lab 7

Aisha Lakshman

3/3/2022

Packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## -- 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(knitr)
```

Part I: Data Prep and Modeling

Exercise 1

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

```
## New names:
## * `` -> ...1
```

```
## 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.
```

```
glimpse(spotify)
```

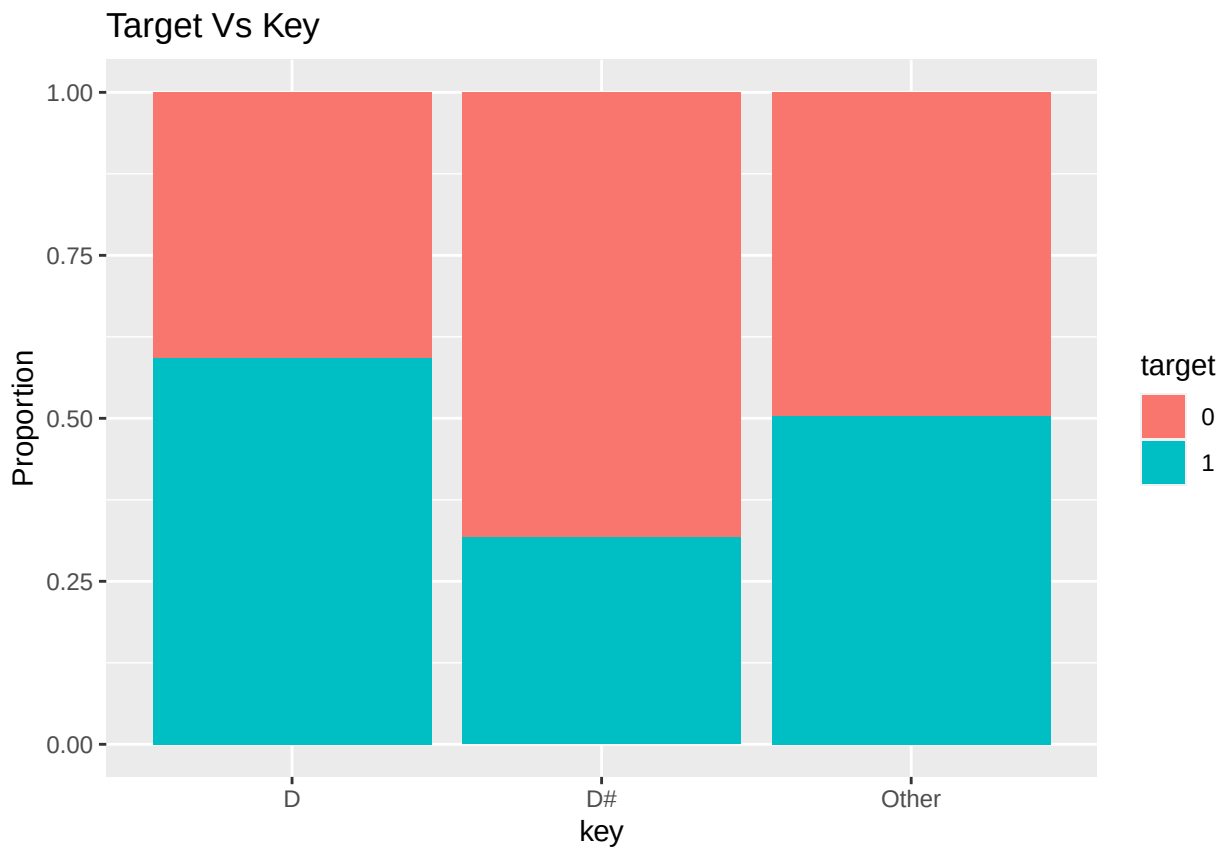
```
## Rows: 2,017
## Columns: 17
## $ ...1             <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
## $ acousticness     <dbl> 0.010200, 0.199000, 0.034400, 0.604000, 0.180000, 0.0~
## $ danceability     <dbl> 0.833, 0.743, 0.838, 0.494, 0.678, 0.804, 0.739, 0.26~
## $ duration_ms      <dbl> 204600, 326933, 185707, 199413, 392893, 251333, 24140~
## $ energy           <dbl> 0.434, 0.359, 0.412, 0.338, 0.561, 0.560, 0.472, 0.34~
## $ instrumentalness <dbl> 2.19e-02, 6.11e-03, 2.34e-04, 5.10e-01, 5.12e-01, 0.0~
## $ key              <dbl> 2, 1, 2, 5, 5, 8, 1, 10, 11, 7, 5, 10, 0, 0, 9, 6, 1,~
## $ liveness         <dbl> 0.1650, 0.1370, 0.1590, 0.0922, 0.4390, 0.1640, 0.207~
## $ loudness         <dbl> -8.795, -10.401, -7.148, -15.236, -11.648, -6.682, -1~
## $ mode             <dbl> 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0,~
## $ speechiness      <dbl> 0.4310, 0.0794, 0.2890, 0.0261, 0.0694, 0.1850, 0.156~
## $ tempo            <dbl> 150.062, 160.083, 75.044, 86.468, 174.004, 85.023, 80~
## $ time_signature   <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 4, 4, 4, 4, 4, 4, 4,~
## $ valence          <dbl> 0.286, 0.588, 0.173, 0.230, 0.904, 0.264, 0.308, 0.39~
## $ target           <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ song_title       <chr> "Mask Off", "Redbone", "Xanny Family", "Master Of Non~
## $ artist           <chr> "Future", "Childish Gambino", "Future", "Beach House"~
```

```
spotify <- spotify %>%
  drop_na() %>%
  mutate(target = as.factor(target),
         key = case_when(
           key == 2 ~ "D",
           key == 3 ~ "D#",
           TRUE ~ "Other"))
```

```
glimpse(spotify)
```

```
## Rows: 2,017
## Columns: 17
## $ ...1             <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
## $ acousticness     <dbl> 0.010200, 0.199000, 0.034400, 0.604000, 0.180000, 0.0~
## $ danceability     <dbl> 0.833, 0.743, 0.838, 0.494, 0.678, 0.804, 0.739, 0.26~
## $ duration_ms      <dbl> 204600, 326933, 185707, 199413, 392893, 251333, 24140~
## $ energy           <dbl> 0.434, 0.359, 0.412, 0.338, 0.561, 0.560, 0.472, 0.34~
## $ instrumentalness <dbl> 2.19e-02, 6.11e-03, 2.34e-04, 5.10e-01, 5.12e-01, 0.0~
## $ key              <chr> "D", "Other", "D", "Other", "Other", "Other", "Other"~
## $ liveness         <dbl> 0.1650, 0.1370, 0.1590, 0.0922, 0.4390, 0.1640, 0.207~
## $ loudness         <dbl> -8.795, -10.401, -7.148, -15.236, -11.648, -6.682, -1~
## $ mode             <dbl> 1, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0,~
## $ speechiness      <dbl> 0.4310, 0.0794, 0.2890, 0.0261, 0.0694, 0.1850, 0.156~
## $ tempo            <dbl> 150.062, 160.083, 75.044, 86.468, 174.004, 85.023, 80~
## $ time_signature   <dbl> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 3, 4, 4, 4, 4, 4, 4, 4,~
## $ valence          <dbl> 0.286, 0.588, 0.173, 0.230, 0.904, 0.264, 0.308, 0.39~
## $ target           <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ song_title       <chr> "Mask Off", "Redbone", "Xanny Family", "Master Of Non~
## $ artist           <chr> "Future", "Childish Gambino", "Future", "Beach House"~
```

```
ggplot(data = spotify, aes(x = key, fill = target)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion", title = "Target Vs Key")
```



The figure above plots the relationship between target and key. About 60% of key D songs have a target value of 1. About 30% of key D# songs have a target value of 1. 50% of key "Other" songs have a target value of 1.

Exercise 2

```
model <- glm(target ~ acousticness + danceability + duration_ms + instrumentalness + loudness + speechi

tidy(model, conf.int = TRUE, exponentiate = FALSE) %>%
  kable(format="markdown", digits = 5)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-2.95548	0.27640	-10.69288	0.00000	-3.50377	-2.41978
acousticness	-1.72231	0.23982	-7.18155	0.00000	-2.19743	-1.25678
danceability	1.63040	0.34421	4.73664	0.00000	0.95847	2.30841
duration_ms	0.00000	0.00000	4.22500	0.00002	0.00000	0.00000
instrumentalness	1.35291	0.20659	6.54883	0.00000	0.95236	1.76298
loudness	-0.08744	0.01727	-5.06219	0.00000	-0.12160	-0.05383
speechiness	4.07238	0.58302	6.98493	0.00000	2.94695	5.23420
valence	0.85638	0.22326	3.83573	0.00013	0.41997	1.29551

Exercise 3

```
model_key <- glm(target ~ acousticness + danceability + duration_ms + instrumentalness + loudness + spe

summary(model_key)$aic
```

```
## [1] 2525.16
```

```
summary(model)$aic
```

```
## [1] 2534.517
```

By comparing the AIC values for the models with key and without key, we can see that the model with key has a lower AIC value. It is best for a model to have a low AIC value, so working with "model_key" is the right way to go.

Exercise 4

```
tidy(model_key, conf.int = TRUE, exponentiate = FALSE) %>%
  kable(format="markdown", digits = 5)
```

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-2.50934	0.31102	-8.06817	0.00000	-3.12416	-1.90422
acousticness	-1.70210	0.24091	-7.06521	0.00000	-2.17930	-1.23436
danceability	1.64880	0.34536	4.77417	0.00000	0.97468	2.32911
duration_ms	0.00000	0.00000	4.18744	0.00003	0.00000	0.00000
instrumentalness	1.38316	0.20745	6.66741	0.00000	0.98104	1.79505
loudness	-0.08662	0.01726	-5.01848	0.00000	-0.12075	-0.05301
speechiness	4.03380	0.58491	6.89650	0.00000	2.90456	5.19917
valence	0.88094	0.22434	3.92682	0.00009	0.44248	1.32224
keyD#	-1.07319	0.33497	-3.20385	0.00136	-1.74517	-0.42805
keyOther	-0.49390	0.16898	-2.92288	0.00347	-0.82793	-0.16468

For observations of key D# the value for target with increase by eˆ-1.07319.

