Gurpinder Singh

STAT 108

1/26/2022

Load all the following library

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(broom)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
##
## Attaching package: 'pROC'
##
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(plotROC)
```

```
##
## Attaching package: 'plotROC'
##
## The following object is masked from 'package:pROC':
##
##     ggroc
```

```
library(arm)
```

```
## Loading required package: MASS
##
## Attaching package: 'MASS'
##
```

```
## The following object is masked from 'package:dplyr':
##
##     select
##
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loading required package: lme4
##
## arm (Version 1.13-1, built: 2022-8-25)
##
## Working directory is /Users/gurpindersingh/Desktop/stat108Real/Stat108/lab7
```

```
library(knitr)
```

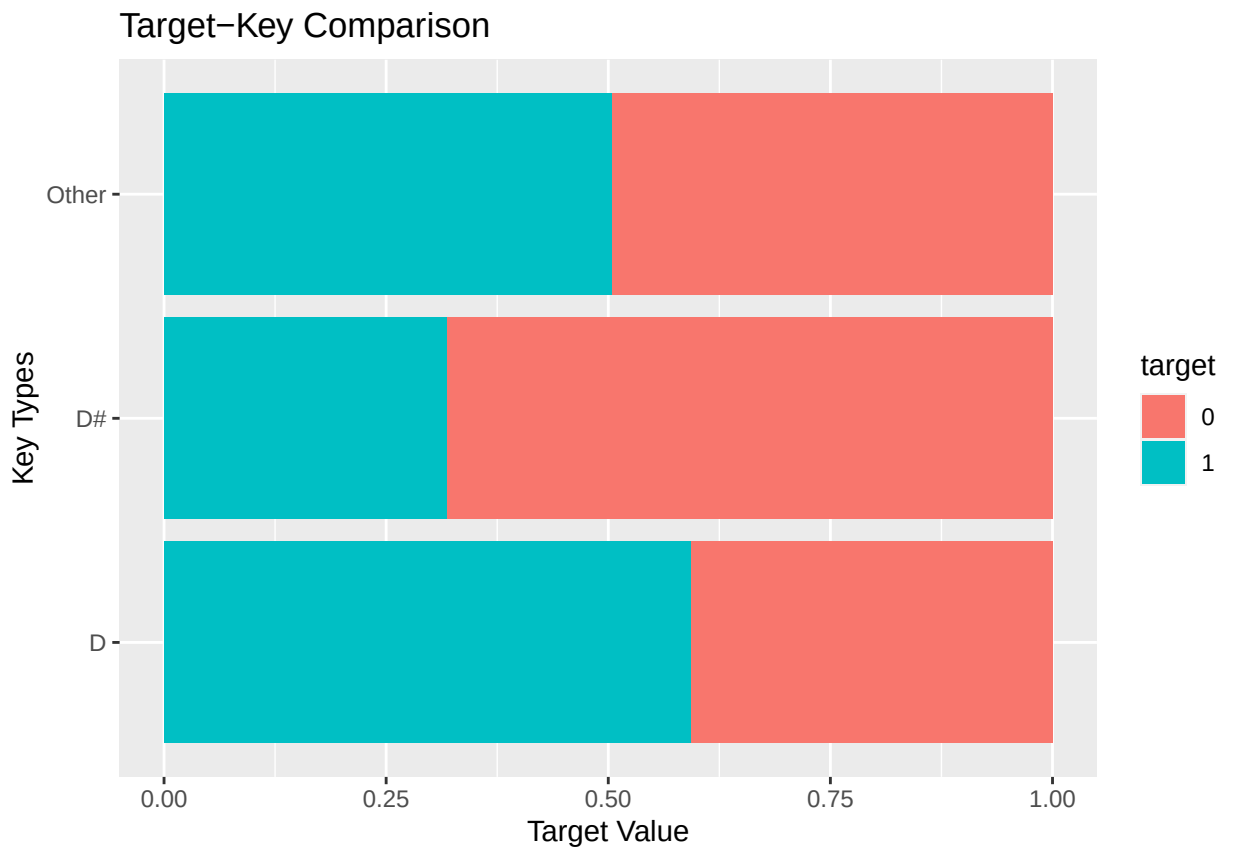
Exercise 1

```
spot <- read_csv("spotify.csv")
```

```
## New names:
## Rows: 2017 Columns: 17
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (2): song_title, artist dbl (15): ...1, acousticness, danceability,
## duration_ms, energy, instrumenta...
## 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.
## * `` -> `...1`
```

