STA303 - Final Project

Yakun Wang - 1007377358

April 7, 2024

```
library(readr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.0.2
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(knitr)

# # Load the original data and clean data
mmh_data <- read_csv("mmh_survey_data.csv")
```

```
## Rows: 736 Columns: 33
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (26): Timestamp, Primary streaming service, While working, Instrumentali...
## dbl  (7): Age, Hours per day, BPM, Anxiety, Depression, Insomnia, OCD
##
## 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.
```

```
mmh <- mmh_data %>%
  na.omit() %>%
  rename(Hours = `Hours per day`,
         While_working = `While working`,
         Favourite = `Fav genre`,
         Foreign = `Foreign languages`,
         Classical_freq = `Frequency [Classical]`,
         Country_freq = `Frequency [Country]`,
         EDM_freq = `Frequency [EDM]`,
         Folk_freq = `Frequency [Folk]`,
         Gospel_freq = `Frequency [Gospel]`,
         Hippop_freq = `Frequency [Hip hop]`,
         Jazz_freq = `Frequency [Jazz]`,
```

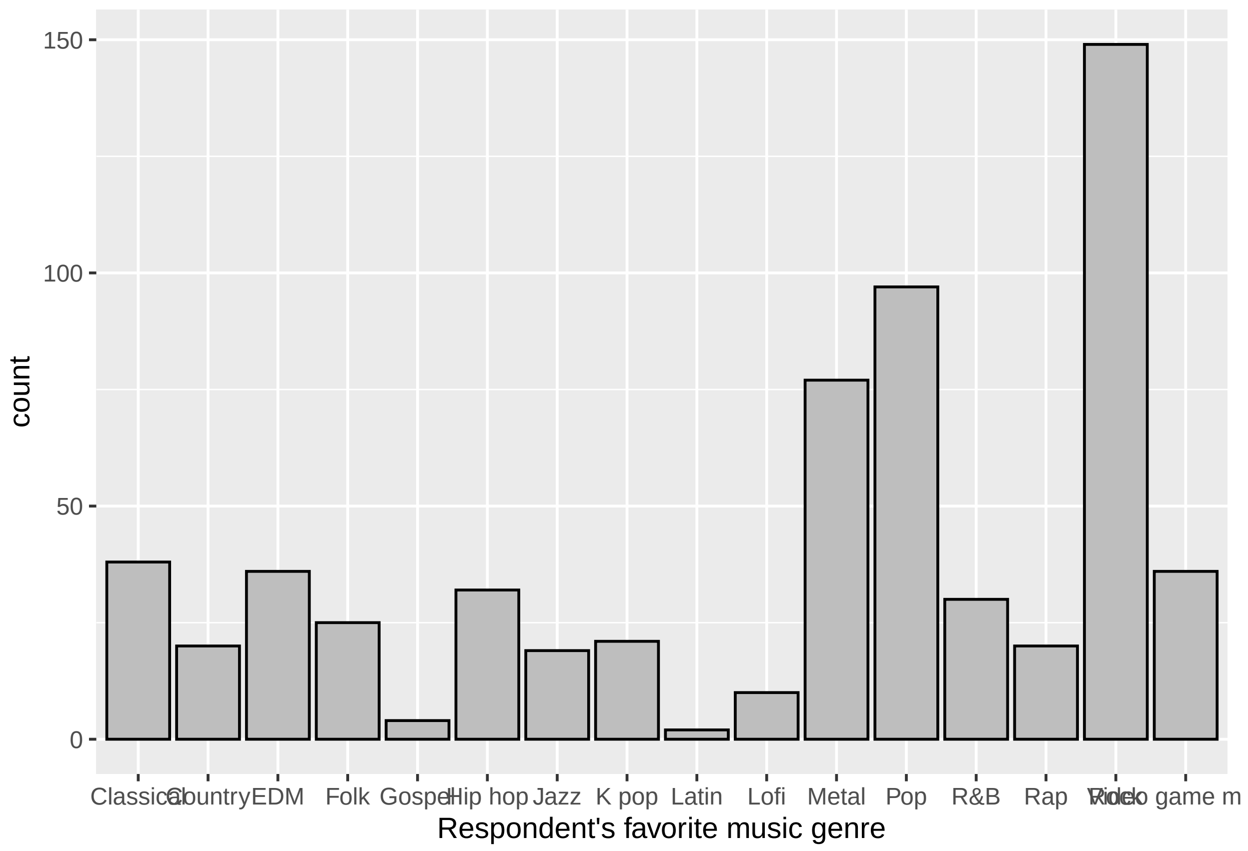
```r
         Kpop_freq = `Frequency [K pop]`,
         Latin_freq = `Frequency [Latin]`,
         Lofi_freq = `Frequency [Lofi]`,
         Metal_freq = `Frequency [Metal]`,
         Pop_freq = `Frequency [Pop]`,
         RnB_freq = `Frequency [R&B]`,
         Rap_freq = `Frequency [Rap]`,
         Rock_freq = `Frequency [Rock]`,
         VGM_freq = `Frequency [Video game music]`,
         Music_effects = `Music effects`) %>%
  mutate(Music_effects = ifelse(Music_effects == "Improve", "Improve", "No effect")) %>%
  select(Age, Hours, While_working, Instrumentalist, Composer, Favourite, Exploratory,
         Foreign, BPM, Classical_freq, Country_freq, EDM_freq, Folk_freq, Gospel_freq, Hippop_freq,
         Jazz_freq, Kpop_freq, Latin_freq, Lofi_freq, Metal_freq, Pop_freq,
         RnB_freq, Rap_freq, Rock_freq, VGM_freq, Anxiety, Depression, Insomnia, OCD, Music_effects)

# Randomly split cleaned data into training set (75%) and test set (25%)
rows <- sample(1:616, 462, replace = FALSE)
training <- mmh[rows,]
test <- mmh[-rows,]

# Summary for all variables
summary(mmh)
```

```
##       Age            Hours        While_working      Instrumentalist
##  Min.   :10.00   Min.   : 0.000   Length:616         Length:616
##  1st Qu.:18.00   1st Qu.: 2.000   Class :character   Class :character
##  Median :21.00   Median : 3.000   Mode  :character   Mode  :character
##  Mean   :24.79   Mean   : 3.702
##  3rd Qu.:27.00   3rd Qu.: 5.000
##  Max.   :89.00   Max.   :24.000
##    Composer          Favourite         Exploratory          Foreign
##  Length:616         Length:616         Length:616         Length:616
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       BPM            Classical_freq     Country_freq         EDM_freq
##  Min.   :        0   Length:616         Length:616         Length:616
##  1st Qu.:      100   Class :character   Class :character   Class :character
##  Median :      120   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  1623500
##  3rd Qu.:      144
##  Max.   :999999999
##   Folk_freq          Gospel_freq        Hippop_freq         Jazz_freq
##  Length:616         Length:616         Length:616         Length:616
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   Kpop_freq          Latin_freq          Lofi_freq          Metal_freq
##  Length:616         Length:616         Length:616         Length:616
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Pop_freq           RnB_freq           Rap_freq           Rock_freq
##  Length:616         Length:616         Length:616         Length:616
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    VGM_freq            Anxiety         Depression        Insomnia
##  Length:616         Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  Class :character   1st Qu.: 4.000   1st Qu.: 2.000   1st Qu.: 1.000
##  Mode  :character   Median : 6.000   Median : 5.000   Median : 3.000
##                     Mean   : 5.884   Mean   : 4.894   Mean   : 3.801
##                     3rd Qu.: 8.000   3rd Qu.: 7.000   3rd Qu.: 6.000
##                     Max.   :10.000   Max.   :10.000   Max.   :10.000
##       OCD         Music_effects
##  Min.   : 0.000   Length:616
##  1st Qu.: 0.000   Class :character
##  Median : 2.000   Mode  :character
##  Mean   : 2.659
##  3rd Qu.: 5.000
##  Max.   :10.000
```

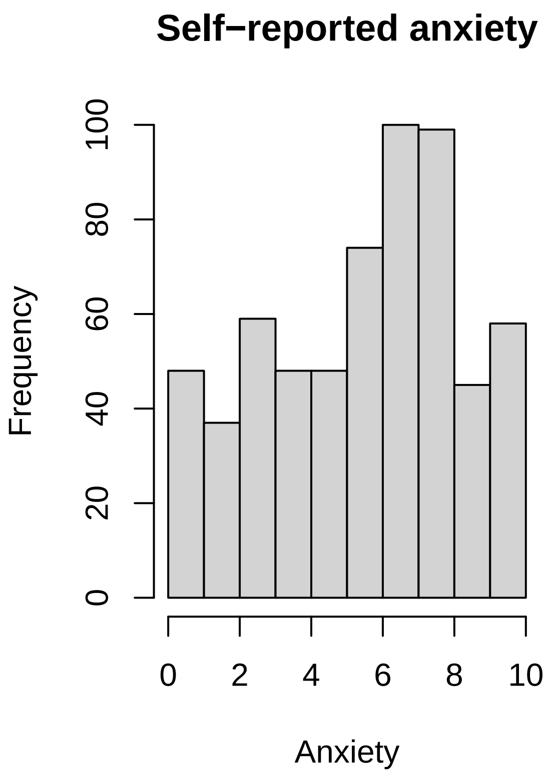
```r
# Number of rows containing NULL values in the original data
NULL_values <- colSums(is.na(mmh_data))
NULL_values
```

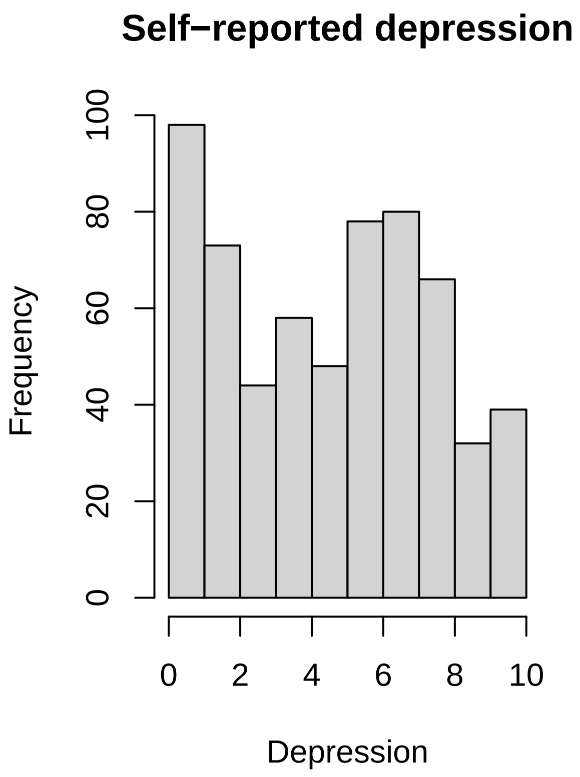
```
##                  Timestamp                          Age
##                          0                            1
##   Primary streaming service                Hours per day
##                          1                            0
##              While working               Instrumentalist
##                          3                            4
##                   Composer                    Fav genre
##                          1                            0
##                Exploratory           Foreign languages
##                          0                            4
##                        BPM        Frequency [Classical]
##                        107                            0
##        Frequency [Country]              Frequency [EDM]
##                          0                            0
##           Frequency [Folk]           Frequency [Gospel]
##                          0                            0
##        Frequency [Hip hop]             Frequency [Jazz]
##                          0                            0
##          Frequency [K pop]            Frequency [Latin]
##                          0                            0
##           Frequency [Lofi]            Frequency [Metal]
##                          0                            0
##              Frequency [Pop]              Frequency [R&B]
##                            0                            0
##              Frequency [Rap]             Frequency [Rock]
##                            0                            0
## Frequency [Video game music]                      Anxiety
##                            0                            0
##                   Depression                     Insomnia
##                            0                            0
##                          OCD                Music effects
##                            0                            8
##                  Permissions
##                            0
```

```
# EDA
library(ggplot2)
library(ggpubr)
attach(mmh)
ggplot(data = mmh, aes(x=Favourite)) + geom_bar(color="black",fill="gray") + labs(x = "Respondent's fav
```

