Final Project

Diane Lin, Stephanie You

Song hotness - artist familiarity, artist hotness, song duration, artist.terms (genre), start of fade out, start of time fade in, song tempo

Linear model, interaction linear model, classifier (logistic)

```r
library(tidymodels)
```

```
-- Attaching packages -------------------------------------- tidymodels 1.1.1 --
```

```
v broom        1.0.5     v recipes      1.0.9
v dials        1.2.0     v rsample      1.2.0
v dplyr        1.1.4     v tibble       3.2.1
v ggplot2      3.4.4     v tidyr        1.3.0
v infer        1.0.5     v tune         1.1.2
v modeldata    1.3.0     v workflows    1.1.3
v parsnip      1.1.1     v workflowsets 1.0.1
v purrr        1.0.2     v yardstick    1.3.1
```

```
-- Conflicts ----------------------------------------- tidymodels_conflicts() --
x purrr::discard() masks scales::discard()
x dplyr::filter()  masks stats::filter()
x dplyr::lag()     masks stats::lag()
x recipes::step()  masks stats::step()
* Learn how to get started at https://www.tidymodels.org/start/
```

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v readr     2.1.5
v lubridate 1.9.3     v stringr   1.5.1
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x readr::col_factor() masks scales::col_factor()
x purrr::discard()    masks scales::discard()
x dplyr::filter()     masks stats::filter()
x stringr::fixed()    masks recipes::fixed()
x dplyr::lag()        masks stats::lag()
x readr::spec()       masks yardstick::spec()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
music <- read_csv("music.csv")
```

```
Rows: 10000 Columns: 35
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (4): artist.id, artist.name, artist.terms, song.id
dbl (31): artist.familiarity, artist.hotttnesss, artist.latitude, artist.loc...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
music_filter <- music |>
  filter(str_detect(artist.terms, "rock") |
         str_detect(artist.terms, "rap") |
         str_detect(artist.terms, "pop") |
         str_detect(artist.terms, "country"))

# creating genrealized variables, bc orignial has some subgenres.

music_filter <- music_filter |>
  mutate(gen_genre =
           if_else(grepl("rock", music_filter$artist.terms),"rock",
                   if_else(grepl("rap", music_filter$artist.terms), "rap",
                           if_else(grepl("pop", music_filter$artist.terms), "pop",
                                   if_else(grepl("country", music_filter$artist.terms), "coun

music_filter <- music_filter |>
  filter(song.hotttnesss > 0)


music_filter
```

```
# A tibble: 1,471 x 36
```

```
   artist.familiarity artist.hotttnesss artist.id          artist.latitude
                <dbl>             <dbl> <chr>                        <dbl>
 1              0.651             0.402 ARXR32B1187FB57099             0  
 2              0.636             0.448 ARD842G1187B997376            43.6
 3              0.707             0.513 ARYKCQI1187FB3B18F             0  
 4              0.435             0.306 AR47JEX1187B995D81            37.8
 5              0.809             0.488 ARPQ4Z01187FB3A736            29.4
 6              0.661             0.443 ARV1JVD1187B9AD195            35.9
 7              0.718             0.479 ARS1OWB1187B99EEAD             0  
 8              0.570             0.412 AROEL1B1187B988B90             0  
 9              0.643             0.501 AR3793X1187FB50CB3             0  
10              0.751             0.524 ARDGB6U1187FB3AD07            51.5
# i 1,461 more rows
# i 32 more variables: artist.location <dbl>, artist.longitude <dbl>,
#   artist.name <chr>, artist.similar <dbl>, artist.terms <chr>,
#   artist.terms_freq <dbl>, release.id <dbl>, release.name <dbl>,
#   song.artist_mbtags <dbl>, song.artist_mbtags_count <dbl>,
#   song.bars_confidence <dbl>, song.bars_start <dbl>,
#   song.beats_confidence <dbl>, song.beats_start <dbl>, ...
```