```
spot$target <- factor(spot$target)
spot <- spot %>% mutate(key = case_when(
           key == 2 ~ 'D',
           key == 3 ~ 'D#',
           key!=2 & key!=3 ~ "Other"))

ggplot(data = spot, aes(x = key, fill = target)) +
  geom_bar(position = "fill") +
  labs(x = "Key Types",y = "Target Value", title = "Target-Key Comparison") +
  coord_flip()
```



The following compares the key types to the target value of 0 and 1. It showcasesw the percentage of target values in each key types. Additionally Exercise 2: Following source was referenced: https://stats.oarc.ucla.edu/r/dae/logit-regression/ for Exercise 2

```
model <- glm(target ~ acousticness+ danceability+ duration_ms+ instrumentalness+ loudness+ speechiness+
tidy(model, conf.int = TRUE) %>% # output model
  kable(digits = 3) # format model output
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-2.955	0.276	-10.693	0	-3.504	-2.420
acousticness	-1.722	0.240	-7.182	0	-2.197	-1.257
danceability	1.630	0.344	4.737	0	0.958	2.308
duration_ms	0.000	0.000	4.225	0	0.000	0.000
instrumentalness	1.353	0.207	6.549	0	0.952	1.763
loudness	-0.087	0.017	-5.062	0	-0.122	-0.054
speechiness	4.072	0.583	6.985	0	2.947	5.234
valence	0.856	0.223	3.836	0	0.420	1.296