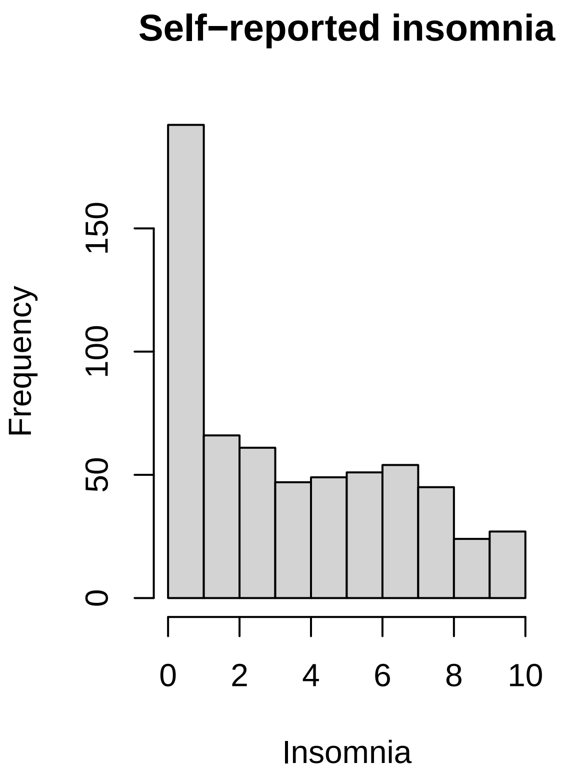


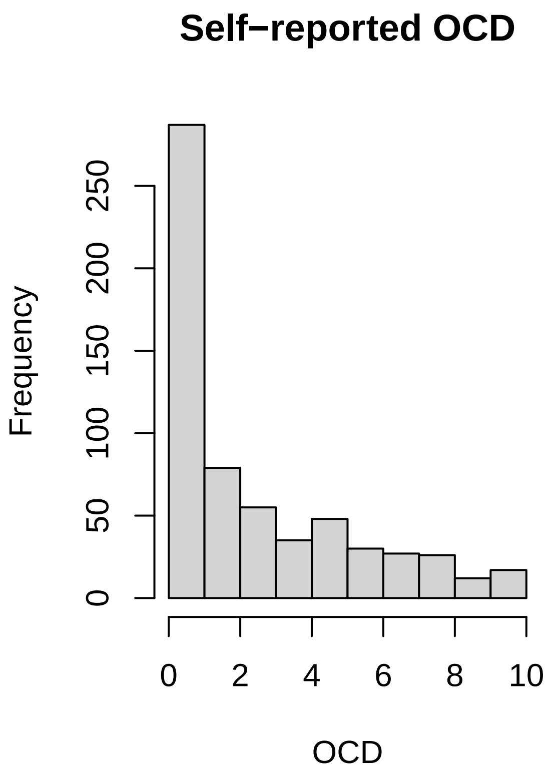
```
par(mfrow=c(1, 2))
hist(Anxiety, main = "Self-reported anxiety")
hist(Depression, main = "Self-reported depression")
```



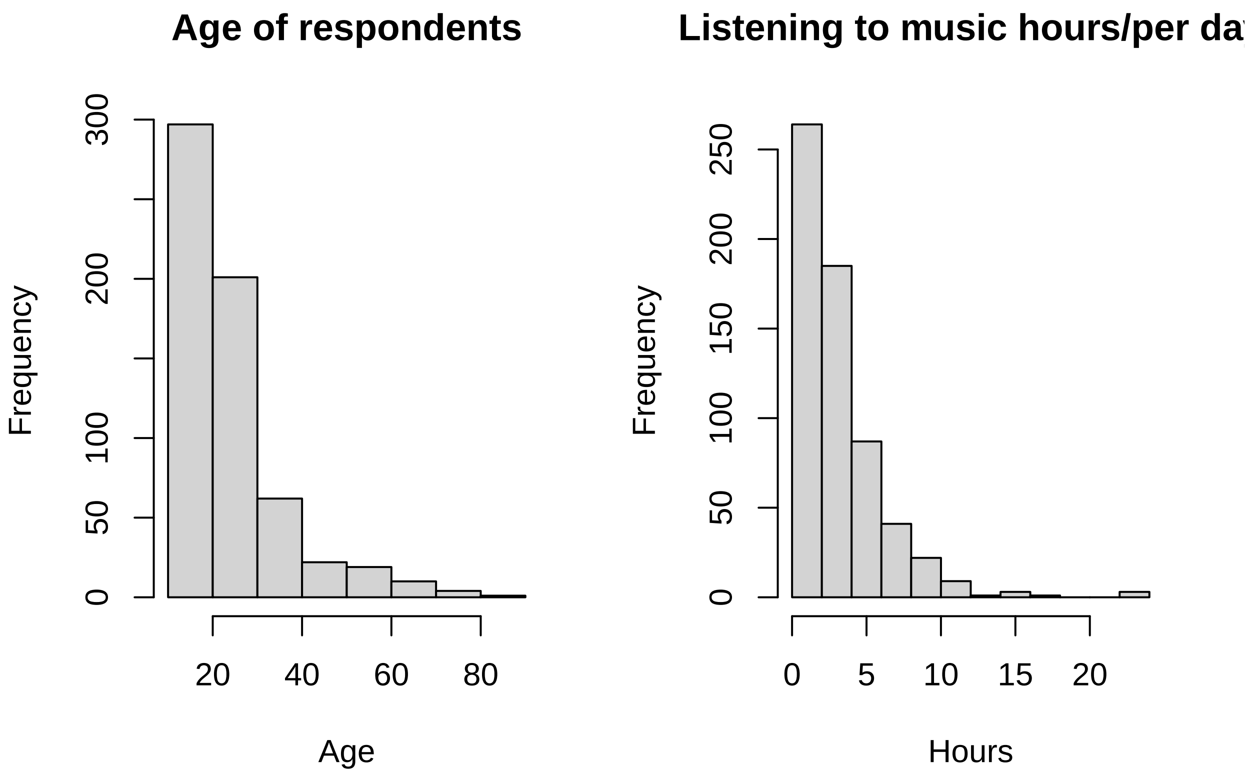


```r
par(mfrow=c(1, 2))
hist(Insomnia, main = "Self-reported insomnia")
hist(OCD, main = "Self-reported OCD")
```

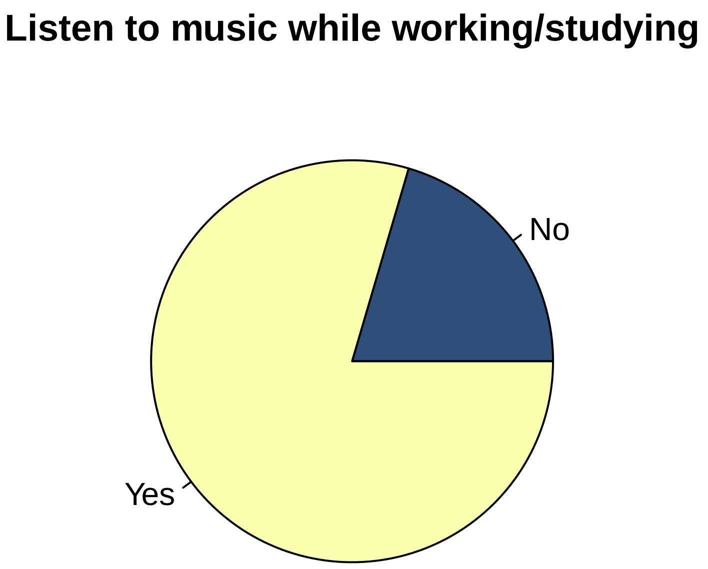




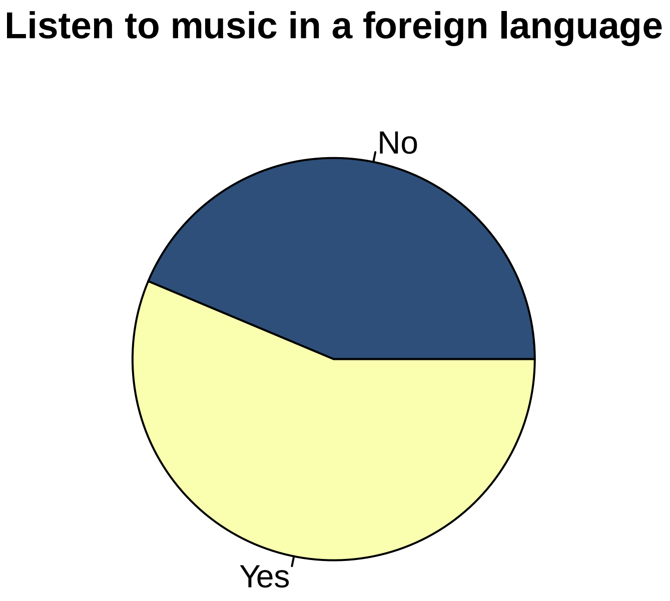
```r
par(mfrow=c(1, 2))
hist(Age, main = "Age of respondents")
hist(Hours, main = "Listening to music hours/per day")
```



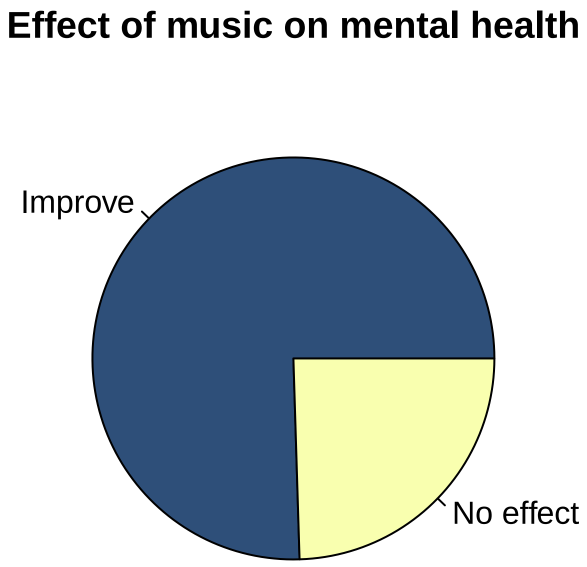
```r
par(mfrow = c(1, 1))
t1 <- table(While_working)
pie(t1, col = hcl.colors(length(t1), "BluYl"), radius = 0.85, main = "Listen to music while working/stu
```