```
tidy_summary <- sapply(music, summary)
print(tidy_summary)
```

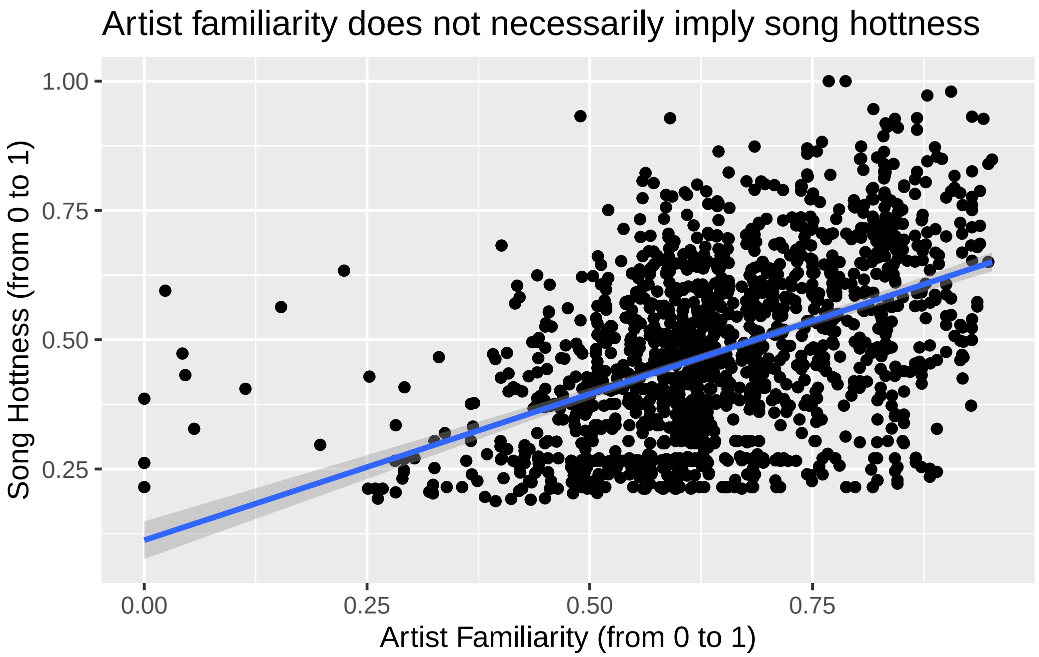
```
$artist.familiarity
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.4676  0.5636  0.5652  0.6680  1.0000 

$artist.hotttnesss
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.3253  0.3807  0.3856  0.4539  1.0825 

$artist.id
   Length     Class      Mode 
    10000 character character 

$artist.latitude
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -41.28    0.00    0.00   13.90   34.42   69.65 

$artist.location
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   0.000   0.000   0.078   0.000 780.000 
```

```
$artist.longitude
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-162.44  -73.95    0.00  -23.92    0.00  174.77

$artist.name
   Length     Class      Mode
    10000 character character

$artist.similar
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0       0       0       0       0       0

$artist.terms
   Length     Class      Mode
    10000 character character

$artist.terms_freq
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
      0.0       0.9       1.0     224.9       1.0 2239217.0

$release.id
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      0  172858  333103  371024  573532  823599

$release.name
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0     0.0     0.0    23.1     0.0 85555.0

$song.artist_mbtags
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.00e+00 0.00e+00 0.00e+00 3.33e-05 0.00e+00 3.33e-01

$song.artist_mbtags_count
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.0000  0.5247  1.0000  9.0000

$song.bars_confidence
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0350  0.1200  0.2396  0.3510  8.8552

$song.bars_start
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.4416  0.7855  1.0653  1.2241 59.7435

$song.beats_confidence
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.4098  0.6860  0.6140  0.8820  1.0000

$song.beats_start
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-60.0000   0.1947   0.3326   0.4285   0.5008  12.2458

$song.duration
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
    1.044   176.032   223.059   240.622   276.375 22050.000

$song.end_of_fade_in
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.0000  0.1990  0.7567  0.4210 43.1190