Part II: Checking Assumptions

Exercise 5

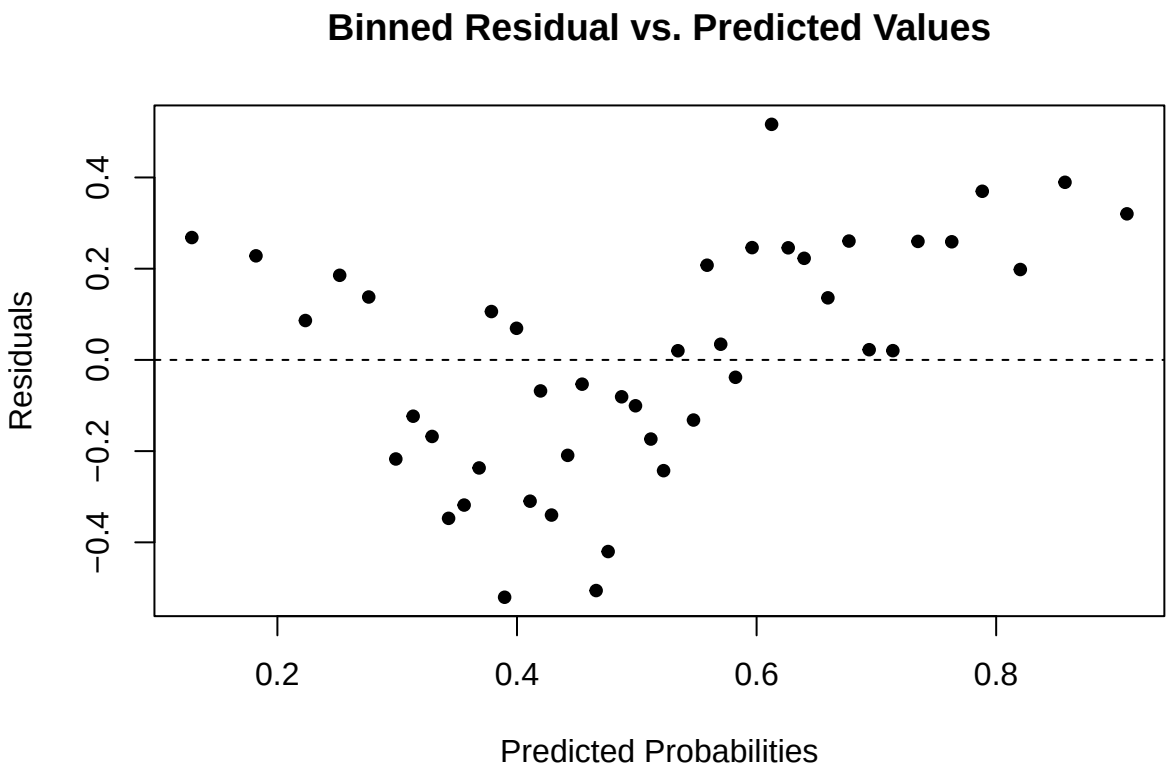
```
final_model <- augment(model_key, type.predict = "response", type.residuals = "deviance")

glimpse(final_model)
```

```
## Rows: 2,017
## Columns: 15
## $ target           <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ acousticness     <dbl> 0.010200, 0.199000, 0.034400, 0.604000, 0.180000, 0.0~
## $ danceability     <dbl> 0.833, 0.743, 0.838, 0.494, 0.678, 0.804, 0.739, 0.26~
## $ duration_ms      <dbl> 204600, 326933, 185707, 199413, 392893, 251333, 24140~
## $ instrumentalness <dbl> 2.19e-02, 6.11e-03, 2.34e-04, 5.10e-01, 5.12e-01, 0.0~
## $ loudness         <dbl> -8.795, -10.401, -7.148, -15.236, -11.648, -6.682, -1~
## $ speechiness      <dbl> 0.4310, 0.0794, 0.2890, 0.0261, 0.0694, 0.1850, 0.156~
## $ valence          <dbl> 0.286, 0.588, 0.173, 0.230, 0.904, 0.264, 0.308, 0.39~
## $ key              <chr> "D", "Other", "D", "Other", "Other", "Other", "Other"~
## $ .fitted          <dbl> 0.9015924, 0.6379788, 0.7829667, 0.4223972, 0.8489462~
## $ .resid           <dbl> 0.4551764, 0.9481036, 0.6995214, 1.3128664, 0.5722926~
## $ .std.resid       <dbl> 0.4567745, 0.9493120, 0.7028629, 1.3163187, 0.5733889~
## $ .hat             <dbl> 0.006985200, 0.002544191, 0.009485687, 0.005238444, 0~
## $ .sigma           <dbl> 1.117466, 1.117311, 1.117402, 1.117126, 1.117439, 1.1~
## $ .cooksd          <dbl> 7.731894e-05, 1.451076e-04, 2.679972e-04, 7.238900e-0~
```