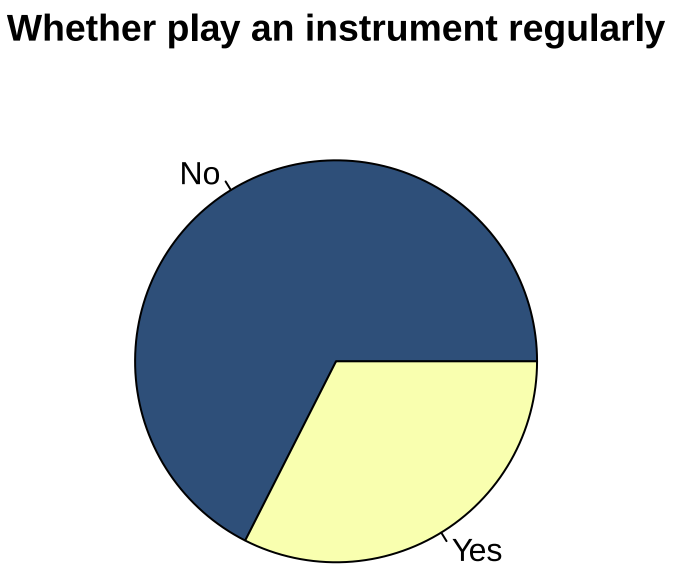
```r
par(mfrow = c(1, 1))
t2 <- table(Foreign)
pie(t2, col = hcl.colors(length(t2), "BluYl"), radius = 0.85, main = "Listen to music in a foreign langu
```



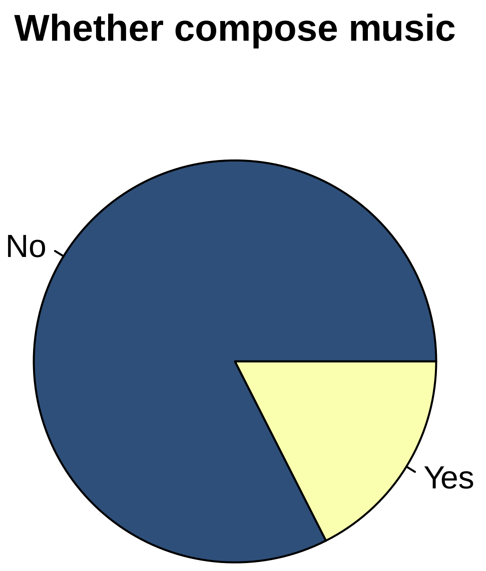
```r
par(mfrow = c(1, 1))
t3 <- table(Music_effects)
pie(t3, col = hcl.colors(length(t3), "BluYl"), radius = 0.85, main = "Effect of music on mental health")
```



```r
par(mfrow = c(1, 1))
t4 <- table(Instrumentalist)
pie(t4, col = hcl.colors(length(t4), "BluYl"), radius = 0.85, main = "Whether play an instrument regula
```

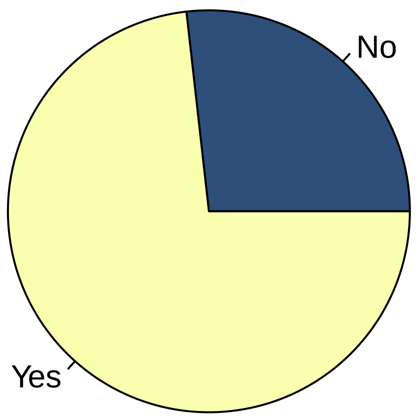


```r
par(mfrow = c(1, 1))
t5 <- table(Composer)
pie(t5, col = hcl.colors(length(t5), "BluYl"), radius = 0.85, main = "Whether compose music")
```

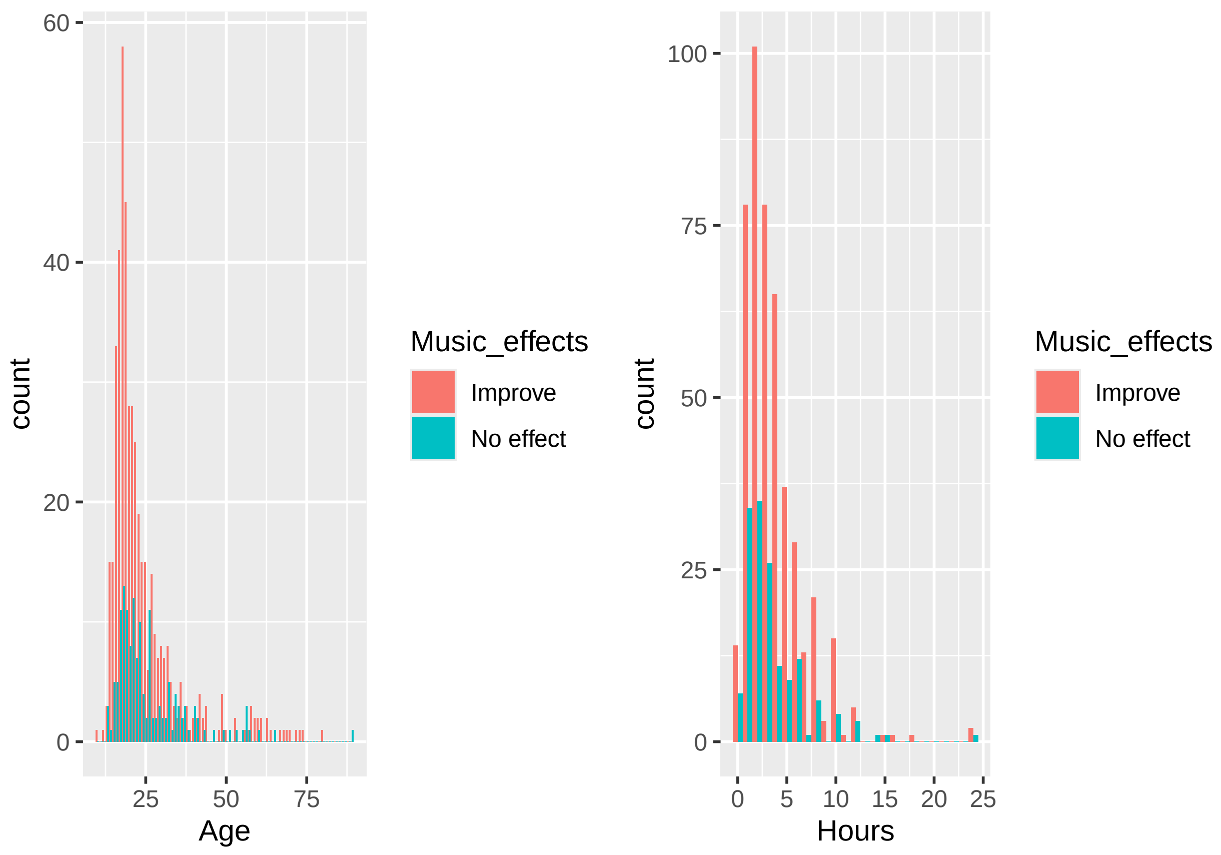


```r
par(mfrow = c(1, 1))
t6 <- table(Exploratory)
pie(t6, col = hcl.colors(length(t6), "BluYl"), radius = 0.85, main = "Whether actively explore new arti
```

Whether actively explore new artists/genres

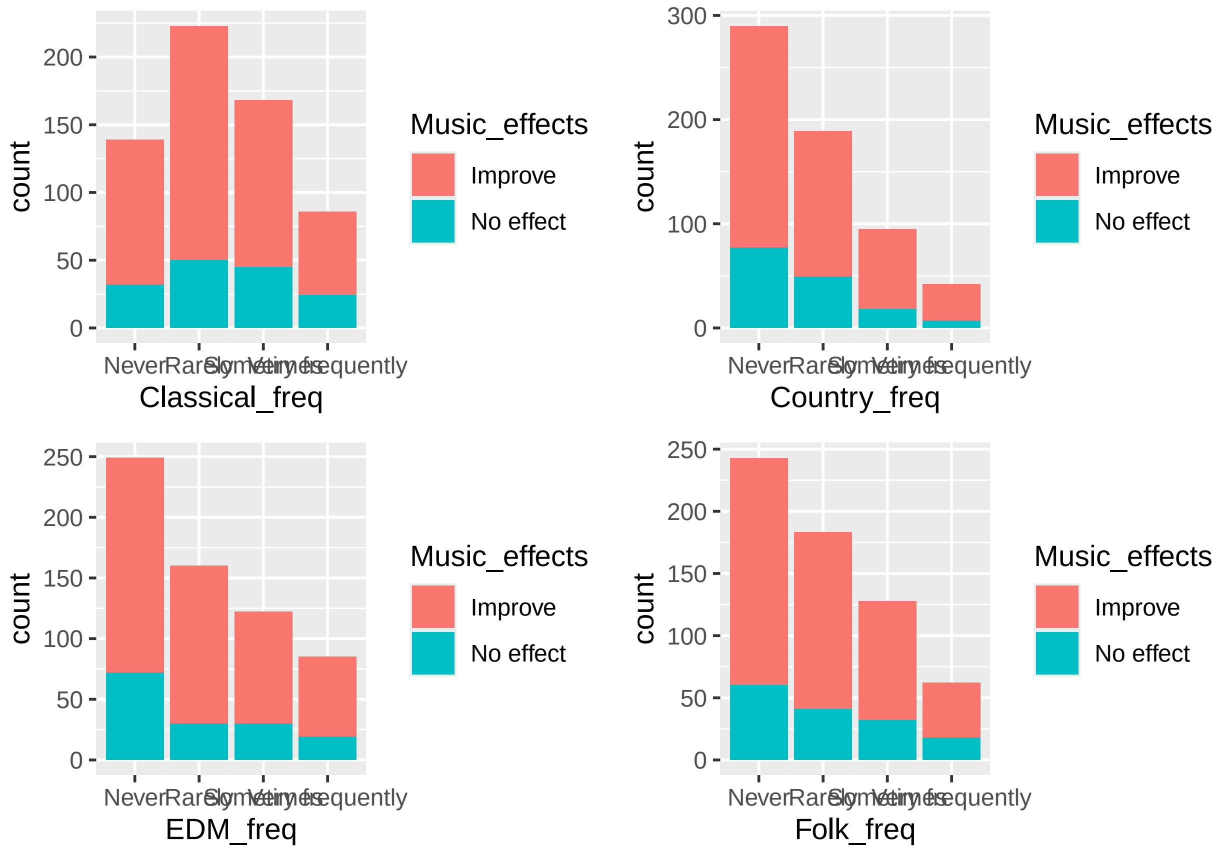


```
a = ggplot(data = mmh, aes(x=Age,fill=Music_effects)) + geom_histogram(position="dodge",binwidth=1)
b = ggplot(data = mmh, aes(x=Hours,fill=Music_effects)) + geom_histogram(position="dodge",binwidth=1)
ggarrange(a,b, ncol = 2, nrow = 1)
```

