lab7

```
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(knitr)
library(broom)
library(nnet)
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 1.12-2, built: 2021-10-15)

## Working directory is C:/Users/theoh/Desktop/lab7
```

Exercise 1:

```
data <- read.csv("data.csv")
glimpse(data)
```

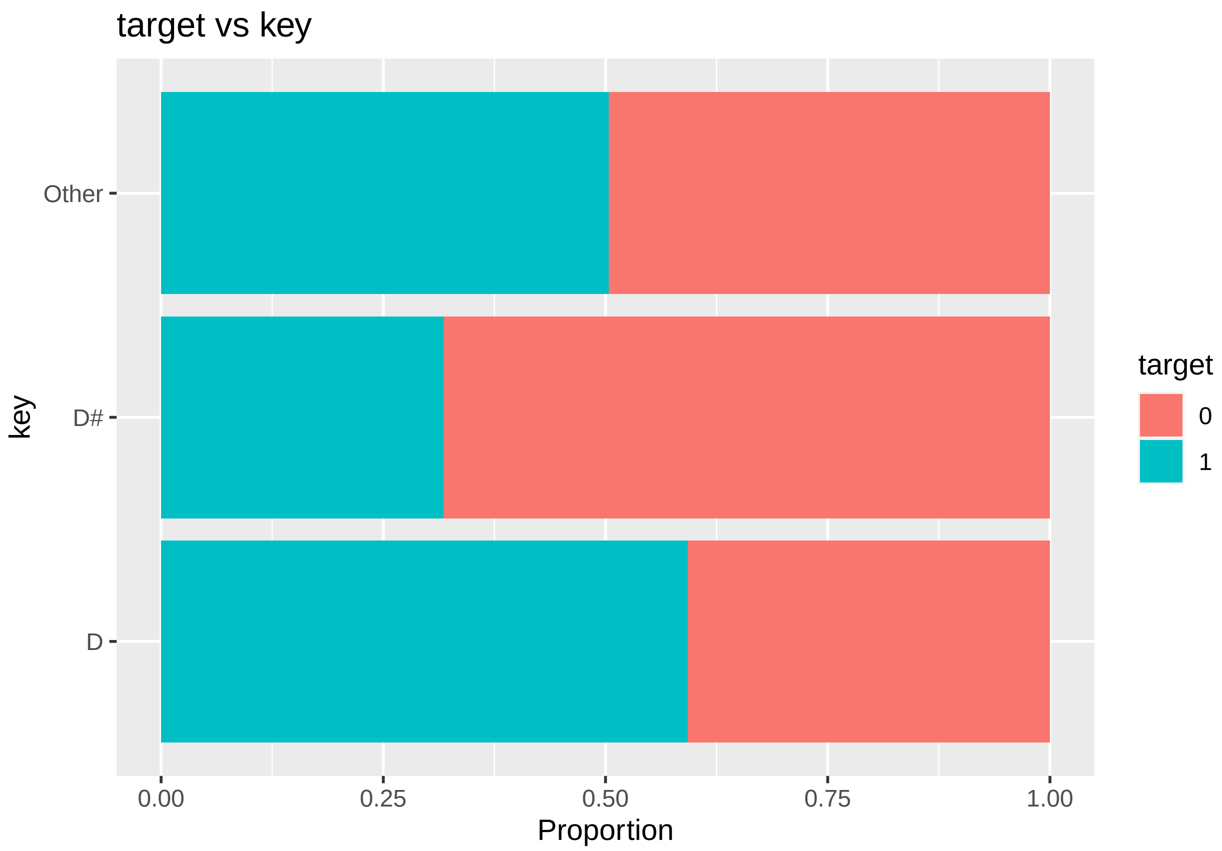
```
## Rows: 2,017
## Columns: 17
## $ X                <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,~
## $ acousticness     <dbl> 0.010200, 0.199000, 0.034400, 0.604000, 0.180000, 0.0~
## $ danceability     <dbl> 0.833, 0.743, 0.838, 0.494, 0.678, 0.804, 0.739, 0.26~
## $ duration_ms      <int> 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              <int> 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             <int> 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           <int> 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"~
```

```
data_lv <- data %>% mutate(target = as.factor(target), key = as.factor(ifelse(key == 2, "D", ifelse(key

glimpse(data_lv)
```

```
## Rows: 2,017
## Columns: 17
## $ X                <int> 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      <int> 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              <fct> D, Other, D, Other, Other, 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             <int> 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"~
```

```
p <- ggplot(data = data_lv, aes(x = key, fill = target)) +
  geom_bar(position = "fill") +
  labs(y = "Proportion",
       title = "target vs key") +
  coord_flip()
p
```



It appears that D and Other both have around an equal proportion of 0 and 1 targets with slightly more 1 target values. Key D# has a little over twice as many 0 target values than 1 target values.