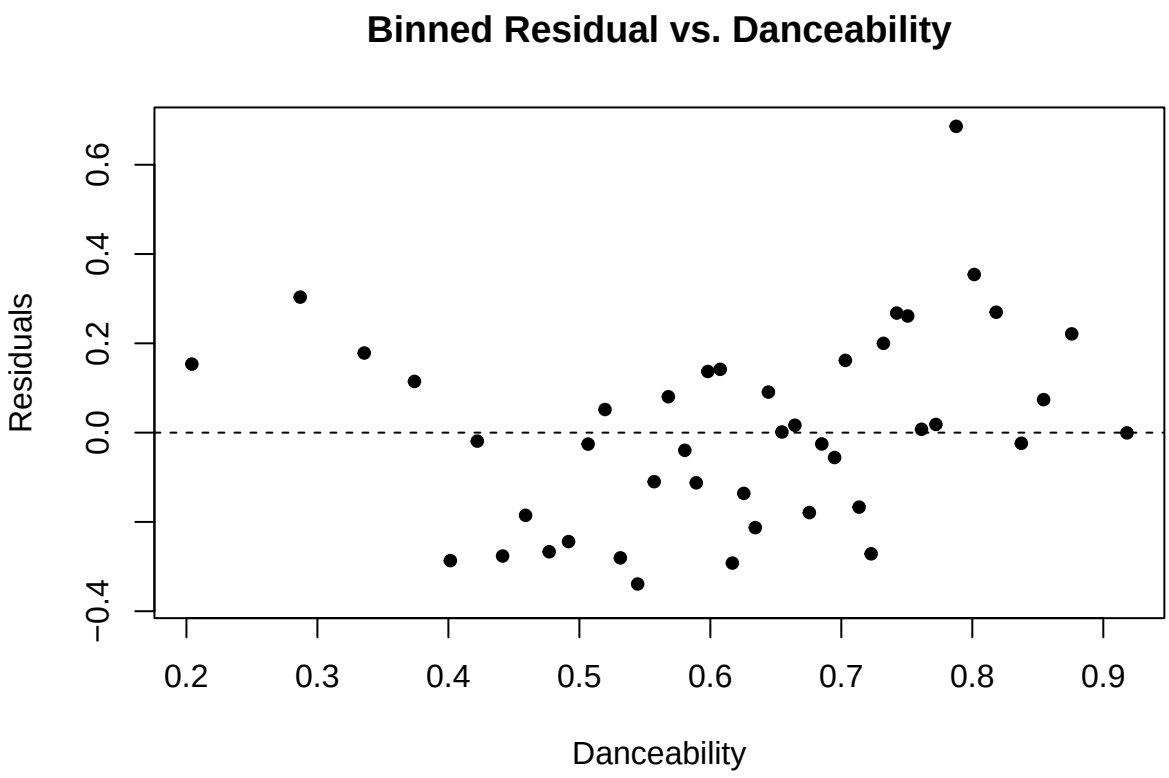
Exercise 6

```
arm::binnedplot(x = final_model$.fitted, y = final_model$.resid,
                xlab = "Predicted Probabilities",
                ylab = "Residuals",
                main = "Binned Residual vs. Predicted Values",
                col.int = FALSE)
```



Exercise 7

```
arm::binnedplot(x = final_model$danceability, y = final_model$.resid,
                xlab = "Danceability",
                ylab = "Residuals",
                main = "Binned Residual vs. Danceability",
                col.int = FALSE)
```



Exercise 8

```
final_model %>%
  mutate(resid_key = if_else(.resid > 1 | .resid < -1, "high Residual", "low Residual")) %>%
  group_by(key, resid_key) %>%
  summarise(n = n()) %>%
  kable(format="markdown")
```

```
## `summarise()` has grouped output by 'key'. You can override using the `.groups`
## argument.
```

key	resid_key	n
D	high Residual	98
D	low Residual	86
D#	high Residual	21
D#	low Residual	42
Other	high Residual	1039
Other	low Residual	731

Exercise 9

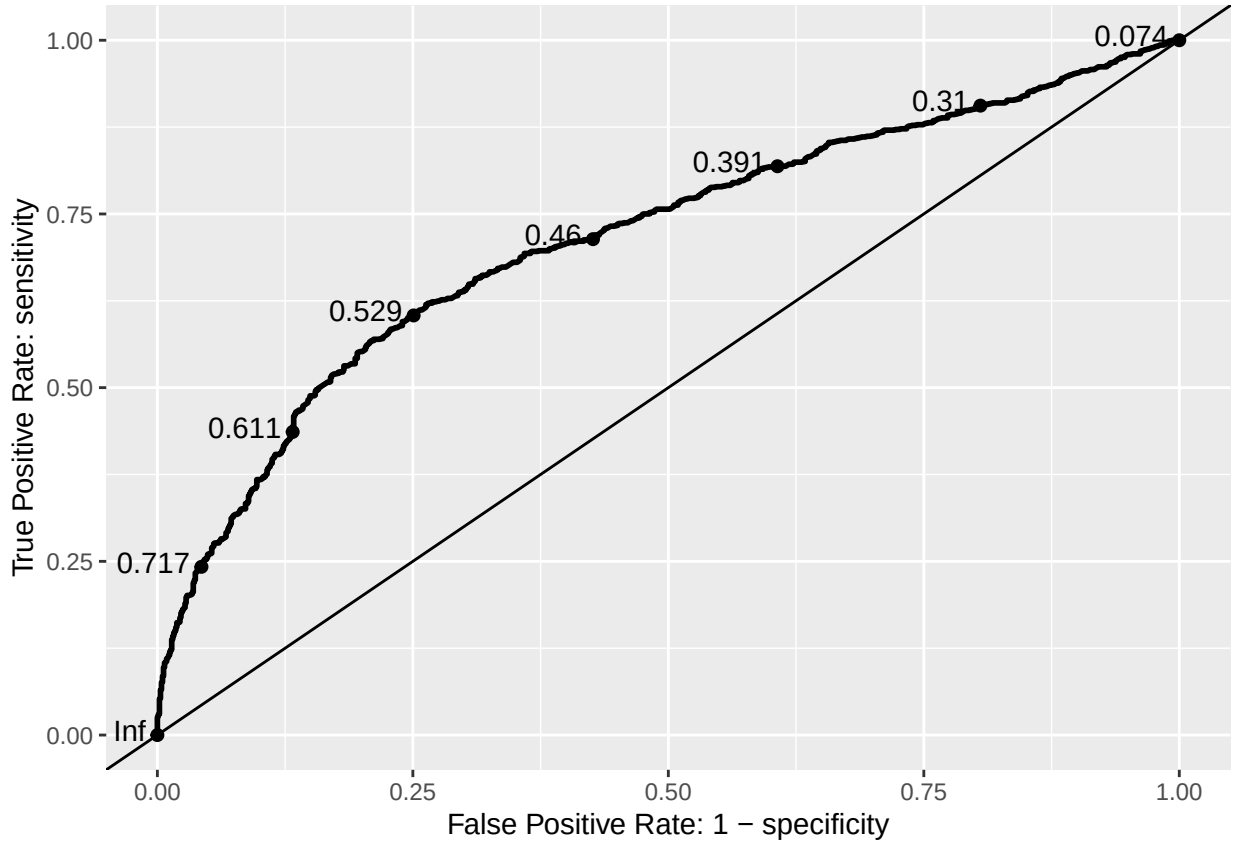
The plot in exercise 6 does not indicate a linear relationship between the fitted and predictor values. Therefore, I would say that the linearity assumption is not satisfied.

Part 3: Model Assessment and Prediction

Exercise 10

```
roc_plot <- ggplot(final_model, aes(d = as.numeric(target), m = .fitted)) +
  geom_roc(n.cuts = 8, labelround = 3) +
  geom_abline(intercept = 0) +
  labs(x = "False Positive Rate: 1 - specificity",
       y = "True Positive Rate: sensitivity")
roc_plot
```

```
## Warning in verify_d(data$d): D not labeled 0/1, assuming 1 = 0 and 2 = 1!
```




```
calc_auc(roc_plot)$AUC
```

```
## Warning in verify_d(data$d): D not labeled 0/1, assuming 1 = 0 and 2 = 1!

## [1] 0.7137869
```

Exercise 11

It seems like this model does not effectively differentiate the songs the user likes versus those he doesn't. The curve shown in my ROC plot is above x = 0, y = 0, representing a random classifier.

Exercise 12

I think a threshold of 0.58 would be best. This threshold value optimizes the true positive rate while maintaining a small false positive rate.

Exercise 13

```
threshold = 0.58
final_model %>%
  mutate(prediction = if_else(.fitted < threshold, "0:No","1:Yes")) %>%
  group_by(target, prediction) %>%
  summarise(n = n()) %>%
  kable(format="markdown")
```

```
## `summarise()` has grouped output by 'target'. You can override using the
## `.groups` argument.
```

target	prediction	n
0	0:No	836
0	1:Yes	161
1	0:No	508
1	1:Yes	512

Exercise 14

Proportion of true positives (sensitivity) = (512 / 512 + 508) = 0.502 Proportion of false positives (1 - specificity) = (508 / 512 + 508) = 0.498 Misclassification rate = (508 + 161 / 512 + 161 + 508 + 836) = 0.302