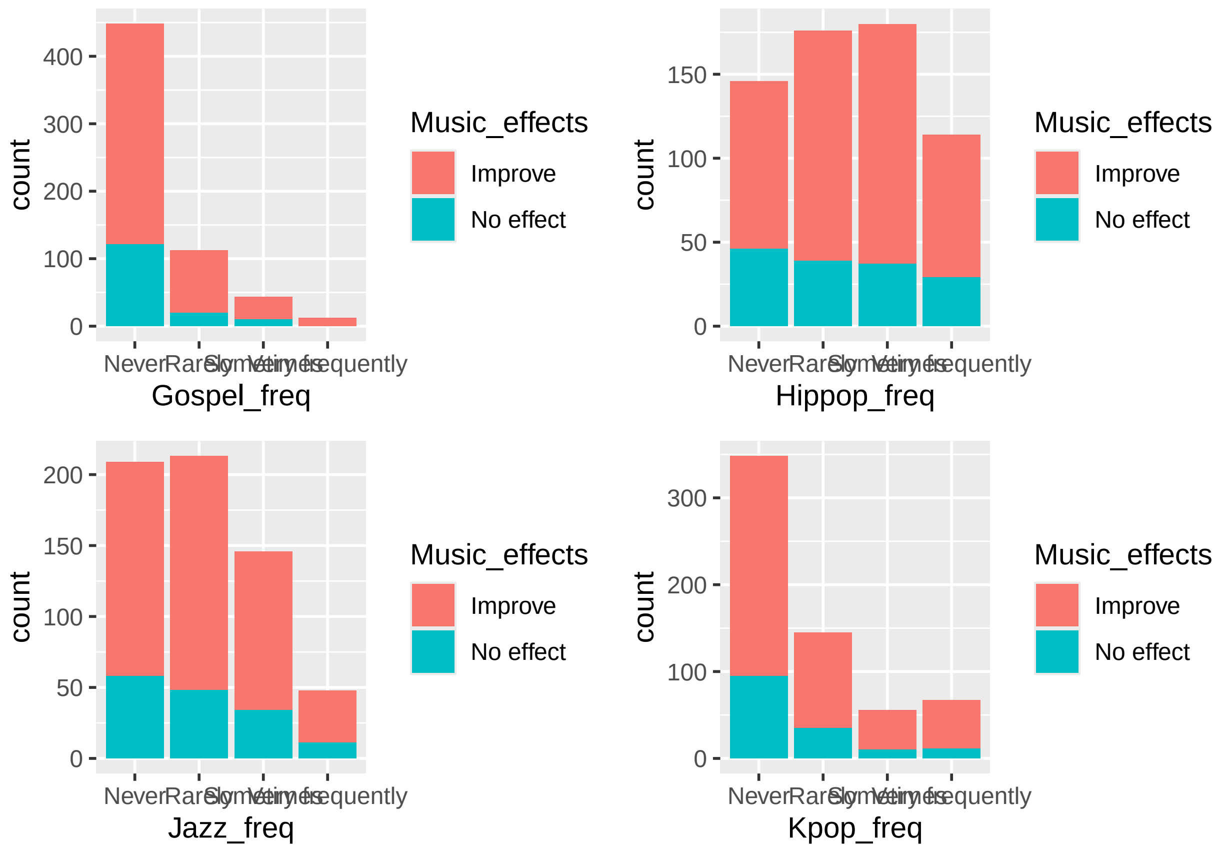


```
g1=ggplot(data=mmh, aes(x=Classical_freq, fill=Music_effects)) + geom_bar()
g2=ggplot(data=mmh, aes(x=Country_freq, fill=Music_effects)) + geom_bar()
g3=ggplot(data=mmh, aes(x=EDM_freq, fill=Music_effects)) + geom_bar()
```

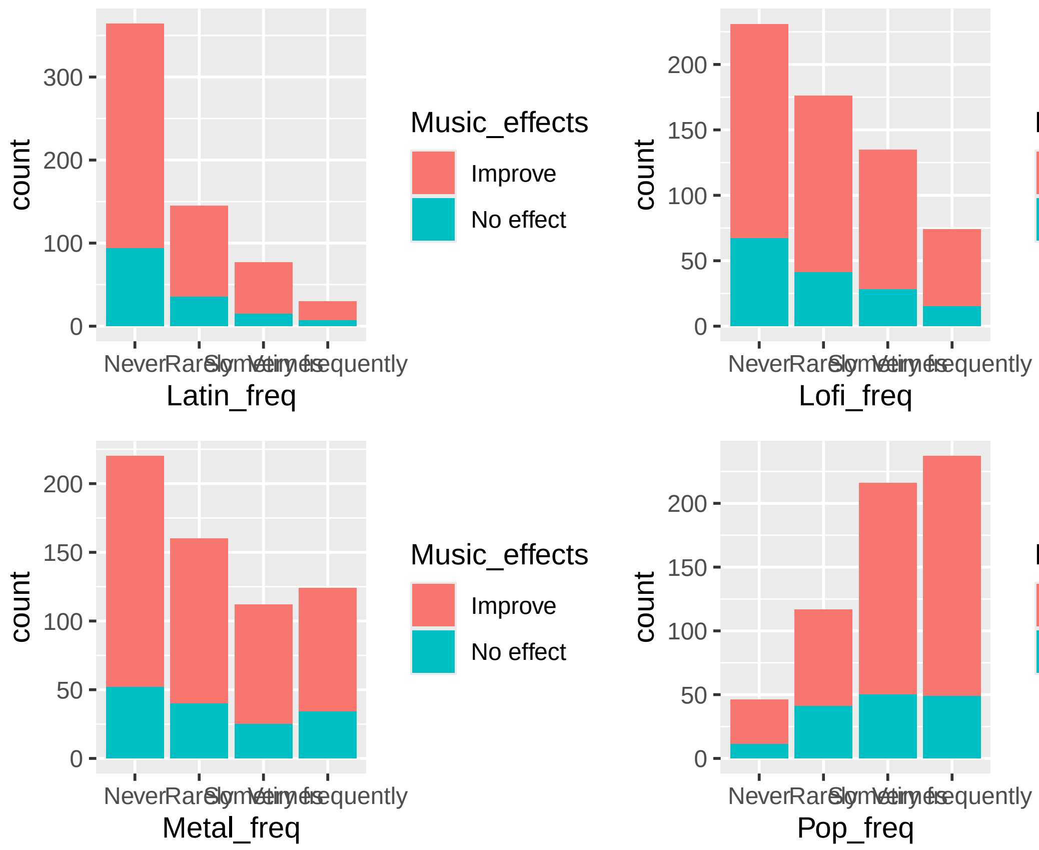
```r
g4=ggplot(data=mmh, aes(x=Folk_freq, fill=Music_effects)) + geom_bar()
ggarrange(g1,g2,g3,g4, ncol = 2, nrow = 2)
```



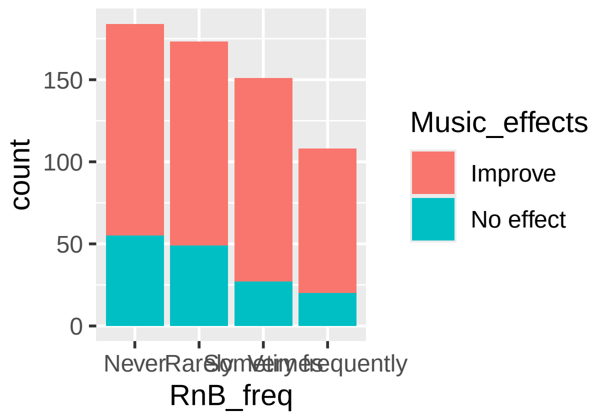
```r
g5=ggplot(data=mmh, aes(x=Gospel_freq, fill=Music_effects)) + geom_bar()
g6=ggplot(data=mmh, aes(x=Hippop_freq, fill=Music_effects)) + geom_bar()
g7=ggplot(data=mmh, aes(x=Jazz_freq, fill=Music_effects)) + geom_bar()
g8=ggplot(data=mmh, aes(x=Kpop_freq, fill=Music_effects)) + geom_bar()
ggarrange(g5,g6,g7,g8, ncol = 2, nrow = 2)
```

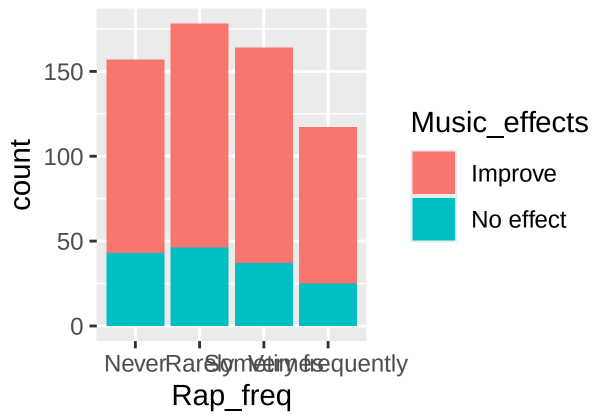


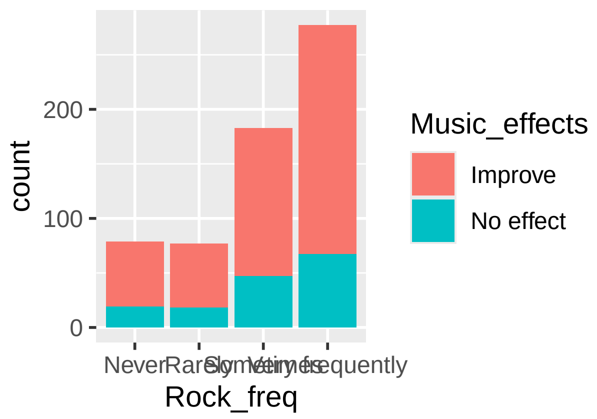
```
g9=ggplot(data=mmh, aes(x=Latin_freq, fill=Music_effects)) + geom_bar()
g10=ggplot(data=mmh, aes(x=Lofi_freq, fill=Music_effects)) + geom_bar()
g11=ggplot(data=mmh, aes(x=Metal_freq, fill=Music_effects)) + geom_bar()
g12=ggplot(data=mmh, aes(x=Pop_freq, fill=Music_effects)) + geom_bar()
ggarrange(g9,g10,g11,g12, ncol = 2, nrow = 2)
```