Exercise 3:

```
model2 <- glm(target ~ acousticness+ danceability+ duration_ms+ instrumentalness+ loudness+ speechiness
tidy(model2, conf.int = TRUE)
```

```
## # A tibble: 10 x 7
```

```
##    term                estimate   std.error statistic  p.value conf.low conf.h~1
##    <chr>                  <dbl>       <dbl>     <dbl>    <dbl>    <dbl>    <dbl>
##  1 (Intercept)      -2.51       0.311           -8.07 7.14e-16 -3.12e+0 -1.90e+0
##  2 acousticness     -1.70       0.241           -7.07 1.60e-12 -2.18e+0 -1.23e+0
##  3 danceability      1.65       0.345            4.77 1.80e- 6  9.75e-1  2.33e+0
##  4 duration_ms       0.00000286 0.000000684      4.19 2.82e- 5  1.55e-6  4.23e-6
##  5 instrumentalness  1.38       0.207            6.67 2.60e-11  9.81e-1  1.80e+0
##  6 loudness         -0.0866     0.0173          -5.02 5.21e- 7 -1.21e-1 -5.30e-2
##  7 speechiness       4.03       0.585            6.90 5.33e-12  2.90e+0  5.20e+0
##  8 valence           0.881      0.224            3.93 8.61e- 5  4.42e-1  1.32e+0
##  9 keyD#            -1.07       0.335           -3.20 1.36e- 3 -1.75e+0 -4.28e-1
## 10 keyOther         -0.494      0.169           -2.92 3.47e- 3 -8.28e-1 -1.65e-1
## # ... with abbreviated variable name 1: conf.high
```

```
anova(model, model2, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model 1: target ~ acousticness + danceability + duration_ms + instrumentalness +
##     loudness + speechiness + valence
## Model 2: target ~ acousticness + danceability + duration_ms + instrumentalness +
##     loudness + speechiness + valence + key
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      2009     2518.5
## 2      2007     2505.2  2   13.357 0.001258 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model)
```

```
##
## Call:
## glm(formula = target ~ acousticness + danceability + duration_ms +
##     instrumentalness + loudness + speechiness + valence, family = binomial,
##     data = spot)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1419  -1.0557   0.4035   1.0602   2.0880
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.955e+00  2.764e-01 -10.693  < 2e-16 ***
## acousticness     -1.722e+00  2.398e-01  -7.182 6.89e-13 ***
## danceability      1.630e+00  3.442e-01   4.737 2.17e-06 ***
## duration_ms       2.871e-06  6.795e-07   4.225 2.39e-05 ***
## instrumentalness  1.353e+00  2.066e-01   6.549 5.80e-11 ***
## loudness         -8.744e-02  1.727e-02  -5.062 4.14e-07 ***
## speechiness       4.072e+00  5.830e-01   6.985 2.85e-12 ***
## valence           8.564e-01  2.233e-01   3.836 0.000125 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2795.9  on 2016  degrees of freedom
## Residual deviance: 2518.5  on 2009  degrees of freedom
## AIC: 2534.5
##
## Number of Fisher Scoring iterations: 4
```

```
summary(model2)
```

```
##
## Call:
## glm(formula = target ~ acousticness + danceability + duration_ms +
##     instrumentalness + loudness + speechiness + valence + key,
##     family = binomial, data = spot)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1534  -1.0500   0.3981   1.0461   2.1031
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.509e+00  3.110e-01  -8.068 7.14e-16 ***
## acousticness     -1.702e+00  2.409e-01  -7.065 1.60e-12 ***
## danceability      1.649e+00  3.454e-01   4.774 1.80e-06 ***
## duration_ms       2.863e-06  6.836e-07   4.187 2.82e-05 ***
## instrumentalness  1.383e+00  2.075e-01   6.667 2.60e-11 ***
## loudness         -8.662e-02  1.726e-02  -5.018 5.21e-07 ***
## speechiness       4.034e+00  5.849e-01   6.896 5.33e-12 ***
## valence           8.809e-01  2.243e-01   3.927 8.61e-05 ***
## keyD#            -1.073e+00  3.350e-01  -3.204  0.00136 **
## keyOther         -4.939e-01  1.690e-01  -2.923  0.00347 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2795.9  on 2016  degrees of freedom
## Residual deviance: 2505.2  on 2007  degrees of freedom
## AIC: 2525.2
##
## Number of Fisher Scoring iterations: 4
```

Using a chisquare test we see the p value is low so we can state that adding the variable key produces a very similar model to model without key. Looking at the AIC value inside of the summarry of the two models we can see that model2 produces a slightly better accuracy and thus we can state model2 with the key variable in it is the better model.

Exercise 4:

```
tidy(model2, conf.int = TRUE) %>% # output model
  kable(digits = 3) # format model output
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-2.509	0.311	-8.068	0.000	-3.124	-1.904
acousticness	-1.702	0.241	-7.065	0.000	-2.179	-1.234
danceability	1.649	0.345	4.774	0.000	0.975	2.329
duration_ms	0.000	0.000	4.187	0.000	0.000	0.000
instrumentalness	1.383	0.207	6.667	0.000	0.981	1.795
loudness	-0.087	0.017	-5.018	0.000	-0.121	-0.053
speechiness	4.034	0.585	6.896	0.000	2.905	5.199
valence	0.881	0.224	3.927	0.000	0.442	1.322
keyD#	-1.073	0.335	-3.204	0.001	-1.745	-0.428
keyOther	-0.494	0.169	-2.923	0.003	-0.828	-0.165

The value of target will decrease by 1.073 if the value of key is d#, aka value of key is 3 Exercise 5