$song.hotttnesss
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.0000 -1.0000  0.0000 -0.2415  0.4051  1.0000

$song.id
   Length     Class      Mode
    10000 character character

$song.key
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   2.000   5.000   5.367   8.000 904.803

$song.key_confidence
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.0000  0.2250  0.4690  0.4515  0.6590 19.0810

$song.loudness
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-51.643 -13.160  -9.380 -10.484  -6.531   0.566

$song.mode
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   1.000   0.691   1.000   1.000

$song.mode_confidence
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.3600  0.4870  0.4778  0.6060  1.0000 

$song.start_of_fade_out
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 -21.39  168.86  213.86  229.88  266.27 1813.43 

$song.tatums_confidence
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.2370  0.5000  0.5079  0.7742  9.2276 

$song.tatums_start
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.1107  0.1915  0.2999  0.2947 12.2458 

$song.tempo
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   0.00   96.96  120.16  122.90  144.01  262.83 

$song.time_signature
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  0.000   3.000   4.000   3.564   4.000   7.000 

$song.time_signature_confidence
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
  0.0000   0.0978   0.5510   0.5998   0.8640 898.8910 

$song.title
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
    0.00     0.00     0.00    10.01     0.00 94496.00 

$song.year
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    0.0     0.0     0.0   934.7  2000.0  2010.0 
```

```r
ggplot(music_filter, aes(x = artist.familiarity, y = song.hotttnesss)) +
  geom_point()+
  geom_smooth(method = "lm", se = TRUE) +
  labs(x = "Artist Familiarity (from 0 to 1)", y = "Song Hottness (from 0 to 1)",
       title = "Artist familiarity does not necessarily imply song hottness")
```

```
`geom_smooth()` using formula = 'y ~ x'
```



```
#LINEAR MODEL

music_filter <- music_filter |>
  filter(song.hotttnesss > 0)

music_filter
```

```
# A tibble: 1,471 x 36
   artist.familiarity artist.hotttnesss artist.id          artist.latitude
                <dbl>             <dbl> <chr>                        <dbl>
 1              0.651             0.402 ARXR32B1187FB57099             0
 2              0.636             0.448 ARD842G1187B997376            43.6
 3              0.707             0.513 ARYKCQI1187FB3B18F             0
 4              0.435             0.306 AR47JEX1187B995D81            37.8
 5              0.809             0.488 ARPQ4Z01187FB3A736            29.4
 6              0.661             0.443 ARV1JVD1187B9AD195            35.9
 7              0.718             0.479 ARS1OWB1187B99EEAD             0
 8              0.570             0.412 AROEL1B1187B988B90             0
 9              0.643             0.501 AR3793X1187FB50CB3             0
10              0.751             0.524 ARDGB6U1187FB3AD07            51.5
# i 1,461 more rows
# i 32 more variables: artist.location <dbl>, artist.longitude <dbl>,
#   artist.name <chr>, artist.similar <dbl>, artist.terms <chr>,
#   artist.terms_freq <dbl>, release.id <dbl>, release.name <dbl>,
#   song.artist_mbtags <dbl>, song.artist_mbtags_count <dbl>,
#   song.bars_confidence <dbl>, song.bars_start <dbl>,
#   song.beats_confidence <dbl>, song.beats_start <dbl>, ...
```

```
m1 <- lm(song.hotttnesss ~ artist.familiarity + artist.hotttnesss
         + song.duration + as.factor(gen_genre) + song.start_of_fade_out
         + song.end_of_fade_in + song.tempo,
         data = music_filter)