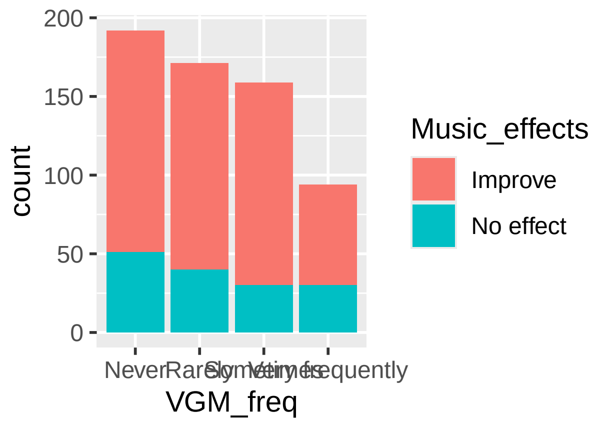


```
g13=ggplot(data=mmh, aes(x=RnB_freq, fill=Music_effects)) + geom_bar()
g14=ggplot(data=mmh, aes(x=Rap_freq, fill=Music_effects)) + geom_bar()
g15=ggplot(data=mmh, aes(x=Rock_freq, fill=Music_effects)) + geom_bar()
g16=ggplot(data=mmh, aes(x=VGM_freq, fill=Music_effects)) + geom_bar()
ggarrange(g13,g14,g15,g16, ncol = 2, nrow = 2)
```









```r
set.seed(1007377358)
# Model fitting using training dataset
model1 <- glm(as.factor(Music_effects) ~ ., data = training, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
summary(model1)
```

```
##
## Call:
## glm(formula = as.factor(Music_effects) ~ ., family = "binomial",
##     data = training)
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   2.800e-01  1.068e+00   0.262  0.79323
## Age                          -1.829e-03  1.453e-02  -0.126  0.89984
## Hours                         4.270e-02  4.956e-02   0.862  0.38889
## While_workingYes             -7.544e-01  3.286e-01  -2.296  0.02170 *
## InstrumentalistYes           -7.875e-01  3.695e-01  -2.131  0.03307 *
## ComposerYes                  -3.453e-01  4.482e-01  -0.770  0.44102
## FavouriteCountry              1.111e+00  1.457e+00   0.762  0.44584
## FavouriteEDM                 -5.254e-01  1.094e+00  -0.480  0.63110
## FavouriteFolk                 8.072e-01  1.111e+00   0.726  0.46768
## FavouriteGospel               1.233e+00  2.341e+03   0.001  0.99958
## FavouriteHip hop             -2.065e+00  1.228e+00  -1.681  0.09276 .
## FavouriteJazz                -6.493e-01  1.351e+00  -0.481  0.63082
## FavouriteK pop                1.009e+00  1.209e+00   0.835  0.40373
## FavouriteLatin                1.679e+01  3.956e+03   0.004  0.99661
## FavouriteLofi                -1.575e+01  1.231e+03  -0.013  0.98979
## FavouriteMetal                3.666e-01  9.608e-01   0.382  0.70282
## FavouritePop                  1.394e-01  9.036e-01   0.154  0.87742
## FavouriteR&B                 -1.267e-01  1.099e+00  -0.115  0.90819
## FavouriteRap                 -1.050e+00  1.105e+00  -0.950  0.34209
## FavouriteRock                 2.963e-01  8.620e-01   0.344  0.73105
## FavouriteVideo game music     1.370e+00  9.140e-01   1.498  0.13405
## ExploratoryYes               -9.358e-01  3.392e-01  -2.759  0.00580 **
## ForeignYes                    1.148e-01  3.197e-01   0.359  0.71939
## BPM                           1.739e-08  3.549e-06   0.005  0.99609
## Classical_freqRarely          2.230e-01  3.977e-01   0.561  0.57501
## Classical_freqSometimes       7.800e-01  4.255e-01   1.833  0.06679 .
## Classical_freqVery frequently 9.396e-01  6.099e-01   1.541  0.12342
## Country_freqRarely            2.433e-01  3.340e-01   0.728  0.46646
## Country_freqSometimes        -5.923e-01  4.590e-01  -1.291  0.19683
## Country_freqVery frequently  -1.326e+00  1.035e+00  -1.281  0.20002
## EDM_freqRarely               -4.758e-01  3.688e-01  -1.290  0.19694
## EDM_freqSometimes            -8.649e-03  3.995e-01  -0.022  0.98273
## EDM_freqVery frequently       1.225e-02  6.061e-01   0.020  0.98387
## Folk_freqRarely              -1.087e-02  3.844e-01  -0.028  0.97744
## Folk_freqSometimes            6.731e-01  4.405e-01   1.528  0.12650
## Folk_freqVery frequently      2.753e-01  5.836e-01   0.472  0.63712
## Gospel_freqRarely            -7.331e-01  4.027e-01  -1.821  0.06864 .
## Gospel_freqSometimes         -5.928e-01  6.556e-01  -0.904  0.36585
## Gospel_freqVery frequently   -1.701e+01  1.448e+03  -0.012  0.99063
## Hippop_freqRarely            -5.360e-01  4.630e-01  -1.158  0.24701
## Hippop_freqSometimes          1.824e-01  5.204e-01   0.351  0.72595
## Hippop_freqVery frequently    1.101e+00  7.165e-01   1.536  0.12453
## Jazz_freqRarely              -3.083e-01  3.455e-01  -0.892  0.37226
## Jazz_freqSometimes            7.493e-03  4.373e-01   0.017  0.98633
## Jazz_freqVery frequently      5.893e-01  7.982e-01   0.738  0.46029
## Kpop_freqRarely              -2.337e-01  3.689e-01  -0.634  0.52637
## Kpop_freqSometimes           -5.862e-01  5.778e-01  -1.014  0.31035
## Kpop_freqVery frequently     -6.596e-01  6.420e-01  -1.027  0.30422
## Latin_freqRarely             -3.902e-01  3.984e-01  -0.979  0.32733
## Latin_freqSometimes          -3.349e-01  4.668e-01  -0.717  0.47310
## Latin_freqVery frequently     8.896e-01  7.475e-01   1.190  0.23399
## Lofi_freqRarely               1.622e-01  3.572e-01   0.454  0.64975
## Lofi_freqSometimes            2.520e-01  4.226e-01   0.596  0.55103
## Lofi_freqVery frequently     -4.980e-01  5.445e-01  -0.915  0.36041
## Metal_freqRarely             -4.321e-02  3.880e-01  -0.111  0.91133
## Metal_freqSometimes          -7.070e-01  4.845e-01  -1.459  0.14446
## Metal_freqVery frequently     7.867e-03  5.548e-01   0.014  0.98869
## Pop_freqRarely                1.031e+00  5.855e-01   1.761  0.07817 .
## Pop_freqSometimes             3.255e-01  6.133e-01   0.531  0.59564
## Pop_freqVery frequently       4.257e-01  6.643e-01   0.641  0.52165
## RnB_freqRarely               -6.487e-02  3.973e-01  -0.163  0.87030
## RnB_freqSometimes            -4.957e-01  4.726e-01  -1.049  0.29425
## RnB_freqVery frequently      -8.335e-01  6.417e-01  -1.299  0.19394
## Rap_freqRarely                6.429e-01  4.445e-01   1.446  0.14808
## Rap_freqSometimes              6.917e-01  5.240e-01   1.320  0.18683
## Rap_freqVery frequently        3.961e-01  7.357e-01   0.538  0.59029
## Rock_freqRarely                4.005e-01  5.942e-01   0.674  0.50027
## Rock_freqSometimes             4.310e-01  5.562e-01   0.775  0.43844
## Rock_freqVery frequently       4.084e-01  6.017e-01   0.679  0.49726
## VGM_freqRarely                -2.333e-01  3.685e-01  -0.633  0.52659
## VGM_freqSometimes             -4.798e-01  4.239e-01  -1.132  0.25776
## VGM_freqVery frequently        1.541e-01  5.196e-01   0.297  0.76680
## Anxiety                       -2.104e-01  6.406e-02  -3.285  0.00102 **
## Depression                     3.225e-02  5.433e-02   0.594  0.55282
## Insomnia                      -1.079e-02  4.871e-02  -0.222  0.82460
## OCD                            1.304e-02  5.078e-02   0.257  0.79738
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 522.87  on 461  degrees of freedom
## Residual deviance: 399.33  on 386  degrees of freedom
## AIC: 551.33
##
## Number of Fisher Scoring iterations: 16
```

```
# Use AIC to get a reduced model with only significant variables
sel.var.aic <- step(model1, trace = 0, k = 2, direction = "both")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
select_var_aic <- attr(terms(sel.var.aic), "term.labels")
select_var_aic
```

```
## [1] "While_working"   "Instrumentalist" "Exploratory"     "BPM"
## [5] "Classical_freq"  "Gospel_freq"     "Pop_freq"        "Anxiety"
```

```r
# Fit a new model with the variable selected by AIC method
reduced.model.aic <- glm(as.factor(Music_effects) ~ While_working + Composer + Exploratory + EDM_freq +
                     + Pop_freq + Anxiety + Depression + Insomnia, data = training, family = "binomial")
summary(reduced.model.aic)
```

```
##
## Call:
## glm(formula = as.factor(Music_effects) ~ While_working + Composer +
##     Exploratory + EDM_freq + Gospel_freq + Pop_freq + Anxiety +
##     Depression + Insomnia, family = "binomial", data = training)
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  0.659257   0.505499   1.304 0.192175
## While_workingYes            -0.725224   0.274564  -2.641 0.008257 **
## ComposerYes                 -0.664774   0.348670  -1.907 0.056572 .
## ExploratoryYes              -0.628479   0.260099  -2.416 0.015679 *
## EDM_freqRarely              -0.325983   0.302533  -1.078 0.281251
## EDM_freqSometimes            0.005259   0.301783   0.017 0.986096
## EDM_freqVery frequently     -0.302770   0.391448  -0.773 0.439248
## Gospel_freqRarely           -0.686550   0.339473  -2.022 0.043135 *
## Gospel_freqSometimes        -0.584170   0.508460  -1.149 0.250597
## Gospel_freqVery frequently -15.648673 725.365484  -0.022 0.982788
## Pop_freqRarely               0.999394   0.496778   2.012 0.044246 *
## Pop_freqSometimes            0.255359   0.486076   0.525 0.599341
## Pop_freqVery frequently      0.222439   0.489936   0.454 0.649817
## Anxiety                     -0.180522   0.049768  -3.627 0.000286 ***
## Depression                   0.043571   0.045755   0.952 0.340954
## Insomnia                     0.020182   0.040183   0.502 0.615491
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 522.87  on 461  degrees of freedom
## Residual deviance: 463.40  on 446  degrees of freedom
## AIC: 495.4
##
## Number of Fisher Scoring iterations: 15
```

```r
# Use BIC to get a reduced model with only significant variables
sel.var.bic <- step(model1, trace = 0, k = log(nrow(training)), direction = "both")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
select_var_bic <- attr(terms(sel.var.bic), "term.labels")
select_var_bic
```

```
## [1] "While_working" "Exploratory"   "Anxiety"
```

```
# Fit a new model with the variable selected by BIC method
reduced.model.bic <- glm(as.factor(Music_effects) ~ While_working + Anxiety, data = training, family =
summary(reduced.model.bic)
```

```
##
## Call:
## glm(formula = as.factor(Music_effects) ~ While_working + Anxiety,
##     family = "binomial", data = training)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)       0.38808    0.31353   1.238 0.215806
## While_workingYes -0.88283    0.25329  -3.485 0.000491 ***
## Anxiety          -0.14052    0.03955  -3.553 0.000381 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 522.87  on 461  degrees of freedom
## Residual deviance: 498.73  on 459  degrees of freedom
## AIC: 504.73
##
## Number of Fisher Scoring iterations: 4
```

```
# LASSO
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
## Loaded glmnet 4.1-8
```

```
x <- as.matrix(training[,-which(colnames(training) == "Music_effects")]) # predictors matrix
y <- training$Music_effects

cv.out = cv.glmnet(x, y, family = "binomial", type.measure = "class", alpha = 0.5)
```

```
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
```

```
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
## Warning in storage.mode(xd) <- "double": NAs introduced by coercion
```

```
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
## Warning in cbind2(1, newx) %*% nbeta: NAs introduced by coercion
```

```
best.lambda <- cv.out$lambda.1se
co <- coef(cv.out, s = "lambda.1se")

thresh <- 0.00
# select variables #
inds <- which(abs(co) > thresh )
variables <- row.names(co)[inds]
sel.var.lasso <- variables[!(variables %in% '(Intercept)')]
sel.var.lasso
```

```
## character(0)
```