```
print(model2)
```

```
##
## Call:  glm(formula = target ~ acousticness + danceability + duration_ms +
##     instrumentalness + loudness + speechiness + valence + key,
##     family = binomial, data = spot)
##
## Coefficients:
##      (Intercept)      acousticness      danceability       duration_ms
##       -2.509e+00        -1.702e+00         1.649e+00         2.863e-06
## instrumentalness          loudness       speechiness           valence
##        1.383e+00        -8.662e-02         4.034e+00         8.809e-01
##            keyD#          keyOther
##       -1.073e+00        -4.939e-01
##
## Degrees of Freedom: 2016 Total (i.e. Null);  2007 Residual
## Null Deviance:       2796
## Residual Deviance: 2505  AIC: 2525
```

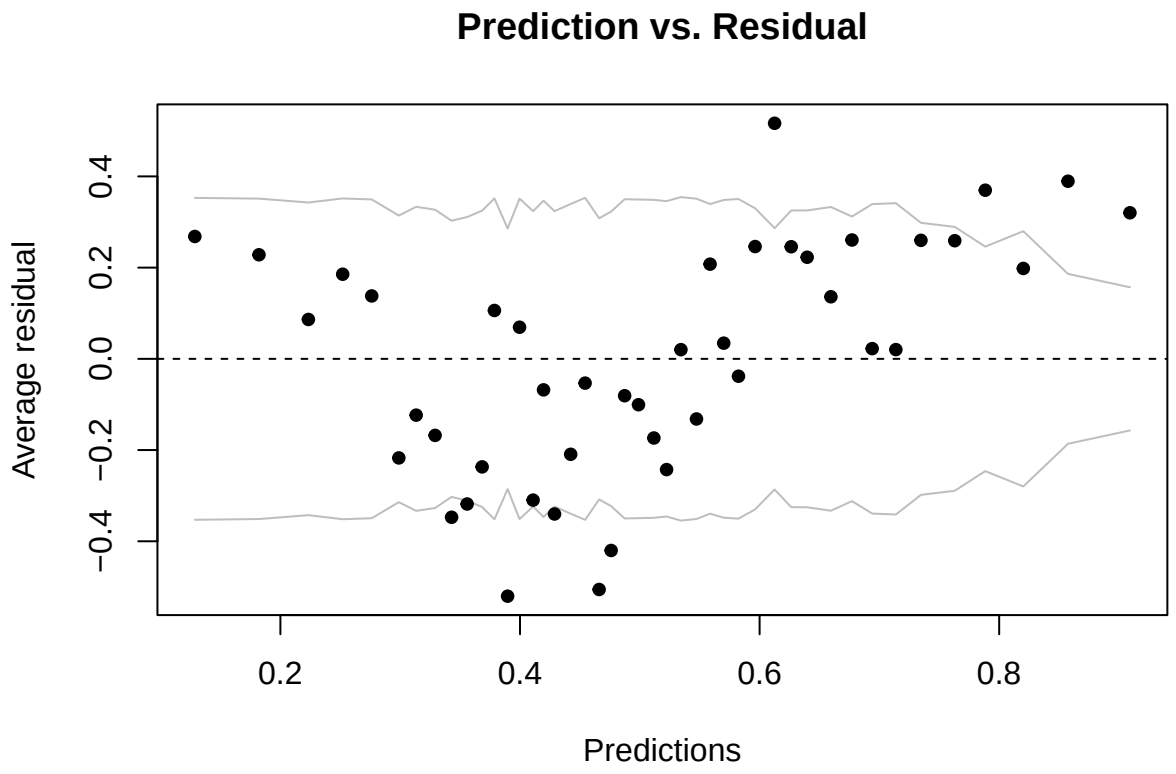
```
aug <- augment(model2, type.predict = "response")
print(aug)
```

```
## # A tibble: 2,017 x 15
##    target acoust~1 dance~2 durat~3 instr~4 loudn~5 speec~6 valence key   .fitted
##    <fct>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>   <dbl>
##  1 1       0.0102    0.833  204600 2.19e-2   -8.80  0.431    0.286 D       0.902
##  2 1       0.199     0.743  326933 6.11e-3  -10.4   0.0794   0.588 Other   0.638
##  3 1       0.0344    0.838  185707 2.34e-4   -7.15  0.289    0.173 D       0.783
##  4 1       0.604     0.494  199413 5.1 e-1  -15.2   0.0261   0.23  Other   0.422
##  5 1       0.18      0.678  392893 5.12e-1  -11.6   0.0694   0.904 Other   0.849
##  6 1       0.00479   0.804  251333 0         -6.68  0.185    0.264 Other   0.644
##  7 1       0.0145    0.739  241400 7.27e-6  -11.2   0.156    0.308 Other   0.680
##  8 1       0.0202    0.266  349667 6.64e-1  -11.6   0.0371   0.393 Other   0.695
##  9 1       0.0481    0.603  202853 0         -3.63  0.347    0.398 Other   0.635
## 10 1       0.00208   0.836  226840 0         -7.79  0.237    0.386 Other   0.729
## # ... with 2,007 more rows, 5 more variables: .resid <dbl>, .std.resid <dbl>,
## #   .hat <dbl>, .sigma <dbl>, .cooksd <dbl>, and abbreviated variable names
```