tidy(m1)
```

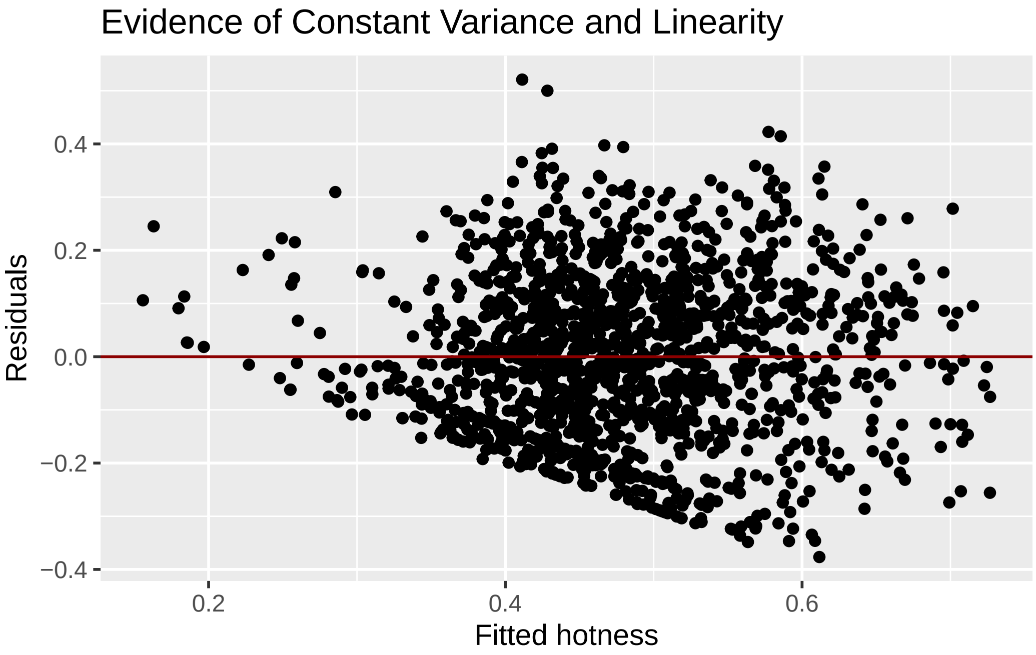
```
# A tibble: 10 x 5
   term                        estimate std.error statistic  p.value
   <chr>                          <dbl>     <dbl>     <dbl>    <dbl>
 1 (Intercept)                 0.0296    0.0298      0.994  3.20e- 1
 2 artist.familiarity          0.330     0.0439      7.53   9.09e-14
 3 artist.hotttnesss           0.335     0.0484      6.92   6.58e-12
 4 song.duration               0.00113   0.000637    1.78   7.52e- 2
 5 as.factor(gen_genre)pop     0.0446    0.0209      2.14   3.28e- 2
 6 as.factor(gen_genre)rap     0.0103    0.0229      0.450  6.53e- 1
 7 as.factor(gen_genre)rock    0.0285    0.0199      1.43   1.53e- 1
 8 song.start_of_fade_out     -0.00106   0.000651   -1.63   1.03e- 1
 9 song.end_of_fade_in         0.000104  0.00254     0.0409 9.67e- 1
10 song.tempo                  0.000205  0.000119    1.73   8.38e- 2
```

```
summary(m1)
```

```


Call:
lm(formula = song.hotttnesss ~ artist.familiarity + artist.hotttnesss +
    song.duration + as.factor(gen_genre) + song.start_of_fade_out +
    song.end_of_fade_in + song.tempo, data = music_filter)

Residuals:
     Min       1Q   Median       3Q      Max
-0.37666 -0.11609 -0.00803  0.10539  0.52091

Coefficients:
```

```
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)               0.0295918  0.0297706   0.994   0.3204    
artist.familiarity        0.3302057  0.0438741   7.526 9.09e-14 ***
artist.hotttnesss         0.3349121  0.0483736   6.923 6.58e-12 ***
song.duration             0.0011337  0.0006368   1.780   0.0752 .  
as.factor(gen_genre)pop   0.0446009  0.0208737   2.137   0.0328 *  
as.factor(gen_genre)rap   0.0102870  0.0228824   0.450   0.6531    
as.factor(gen_genre)rock  0.0284675  0.0199278   1.429   0.1534    
song.start_of_fade_out   -0.0010607  0.0006506  -1.630   0.1033    
song.end_of_fade_in       0.0001040  0.0025412   0.041   0.9673    
song.tempo                0.0002053  0.0001187   1.730   0.0838 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1512 on 1461 degrees of freedom
Multiple R-squared:  0.2547,    Adjusted R-squared:  0.2501 
F-statistic: 55.48 on 9 and 1461 DF,  p-value: < 2.2e-16
```