Exercise 2:

```
target_m_red <- glm(target ~ acousticness + danceability + duration_ms + instrumentalness + loudness +
              data = data_lv, family = binomial)
tidy(target_m_red, conf.int = TRUE, exponentiate = FALSE) %>%
  kable(format = "markdown", digits = 3)
```

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:

```
target_m_full <- glm(target ~ acousticness + danceability + duration_ms + instrumentalness + loudness +
              data = data_lv, family = binomial)
```

```
anova(target_m_red, target_m_full, 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
```

There is evidence to suggest that the key is a significant predictor because we have a low p value of .001258. Therefor based on the test, we should add key to the model.

Exercise 4:

```
model <- target_m_full
tidy(model, conf.int = TRUE, exponentiate = FALSE) %>%
  kable(format = "markdown", digits = 3)
```

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

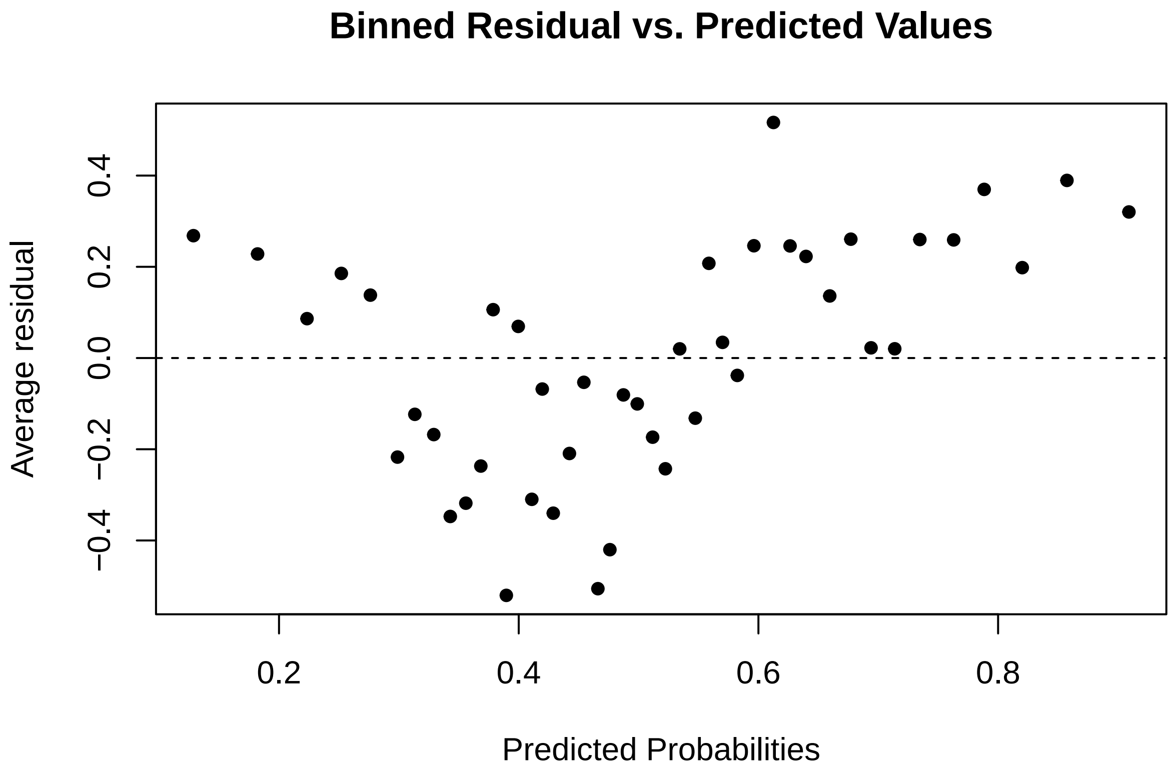
The keyD# coefficent tells us how the log odds of the target = 1 will change if our track is in the key of D#.

Exercise 5:

```r
m_aug <- augment(model, type.predict = "response",
                        type.residuals = "deviance")
```

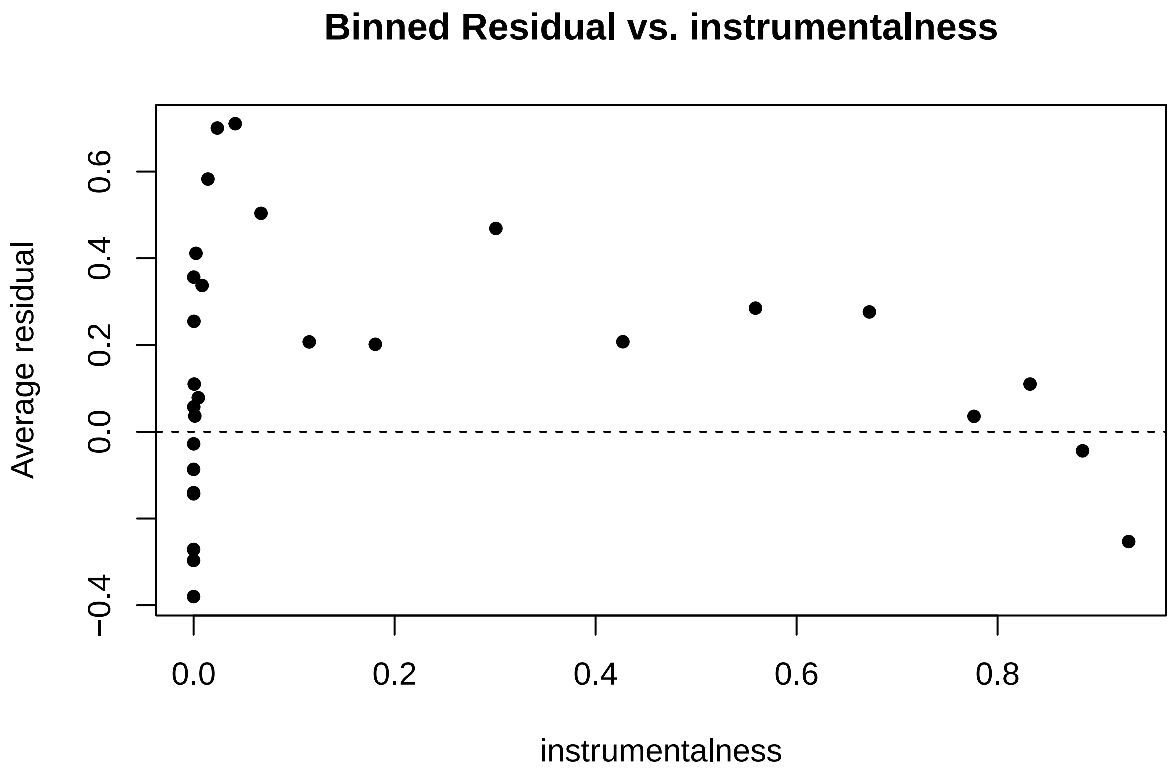
Exercise 6:

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



Exercise 7:

```
arm::binnedplot(x = m_aug$instrumentalness,
                y = m_aug$.resid,
                col.int = FALSE,
                xlab = "instrumentalness",
                main = "Binned Residual vs. instrumentalness")
```



Exercise 8:

