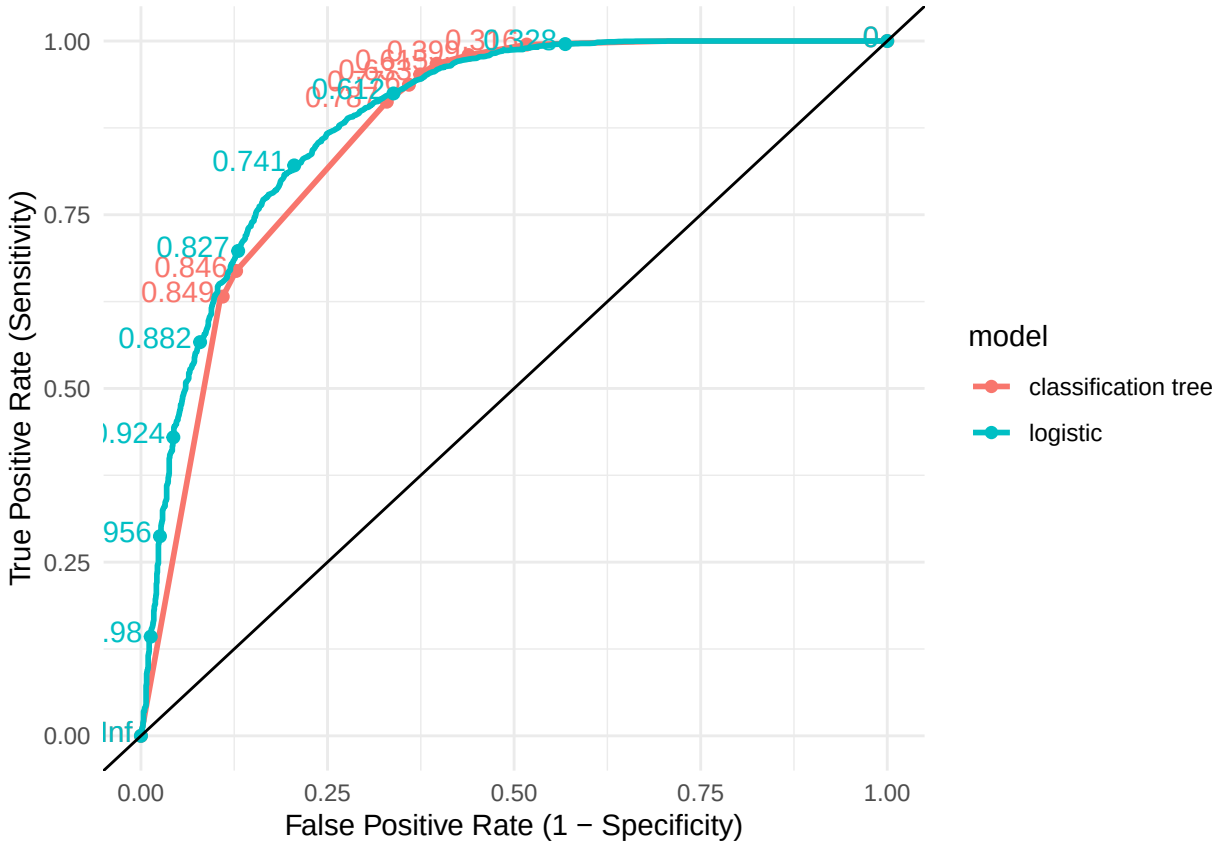
```
## # A tibble: 5,757 x 22
##    song_id     danceability energy   key loudness  mode speechiness acousticness
##    <chr>              <dbl>  <dbl> <int>    <dbl> <int>       <dbl>        <dbl>
##  1 Hall of Fa~        0.421  0.873    10    -4.34     1      0.0564       0.0654
##  2 The Heart ~        0.616  0.789     7    -4.87     0      0.0377       0.053
##  3 DynamiteTa~        0.751  0.783     4    -3.72     1      0.0859       0.00379
##  4 Danza Kudu~        0.706  0.89      0    -6.58     1      0.0847       0.0855
##  5 Coconut Tr~        0.687  0.855     7    -5.34     0      0.0416       0.0486
##  6 Don't You ~        0.608  0.828     2    -3.60     1      0.051        0.125
##  7 Radioactiv~        0.473  0.777     9    -3.70     1      0.059        0.119
##  8 Scream & S~        0.772  0.685     5    -6.85     1      0.0696       0.019
##  9 Please Don~        0.513  0.768     4    -4.87     0      0.0587       0.0118
## 10 HeathensTw~        0.732  0.396     4    -9.35     0      0.0286       0.0841
## # ... with 5,747 more rows, and 14 more variables: instrumentalness <dbl>,
## #   liveness <dbl>, valence <dbl>, tempo <dbl>, time_signature <int>,
## #   spotify_track_popularity <int>, spotify_track_explicit <dbl>,
## #   in_billboard <dbl>, in_billboard_factor <fct>, id <int>, .pred_class <fct>,
## #   .pred_0 <dbl>, .pred_1 <dbl>, model <chr>
```

```
all_predictions = log_reg_predictions %>%
  bind_rows(tree_predictions)
```

ROC Curves

The ROC Curves we made are using the test datasets to see if our model is doing well for the test dataset as well as the training dataset.

```
#ROC curves

library(plotROC) #extension of ggplot2
ggplot(all_predictions,
         aes(d = in_billboard,
         m = .pred_1,
         col = model)) +
  geom_roc(n.cuts = 10, labelround = 3) +
  geom_abline(intercept = 0) +
  labs(x = "False Positive Rate (1 - Specificity)",
       y = "True Positive Rate (Sensitivity)") +
  theme_bw() + theme_minimal()
```



Accuracy

For checking accuracy we are going to use the test dataset on both the classification tree and the logistic regression model.

```
# Logistic Regression Model
augment(spotify_glm, new_data = test) %>%
select(in_billboard, .pred_class) %>%
table()
```

```
##              .pred_class
## in_billboard    0    1
##            0  840  611
##            1  131 4175
```

```
# Classification Tree
augment(spotify_tree, new_data = test) %>%
select(in_billboard, .pred_class) %>%
table()
```

```
##              .pred_class
## in_billboard    0    1
##            0  876  575
##            1  141 4165
```