```r
m1_aug <- augment(m1)
m1_aug|>
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "darkred") +
  labs(title = "Evidence of Constant Variance and Linearity",
         x = "Fitted hotness",
         y = "Residuals")
```

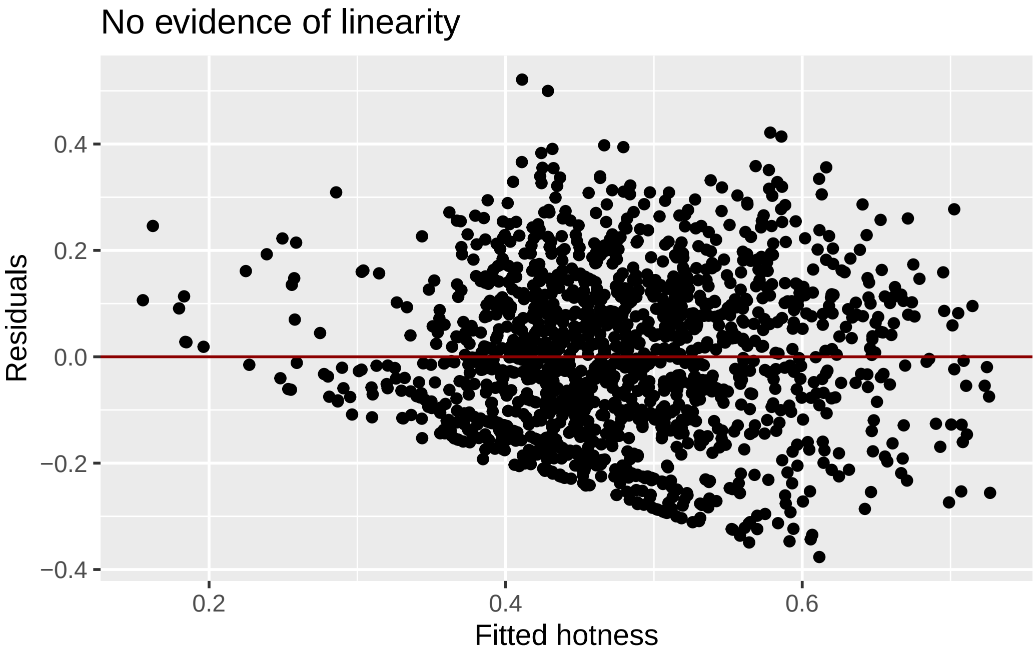


```
#LINEAR W INTERACTION, tempo x duration
library(dplyr)
#music_filter

m2 <- lm(song.hotttnesss ~ artist.familiarity + artist.hotttnesss + song.duration + as.facto
         + song.end_of_fade_in + song.tempo + song.duration*song.tempo, data = music_filter)
summary(m2)
```

```
Call:
lm(formula = song.hotttnesss ~ artist.familiarity + artist.hotttnesss +
    song.duration + as.factor(gen_genre) + song.start_of_fade_out +
    song.end_of_fade_in + song.tempo + song.duration * song.tempo,
    data = music_filter)

Residuals:
     Min       1Q   Median       3Q      Max
-0.37655 -0.11598 -0.00797  0.10564  0.52116

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)               1.821e-02  4.685e-02   0.389   0.6976
```

```
artist.familiarity        3.304e-01  4.389e-02   7.527 9.02e-14 ***
artist.hotttnesss         3.351e-01  4.839e-02   6.925 6.50e-12 ***
song.duration             1.145e-03  6.381e-04   1.795   0.0728 .
as.factor(gen_genre)pop   4.454e-02  2.088e-02   2.133   0.0331 *
as.factor(gen_genre)rap   1.017e-02  2.289e-02   0.444   0.6569
as.factor(gen_genre)rock  2.848e-02  1.993e-02   1.429   0.1533
song.start_of_fade_out   -1.021e-03  6.630e-04  -1.540   0.1239
song.end_of_fade_in       1.189e-04  2.542e-03   0.047   0.9627
song.tempo                2.978e-04  3.169e-04   0.940   0.3476
song.duration:song.tempo -4.142e-07  1.316e-06  -0.315   0.7531
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1513 on 1460 degrees of freedom
Multiple R-squared:  0.2548,    Adjusted R-squared:  0.2496
F-statistic: 49.91 on 10 and 1460 DF,  p-value: < 2.2e-16
```

```r
m2_aug <- augment(m2)
m2_aug|>
  ggplot(aes(x = .fitted, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0, color = "darkred") +
  labs(title = "No evidence of linearity",
         x = "Fitted hotness",
         y = "Residuals")
```


No evidence of linearity

```
#classifier

categorical <- music_filter |>
  mutate(cathot = ifelse(song.hotttnesss <= 0.5, 0, 1))

categorical
```

```
# A tibble: 1,471 x 37
   artist.familiarity artist.hotttnesss artist.id          artist.latitude
                <dbl>             <dbl> <chr>                        <dbl>
 1              0.651             0.402 ARXR32B1187FB57099             0
 2              0.636             0.448 ARD842G1187B997376            43.6
 3              0.707             0.513 ARYKCQI1187FB3B18F             0
 4              0.435             0.306 AR47JEX1187B995D81            37.8
 5              0.809             0.488 ARPQ4Z01187FB3A736            29.4
 6              0.661             0.443 ARV1JVD1187B9AD195            35.9
 7              0.718             0.479 ARS1OWB1187B99EEAD             0
 8              0.570             0.412 AROEL1B1187B988B90             0
 9              0.643             0.501 AR3793X1187FB50CB3             0
10              0.751             0.524 ARDGB6U1187FB3AD07            51.5
# i 1,461 more rows
# i 33 more variables: artist.location <dbl>, artist.longitude <dbl>,
#   artist.name <chr>, artist.similar <dbl>, artist.terms <chr>,
#   artist.terms_freq <dbl>, release.id <dbl>, release.name <dbl>,
#   song.artist_mbtags <dbl>, song.artist_mbtags_count <dbl>,
#   song.bars_confidence <dbl>, song.bars_start <dbl>,
#   song.beats_confidence <dbl>, song.beats_start <dbl>, ...
```

```
m3 <- glm(as.factor(cathot) ~ artist.familiarity + artist.hotttnesss
         + song.duration + as.factor(gen_genre) + song.start_of_fade_out
         + song.end_of_fade_in + song.tempo,
  data = categorical,
  family = "binomial")

tidy(m3)
```

```
# A tibble: 10 x 5
   term                        estimate std.error statistic  p.value
   <chr>                          <dbl>     <dbl>     <dbl>    <dbl>
 1 (Intercept)                 -5.55      0.503     -11.0   2.79e-28
 2 artist.familiarity           3.70      0.735       5.03  4.94e- 7
 3 artist.hotttnesss            3.90      0.828       4.72  2.38e- 6
 4 song.duration                0.0240    0.00984     2.44  1.48e- 2
 5 as.factor(gen_genre)pop      0.654     0.331       1.98  4.77e- 2
 6 as.factor(gen_genre)rap      0.163     0.358       0.455 6.49e- 1
 7 as.factor(gen_genre)rock     0.343     0.318       1.08  2.82e- 1
 8 song.start_of_fade_out      -0.0228    0.00999    -2.28  2.24e- 2
 9 song.end_of_fade_in         -0.0124    0.0372     -0.332 7.40e- 1
10 song.tempo                   0.00148   0.00173     0.854 3.93e- 1
```