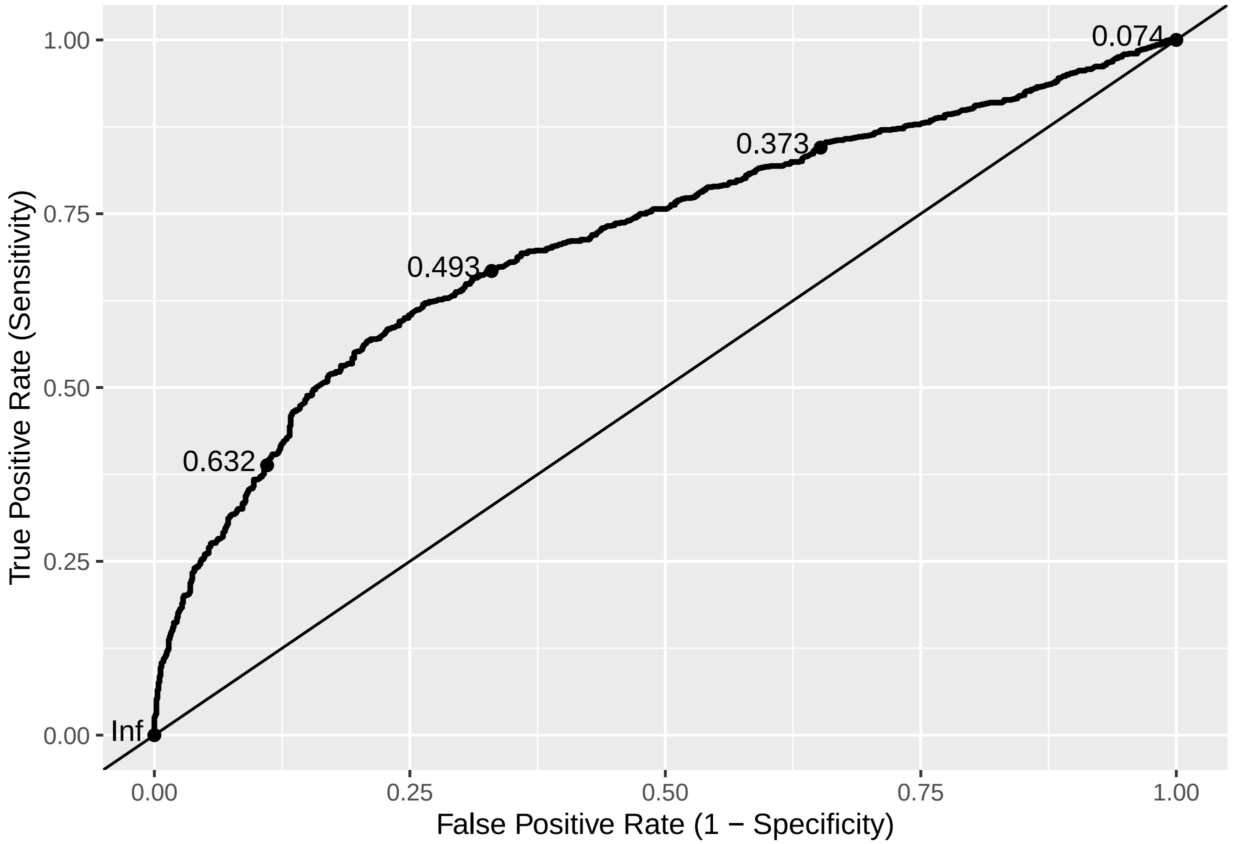
```
m_aug %>%
  group_by(key) %>%
  summarise(mean_resid = mean(.resid))
```

```
## # A tibble: 3 x 2
##   key   mean_resid
##   <fct>      <dbl>
## 1 D        0.0542
## 2 D#      -0.0992
## 3 Other    0.00316
```

Exercise 9: Both the key and instrumental residuals do not show evidence of constant variance. There also seems to be a partern accociated with the average residual vs probability plot. Based on this, linearity assumption is not statsfied.

Exercise 10:

```
(roc_curve <- ggplot(m_aug,
                     aes(d = as.numeric(target) - 1,
                         m = .fitted)) +
  geom_roc(n.cuts = 5, labelround = 3) +
  geom_abline(intercept = 0) +
  labs(x = "False Positive Rate (1 - Specificity)",
       y = "True Positive Rate (Sensitivity)") )
```



```
calc_auc(roc_curve)$AUC
```

```
## [1] 0.7137869
```

Exercise 11: The model appears to be somewhat effective. However, we would like the AOC to be higher.

Exercise 12:

```
threshold <- .493
```

I chose this threshold because it is closest to the top left corner of the plot. That is, maximum true positive rate and minimum false positive rate.

Exercise 13:

```
m_aug %>%
  mutate(predict_target = if_else(.fitted > threshold, "1", "0")) %>%
  group_by(target, predict_target) %>%
  summarise(n = n()) %>%
  kable(format="markdown")
```

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

target	predict_target	n
0	0	671
0	1	326
1	0	340
1	1	680

Exercise 14: The proprotion of true positives is 680/(680+340) = 2/3 The proportion of false positives is 326/(326+671)= .32 The misclassification rate is (340+326)/(671+326+340+680) = .33