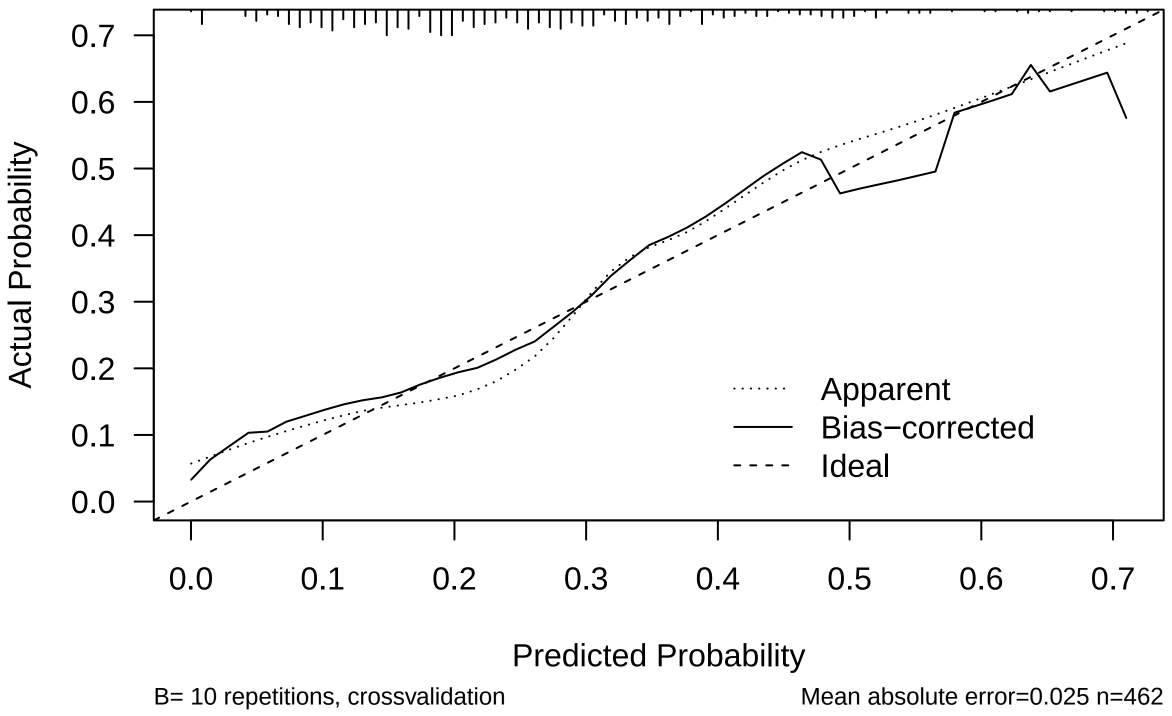
```
# No variables are selected by LASSO
```

```
# For model selected by AIC
# Cross Validation
library(rms)
```

```
## Loading required package: Hmisc
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
## Fit the model with lrm from rms package ##
lrm.final.aic <- lrm(as.factor(Music_effects) ~ While_working + Composer + Exploratory + EDM_freq + Gosp
                  + Pop_freq + Anxiety + Depression + Insomnia, data = training, x = TRUE, y = TRUE, mode
cross.calib <- calibrate(lrm.final.aic, method="crossvalidation", B=10) # model calibration
plot(cross.calib, las=1, xlab = "Predicted Probability")
```



```
##
## n=462   Mean absolute error=0.025   Mean squared error=0.00098
## 0.9 Quantile of absolute error=0.047
```

```
# Discrimination with ROC curve
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
p <- predict(lrm.final.aic, type = "fitted")
```

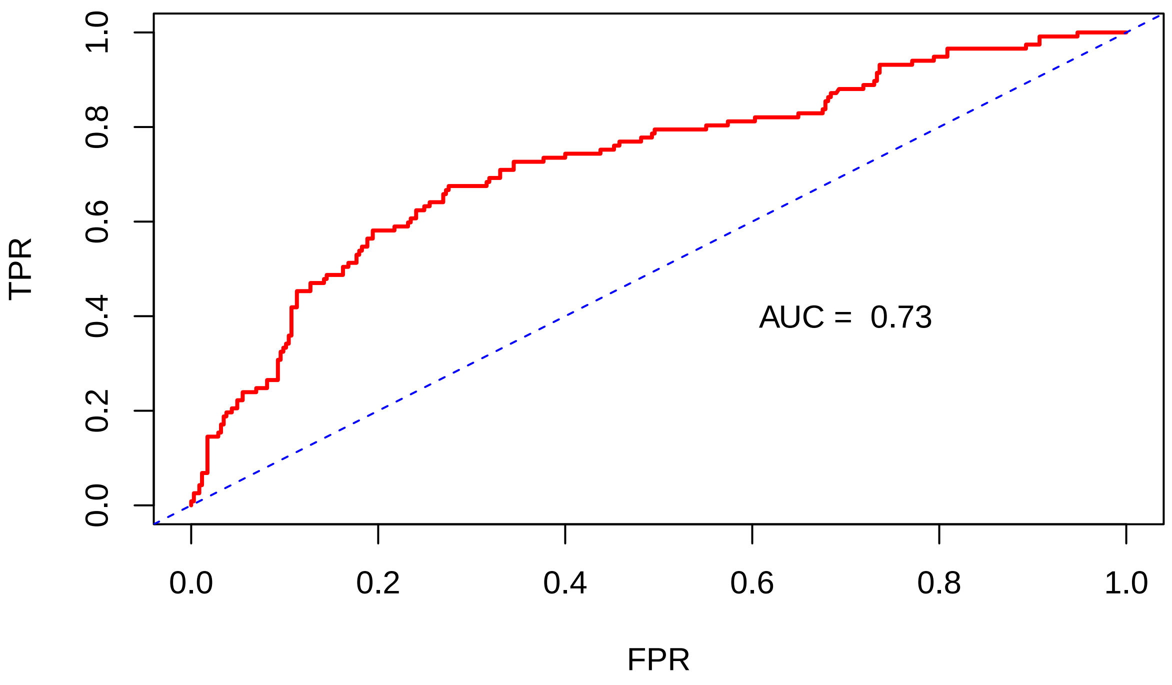
```
## Warning in formula.character(object, env = baseenv()): Using formula(x) is deprecated when x is a cha
##   Consider formula(paste(x, collapse = " ")) instead.
```

```
roc_logit <- roc(training$Music_effects ~ p)
```

```
## Setting levels: control = Improve, case = No effect
```

```
## Setting direction: controls < cases
```

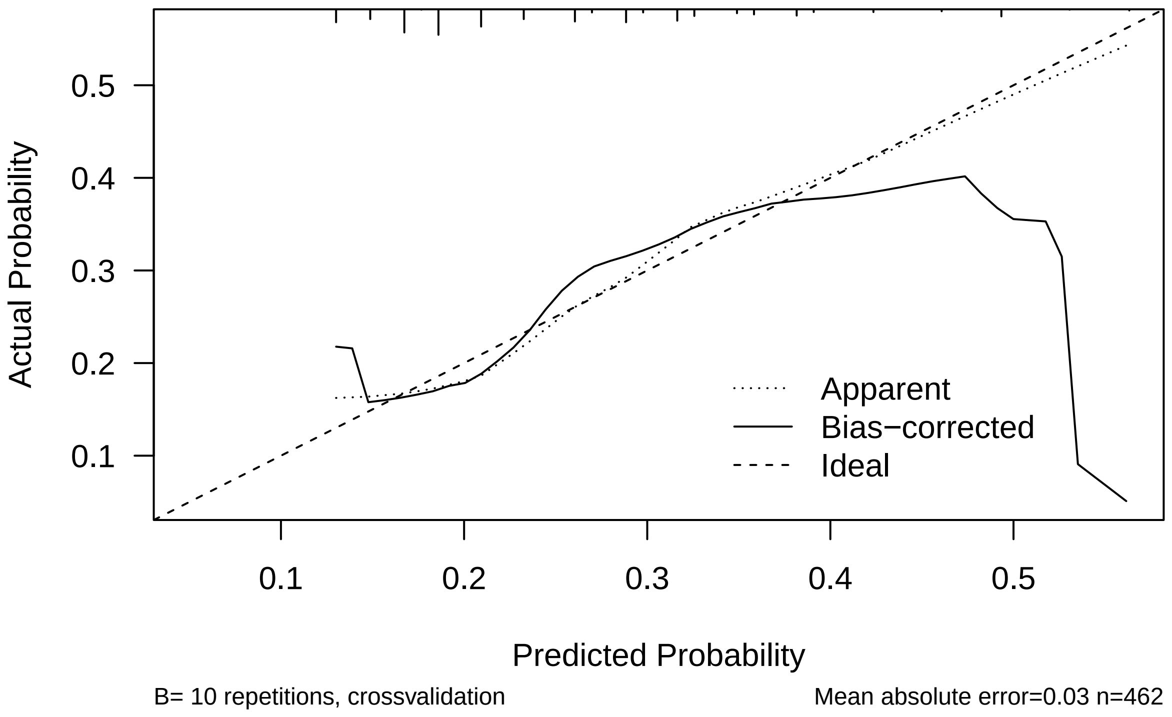
```
## The True Positive Rate ##
TPR <- roc_logit$sensitivities
## The False Positive Rate ##
FPR <- 1 - roc_logit$specificities

plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col = 'red')
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit),2)))
```



```
auc(roc_logit)
```

```
## Area under the curve: 0.729
```

```r
# For model selected by BIC
# Cross Validation
library(rms)
## Fit the model with lrm from rms package ##
lrm.final.bic <- lrm(as.factor(Music_effects) ~ While_working + Anxiety, data = training, x = TRUE, y =
cross.calib <- calibrate(lrm.final.bic, method="crossvalidation", B=10) # model calibration
plot(cross.calib, las=1, xlab = "Predicted Probability")
```

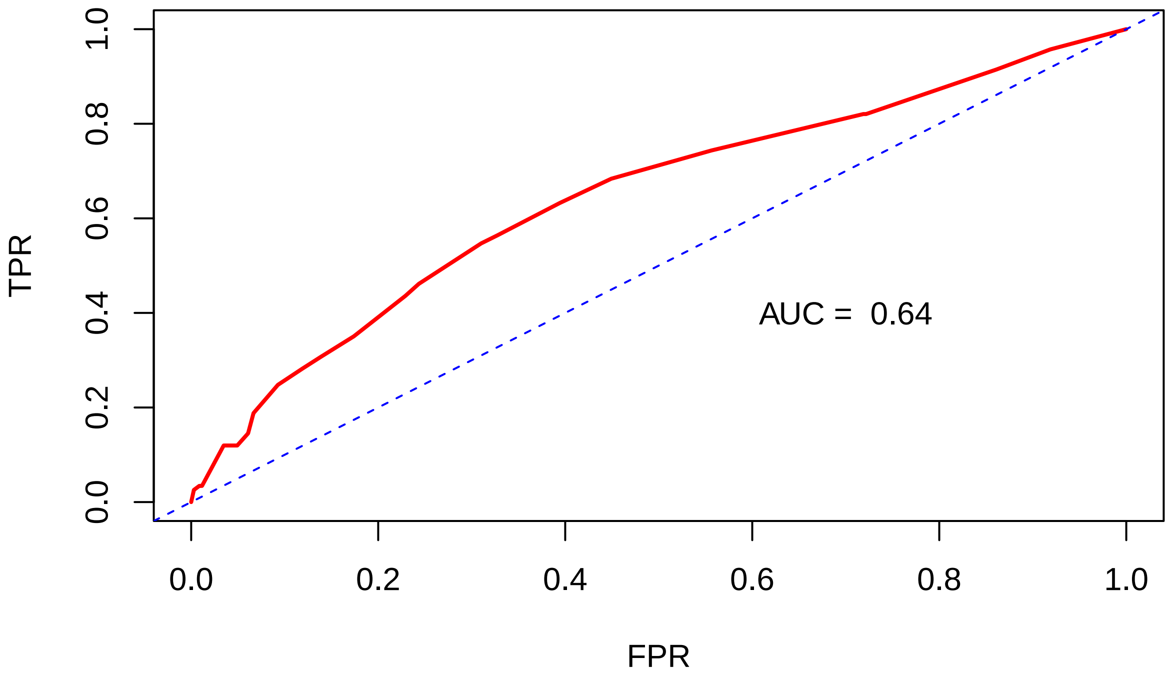


```
##
## n=462   Mean absolute error=0.03   Mean squared error=0.00332
## 0.9 Quantile of absolute error=0.088
```

```r
# Discrimination with ROC curve
library(pROC)
p <- predict(lrm.final.bic, type = "fitted")
roc_logit <- roc(training$Music_effects ~ p)
```

```
## Setting levels: control = Improve, case = No effect
## Setting direction: controls < cases
```

```r
## The True Positive Rate ##
TPR <- roc_logit$sensitivities
## The False Positive Rate ##
FPR <- 1 - roc_logit$specificities

plot(FPR, TPR, xlim = c(0,1), ylim = c(0,1), type = 'l', lty = 1, lwd = 2,col = 'red')
abline(a = 0, b = 1, lty = 2, col = 'blue')
text(0.7,0.4,label = paste("AUC = ", round(auc(roc_logit),2)))
```



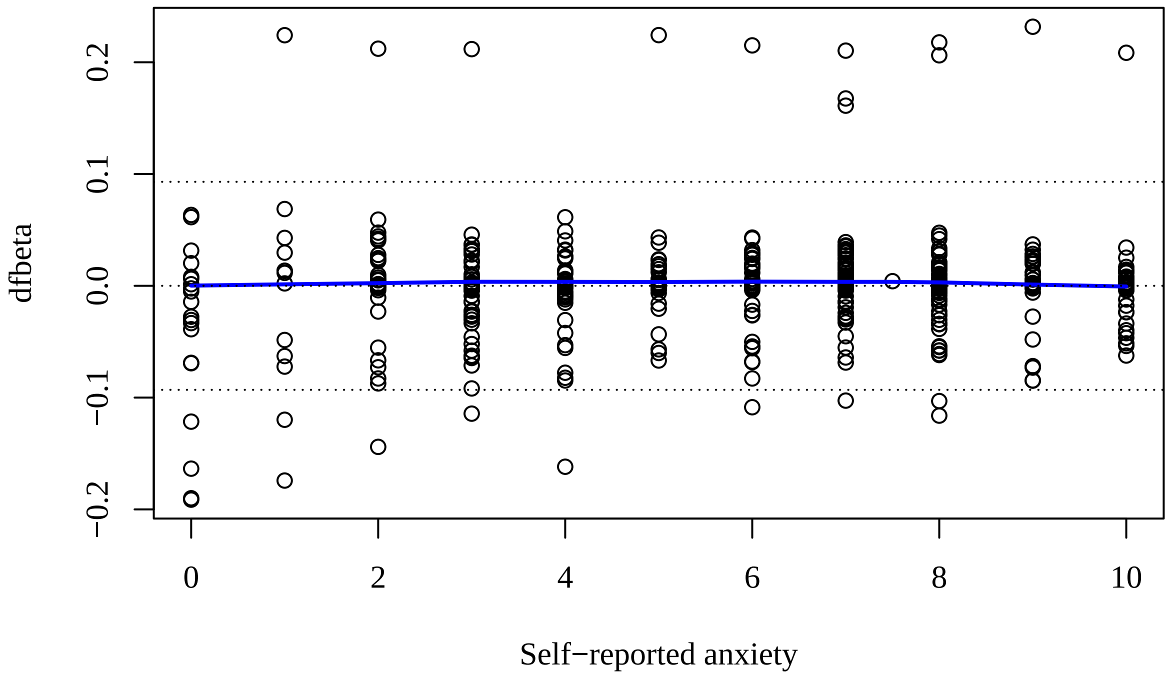
```
auc(roc_logit)
```

```
## Area under the curve: 0.6442
```

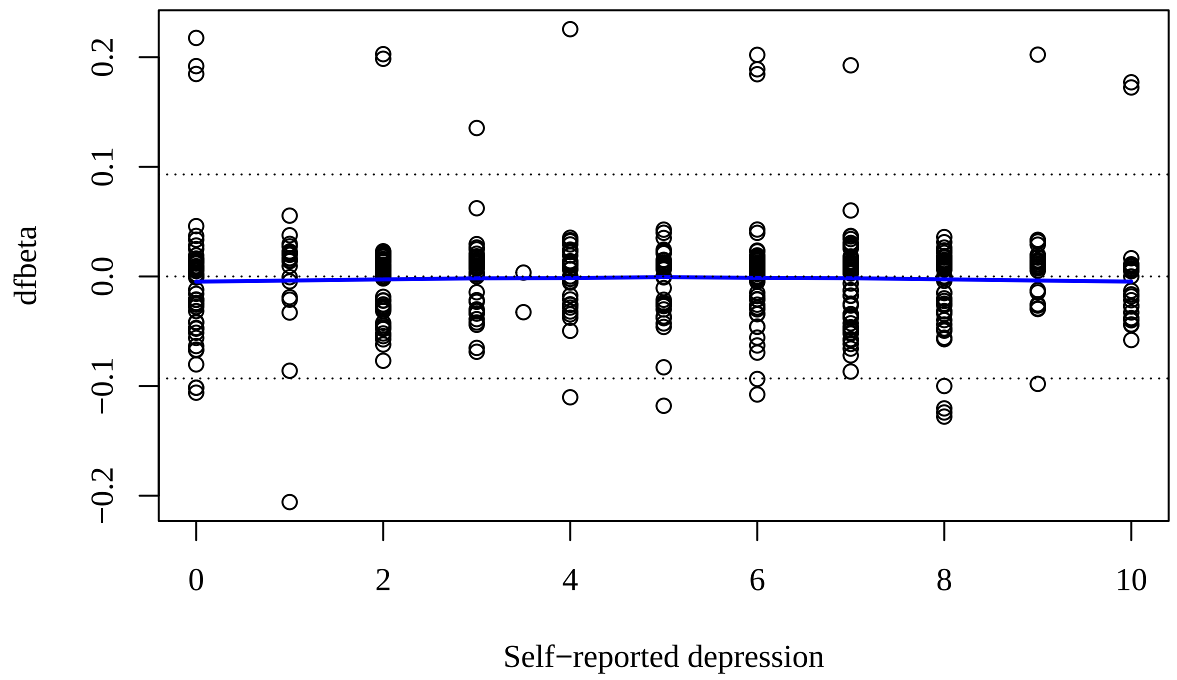
```
# Dfbetas
log.mod.final <- glm(as.factor(Music_effects) ~ While_working + Composer + Exploratory + EDM_freq + Gosp
                     + Pop_freq + Anxiety + Depression + Insomnia, data=training,family="binomial")
df.final <- dfbetas(log.mod.final)
head(df.final)
```

```
##    (Intercept) While_workingYes  ComposerYes ExploratoryYes EDM_freqRarely
## 1  0.009234249      -0.10051638 -0.033222020    -0.07297795    0.001995082
## 2  0.053788226      -0.13659316 -0.019846721    -0.11691083    0.007101832
## 3  0.011647006      -0.01145506  0.005880764    -0.01191007   -0.033470960
## 4 -0.017017357      -0.02330994  0.009480114    -0.02192363    0.037809064
## 5  0.012426386       0.02720594 -0.029704436     0.04532954    0.004176274
## 6  0.018529234      -0.02802083  0.023301278    -0.01748294    0.046229634
##   EDM_freqSometimes EDM_freqVery frequently Gospel_freqRarely
## 1       0.007770666            0.1611525920      -0.046279193
## 2       0.147085561           -0.0006807627       0.202148806
## 3       0.001618019            0.0010310878       0.008839664
## 4       0.046395409            0.0316320247       0.017945197
## 5      -0.001622083            0.2121359398      -0.054405891
## 6       0.042620454            0.0328358795       0.011437960
##   Gospel_freqSometimes Gospel_freqVery frequently Pop_freqRarely
## 1          0.002380642              -1.861104e-05    0.072812820
## 2          0.019344087               1.104817e-06    0.005867460
## 3          0.005939953               7.675705e-06    0.007615783
## 4          0.018903573               4.801178e-06   -0.005516797
## 5         -0.036237740              -3.083159e-05    0.010990284
## 6          0.012188303               7.450060e-06    0.004708244
##   Pop_freqSometimes Pop_freqVery frequently     Anxiety   Depression
## 1       0.014506271             0.030105881 -0.01202648  0.009812301
## 2       0.008016471             0.056355825 -0.01668982  0.054435641
## 3      -0.008538325             0.007958566 -0.01747853 -0.010557223
## 4      -0.005440801            -0.033451947  0.02288234  0.021612419
## 5       0.062966317             0.004743541 -0.05424033 -0.007431798
## 6      -0.020635068             0.005497301  0.01448541 -0.034676442
##       Insomnia
## 1  0.108835115
## 2 -0.050874252
## 3  0.024534881
## 4 -0.001312367
## 5 -0.075195613
## 6 -0.061126942
```