```
## #   1: acousticness, 2: danceability, 3: duration_ms, 4: instrumentalness,
## #   5: loudness, 6: speechiness
```

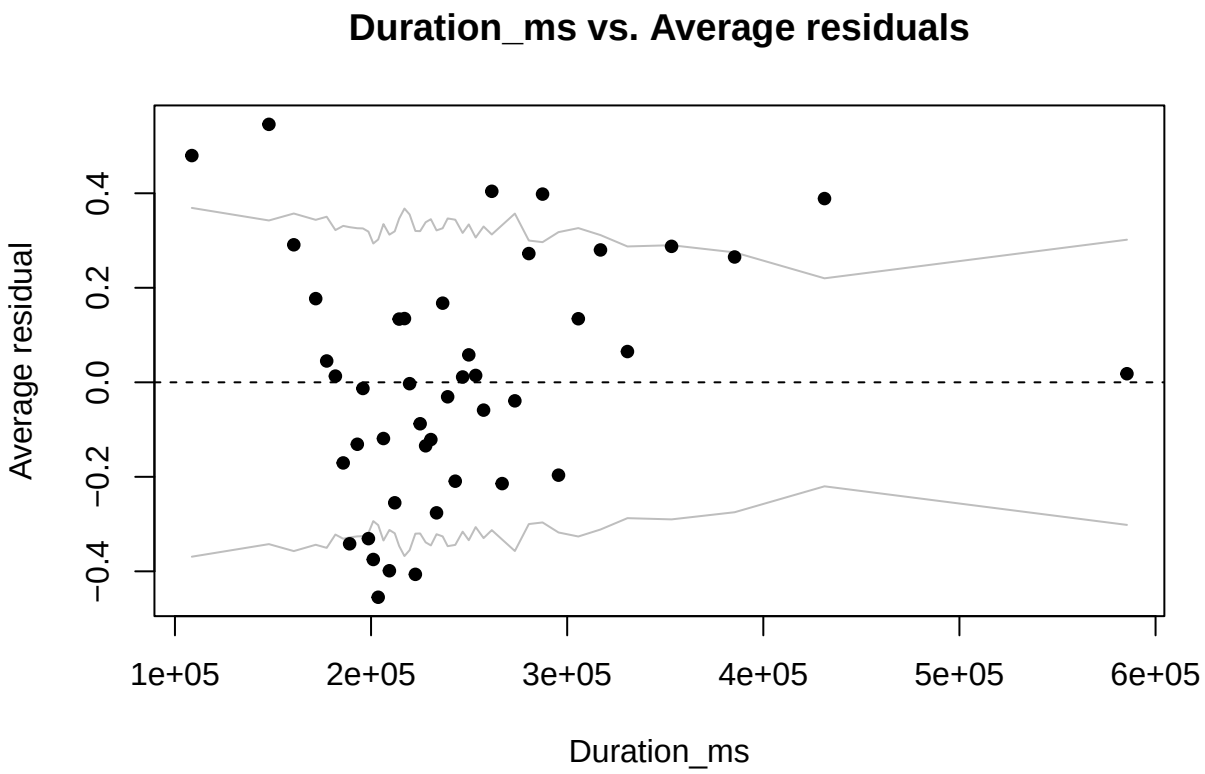
Exercise 6

```
arm::binnedplot(aug$.fitted ,aug$.resid,
    xlab="Predictions", ylab="Average residual",
    main="Prediction vs. Residual", col.int="gray")
```



Exercise 7