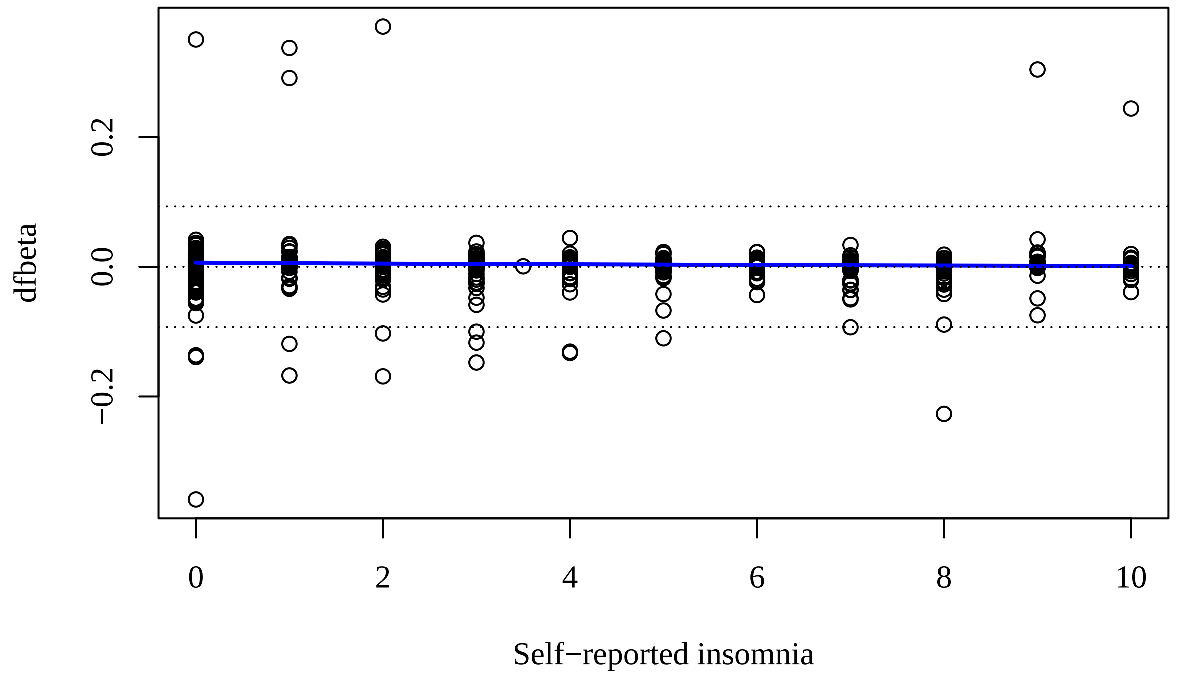
```r
par(family = 'serif')
plot(training$Anxiety, df.final[,7], xlab='Self-reported anxiety', ylab='dfbeta')
lines(lowess(training$Anxiety, df.final[,7]), lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.final)), lty='dotted')
abline(h=2/sqrt(nrow(df.final)), lty='dotted')
```



```r
plot(training$Depression, df.final[,8], xlab='Self-reported depression', ylab='dfbeta')
lines(lowess(training$Depression, df.final[,8]), lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.final)), lty='dotted')
abline(h=2/sqrt(nrow(df.final)), lty='dotted')
```

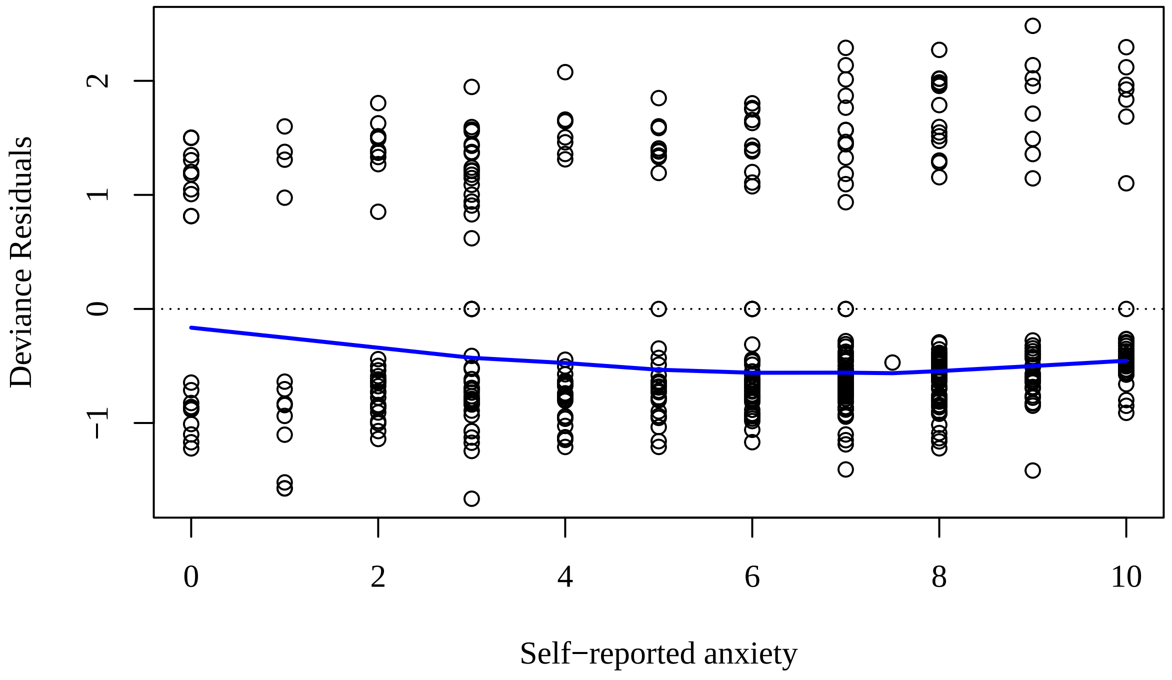


```r
plot(training$Insomnia, df.final[,9], xlab='Self-reported insomnia', ylab='dfbeta')
lines(lowess(training$Insomnia, df.final[,9]), lwd=2, col='blue')
abline(h=0, lty='dotted')
abline(h=-2/sqrt(nrow(df.final)), lty='dotted')
abline(h=2/sqrt(nrow(df.final)), lty='dotted')
```

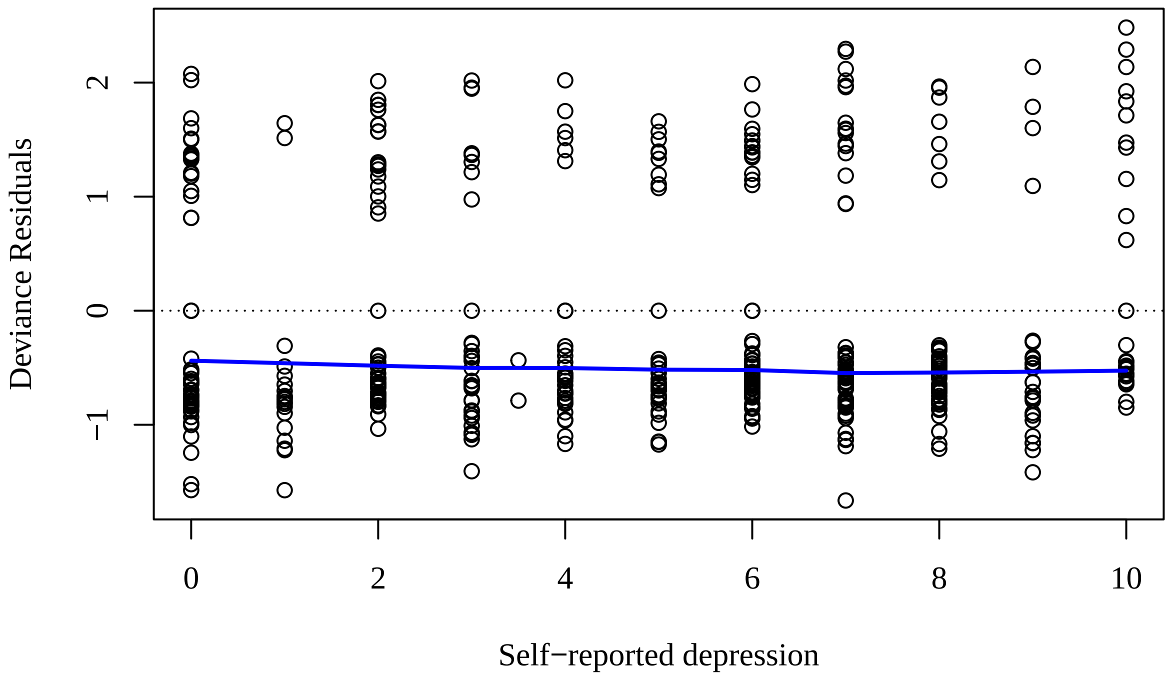


```r
# Plot the deviance residuals
res.dev <- residuals(log.mod.final, type = "deviance")
par(family = 'serif')
plot(training$Anxiety, res.dev, xlab='Self-reported anxiety', ylab='Deviance Residuals')
```

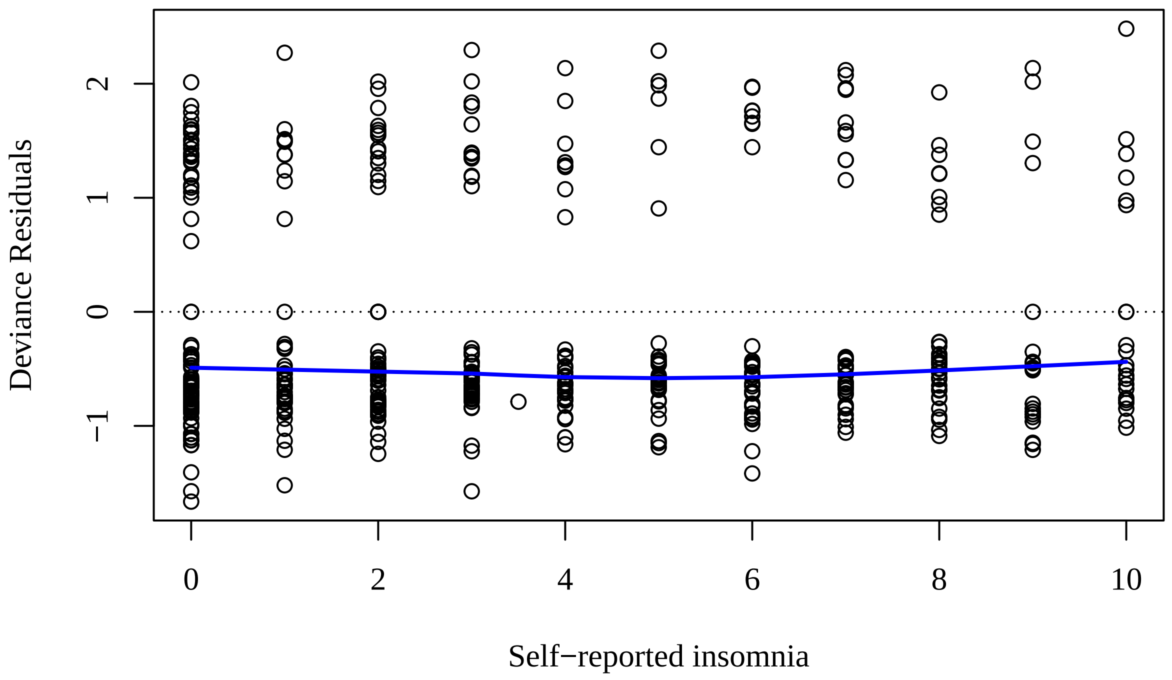
```
lines(lowess(training$Anxiety, res.dev), lwd=2, col='blue')
abline(h=0, lty='dotted')
```



```
plot(training$Depression, res.dev, xlab='Self-reported depression', ylab='Deviance Residuals')
lines(lowess(training$Depression, res.dev), lwd=2, col='blue')
abline(h=0, lty='dotted')
```



```r
plot(training$Insomnia, res.dev, xlab='Self-reported insomnia', ylab='Deviance Residuals')
lines(lowess(training$Insomnia, res.dev), lwd=2, col='blue')
abline(h=0, lty='dotted')
```



```r
# Test set prediction
n= nrow(test)
test$Music_effects <- ifelse(test$Music_effects == "Improve", 1, 0)
pre.prob = predict(reduced.model.aic, test, type="response")
pre.prob <- ifelse(pre.prob < 0.5, 0, 1)

# Calculate the number of correct predictions
correct_predictions = sum(test$Music_effects == pre.prob)

# Calculate accuracy
accuracy = correct_predictions / n
accuracy
```

```
## [1] 0.2532468
```