```
arm::binnedplot(aug$duration_ms ,aug$.resid,
    xlab="Duration_ms", ylab="Average residual",
    main="Duration_ms vs. Average residuals", col.int="gray")
```



Exercise 8

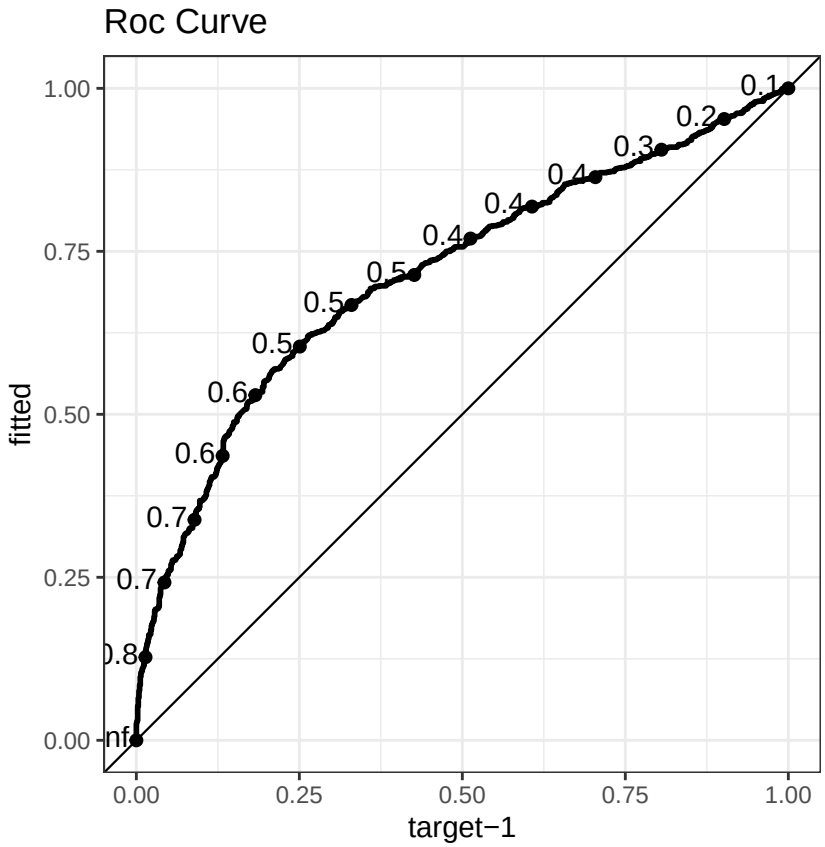
```
aug %>%
  group_by(key) %>%
  summarise(n = n(),mean = mean(.resid)) %>%
  kable(digits = 3) # format model output
```

key	n	mean
D	184	0.054
D#	63	-0.099
Other	1770	0.003

Exersize 9: There is no clear linear relationship as the residual vs predicted values plot showcases a u shape instead of a distinct linear line. So assumption is not satisfied.

Exersize 10: The following source was refrenced: https://rdrr.io/cran/plotROC/man/geom_roc.html

```
plot(ggplot(aug, aes(d= as.numeric(target) - 1,m= .fitted)) +
  geom_roc(n.cuts = 15) +
  geom_abline(slope = 1, intercept = 0, size = 0.4) +
  coord_equal() +
  theme_bw()+
  labs(x ="target-1",
       y = "fitted",
       title = "Roc Curve"))
```



Exersize 11: The model does effectively effectively differentiates between the songs the user likes versus those he doesn't. Exersize 12: I would choose a threshold value of .5725 as it is an inflection point on the curve where the True positive is maximized and false negative is small as possible.

Excersize 13:

```
aug <-aug %>% mutate(prediction = case_when(
           .fitted >= .5725 ~ '1',
           .fitted < .5725 ~ '0',))
aug %>%
  group_by(prediction) %>%
  summarise(n = n()) %>%
  kable(digits = 3) # format model output
```

prediction	n
0	1320
1	697

Excersize 14: