Does Music affect your mental health?

2023-03-27

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   1.0.1
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## 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(ggplot2)
library(dplyr)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
##
## The following object is masked from 'package:purrr':
##
##     set_names
##
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(stringr)
library(tidyr)
library(knitr)
library(tidyverse)
library(tidyr)
library(readxl)
library(readr)
library(broom)
library(MASS)
```

```
##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(plyr)
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
##
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
data <- read_csv('/Users/sushmithakeerthy/Documents/Working Folder/Indiana University/Semester 4/EDA/mxm
```

```
## 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.
```

```
summary(data)
```

```
##   Timestamp              Age        Primary streaming service Hours per day
##  Length:736         Min.   :10.00   Length:736                Min.   : 0.000
##  Class :character   1st Qu.:18.00   Class :character          1st Qu.: 2.000
##  Mode  :character   Median :21.00   Mode  :character          Median : 3.000
##                     Mean   :25.21                             Mean   : 3.573
##                     3rd Qu.:28.00                             3rd Qu.: 5.000
##                     Max.   :89.00                             Max.   :24.000
##                     NA's   :1
##  While working      Instrumentalist      Composer           Fav genre
##  Length:736         Length:736         Length:736         Length:736
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Exploratory        Foreign languages       BPM          Frequency [Classical]
##  Length:736         Length:736         Min.   :0.00e+00   Length:736
##  Class :character   Class :character   1st Qu.:1.00e+02   Class :character
##  Mode  :character   Mode  :character   Median :1.20e+02   Mode  :character
##                                        Mean   :1.59e+06
##                                        3rd Qu.:1.44e+02
##                                        Max.   :1.00e+09
##                                        NA's   :107
##  Frequency [Country] Frequency [EDM]    Frequency [Folk]   Frequency [Gospel]
##  Length:736          Length:736         Length:736         Length:736
##  Class :character    Class :character   Class :character   Class :character
##  Mode  :character    Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Frequency [Hip hop] Frequency [Jazz]   Frequency [K pop]  Frequency [Latin]
##  Length:736          Length:736         Length:736         Length:736
##  Class :character    Class :character   Class :character   Class :character
##  Mode  :character    Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Frequency [Lofi]   Frequency [Metal]  Frequency [Pop]    Frequency [R&B]
##  Length:736         Length:736         Length:736         Length:736
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Frequency [Rap]    Frequency [Rock]   Frequency [Video game music]
##  Length:736         Length:736         Length:736
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Anxiety        Depression       Insomnia           OCD
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 0.000
##  Median : 6.000   Median : 5.000   Median : 3.000   Median : 2.000
##  Mean   : 5.838   Mean   : 4.796   Mean   : 3.738   Mean   : 2.637
##  3rd Qu.: 8.000   3rd Qu.: 7.000   3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000   Max.   :10.000   Max.   :10.000
```

```
##
##   Music effects      Permissions
##   Length:736         Length:736
##   Class :character   Class :character
##   Mode  :character   Mode  :character
##
##
##
##
```

```
colnames(data) <- gsub(" ", "", colnames(data))
```

Keep atributes Age, Hours per day, Instrumentalist, Composer, Favgenre, Frequency..Classical, Frequency..pop, Frequency..Rock, Anxiety, Depression, Music effects

```
df <- subset(data, select = c("Age", "Hoursperday", "Instrumentalist", "Composer", "Favgenre", "Frequen
head(df)
```

```
## # A tibble: 6 x 11
##     Age Hoursp~1 Instr~2 Compo~3 Favge~4 Frequ~5 Frequ~6 Frequ~7 Anxiety Depre~8
##   <dbl>    <dbl> <chr>   <chr>   <chr>   <chr>   <chr>   <chr>     <dbl>   <dbl>
## 1    18      3   Yes     Yes     Latin   Rarely  Very f~ Never         3       0
## 2    63      1.5 No      No      Rock    Someti~ Someti~ Very f~       7       2
## 3    18      4   No      No      Video ~ Never   Rarely  Rarely        7       7
## 4    61      2.5 No      Yes     Jazz    Someti~ Someti~ Never         9       7
## 5    18      4   No      No      R&B     Never   Someti~ Never         7       2
## 6    18      5   Yes     Yes     Jazz    Rarely  Very f~ Very f~       8       8
## # ... with 1 more variable: Musiceffects <chr>, and abbreviated variable names
## #   1: Hoursperday, 2: Instrumentalist, 3: Composer, 4: Favgenre,
## #   5: ‘Frequency[Classical]‘, 6: ‘Frequency[Pop]‘, 7: ‘Frequency[Rock]‘,
## #   8: Depression
```

Remove Null values

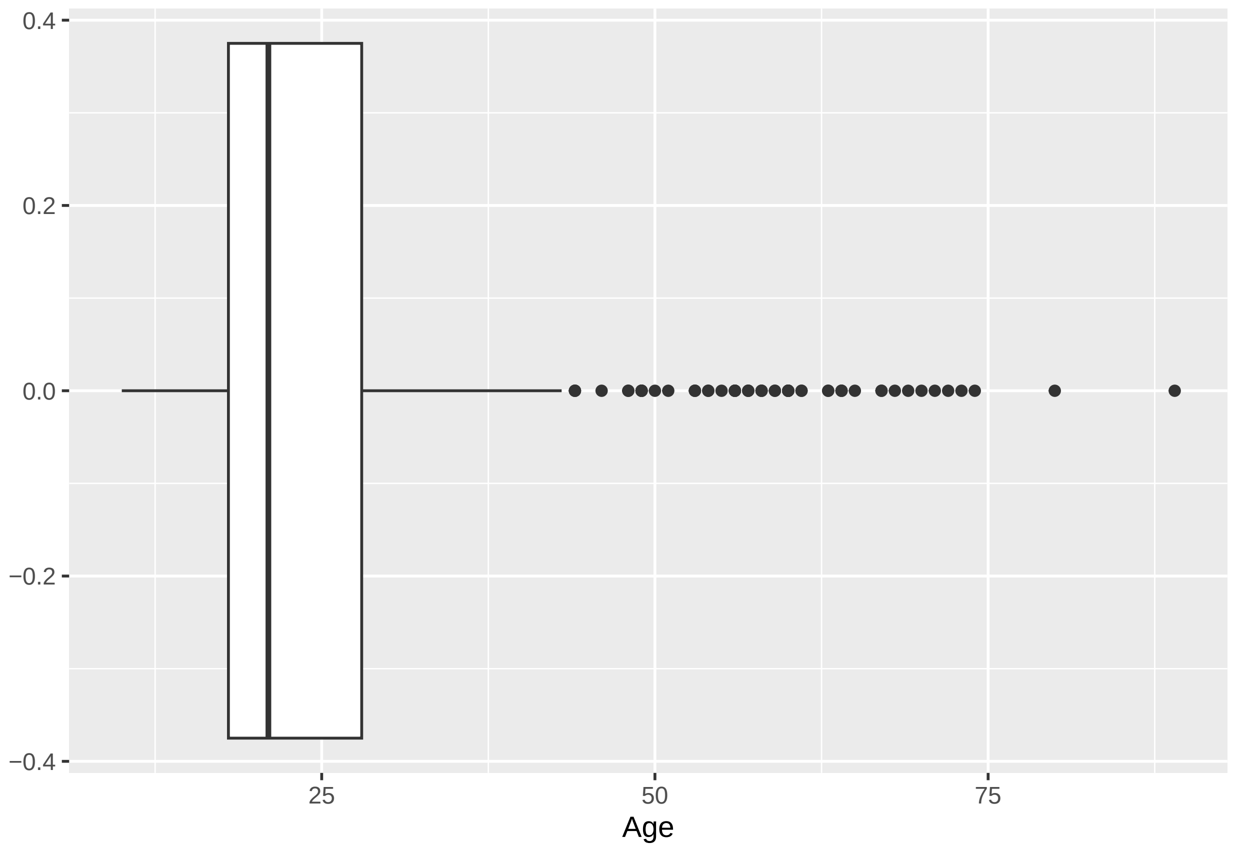
```
df <- subset(df, Musiceffects != '')
str(df$Musiceffects)
```

```
##  chr [1:728] "No effect" "Improve" "Improve" "Improve" "Improve" "Improve" ...
```

Check for outliers

```
ggplot(df,aes(x = Age))+geom_boxplot()
```

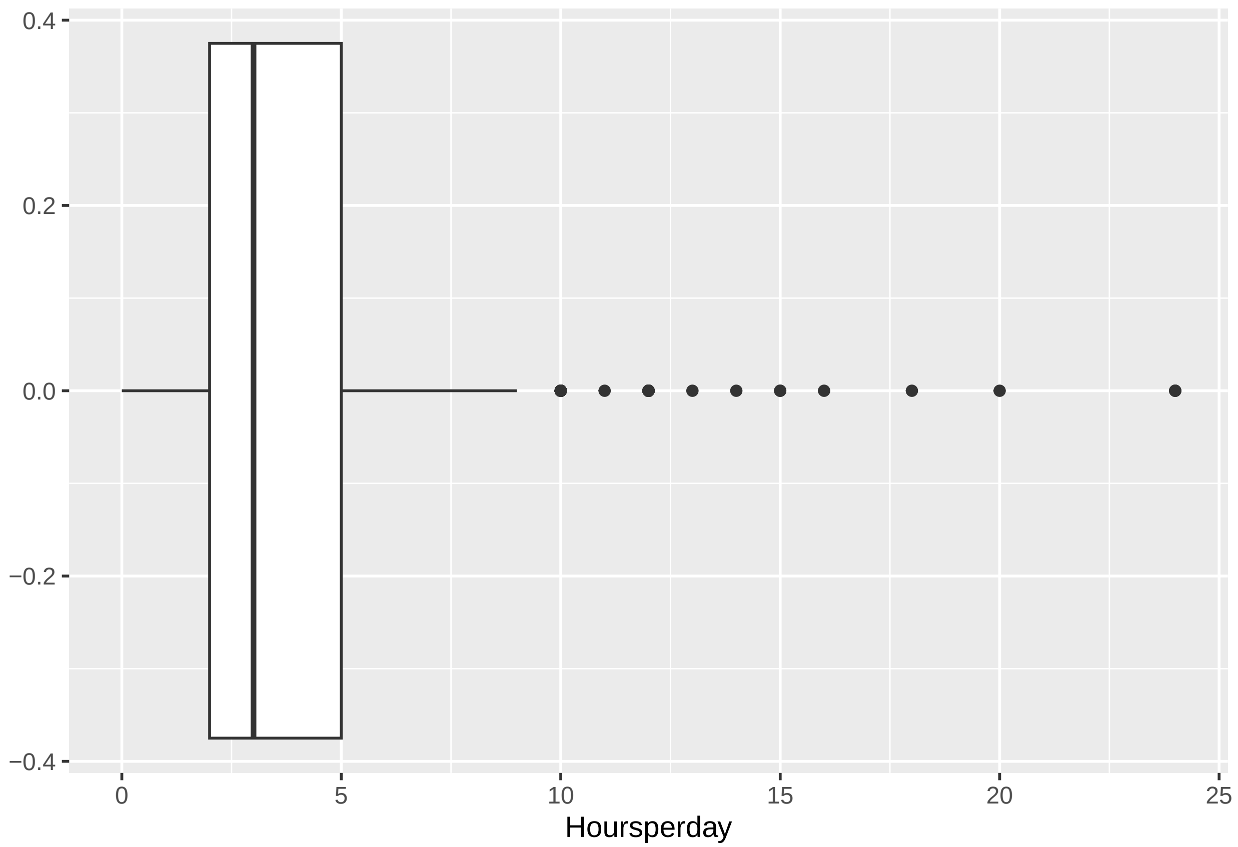
```
## Warning: Removed 1 rows containing non-finite values (‘stat_boxplot()‘).
```



```
summary(df$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   10.00   18.00   21.00   25.14   28.00   89.00       1
```

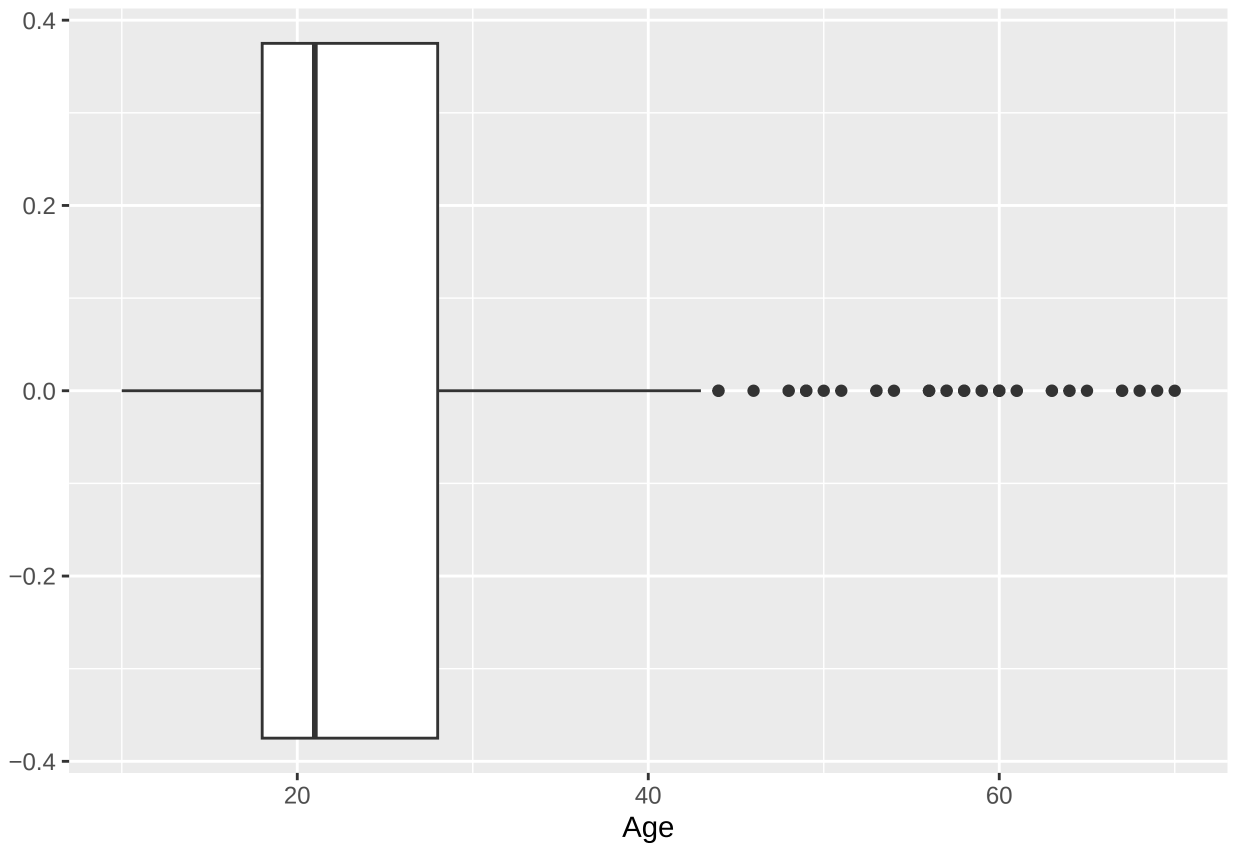
```
ggplot(df,aes(x = Hoursperday))+geom_boxplot()
```



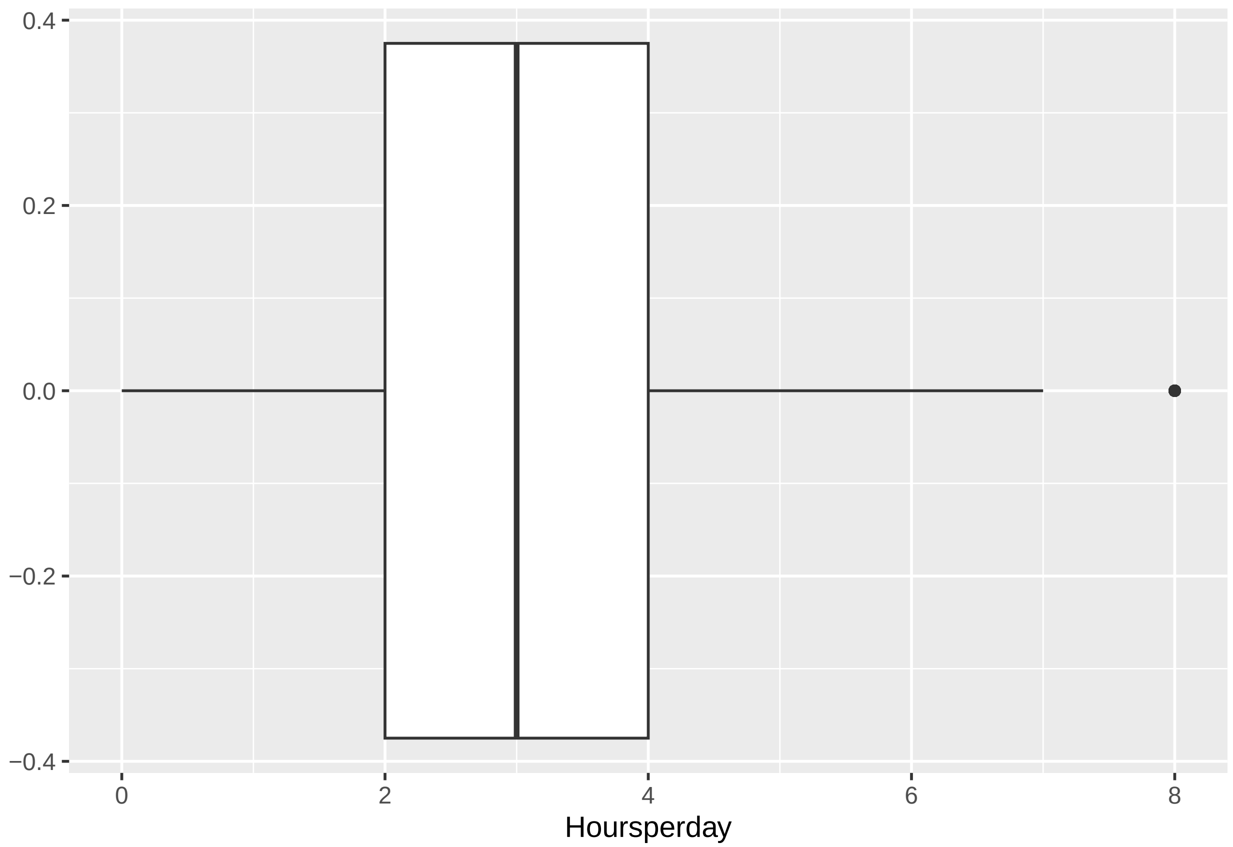
```
summary(df$Hoursperday)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   0.000   2.000   3.000   3.591   5.000  24.000
```

```
df1<- subset(df, Age<=70 & Hoursperday<=8)
ggplot(df1,aes(x = Age))+geom_boxplot()
```



```
ggplot(df1,aes(x = Hoursperday))+geom_boxplot()
```



```
summary(df1$Hoursperday)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   3.000   3.048   4.000   8.000
```

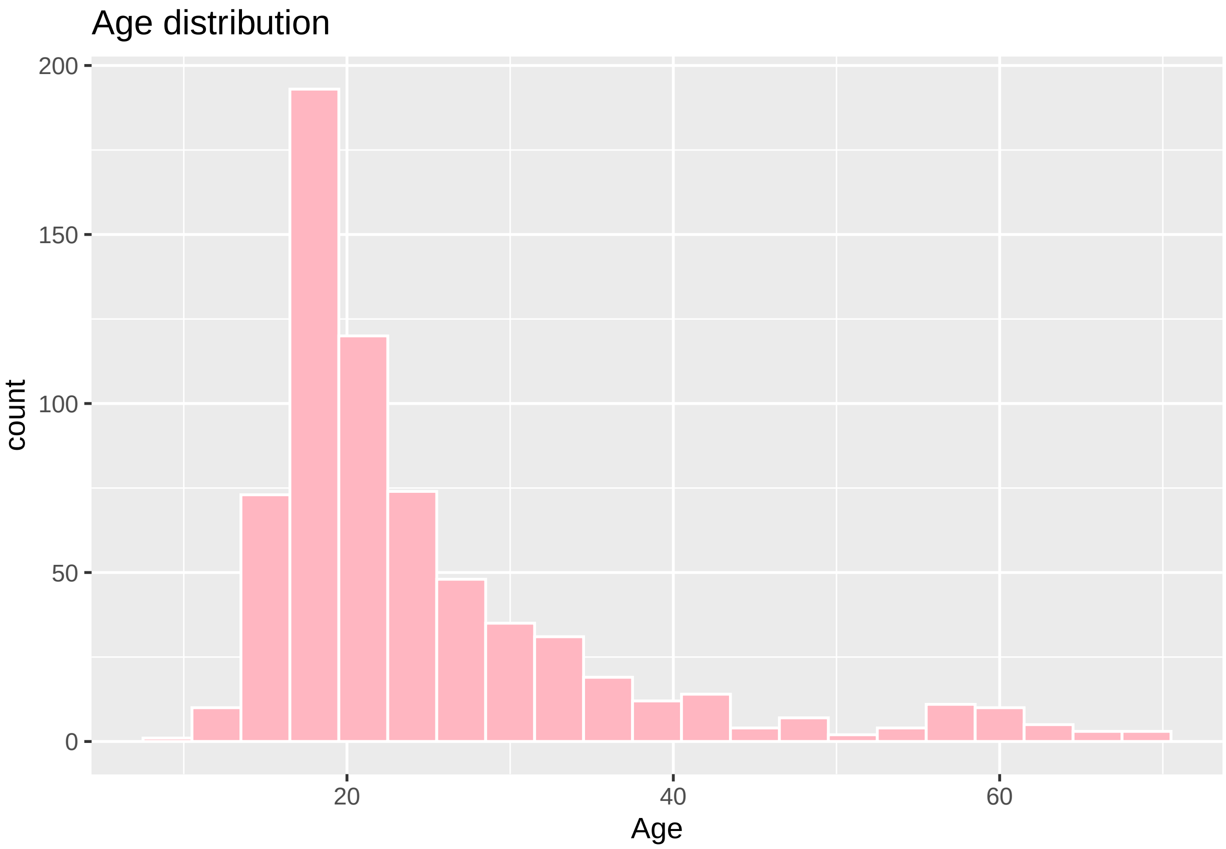
```
summary(df1$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   18.00   21.00   24.87   28.00   70.00
```

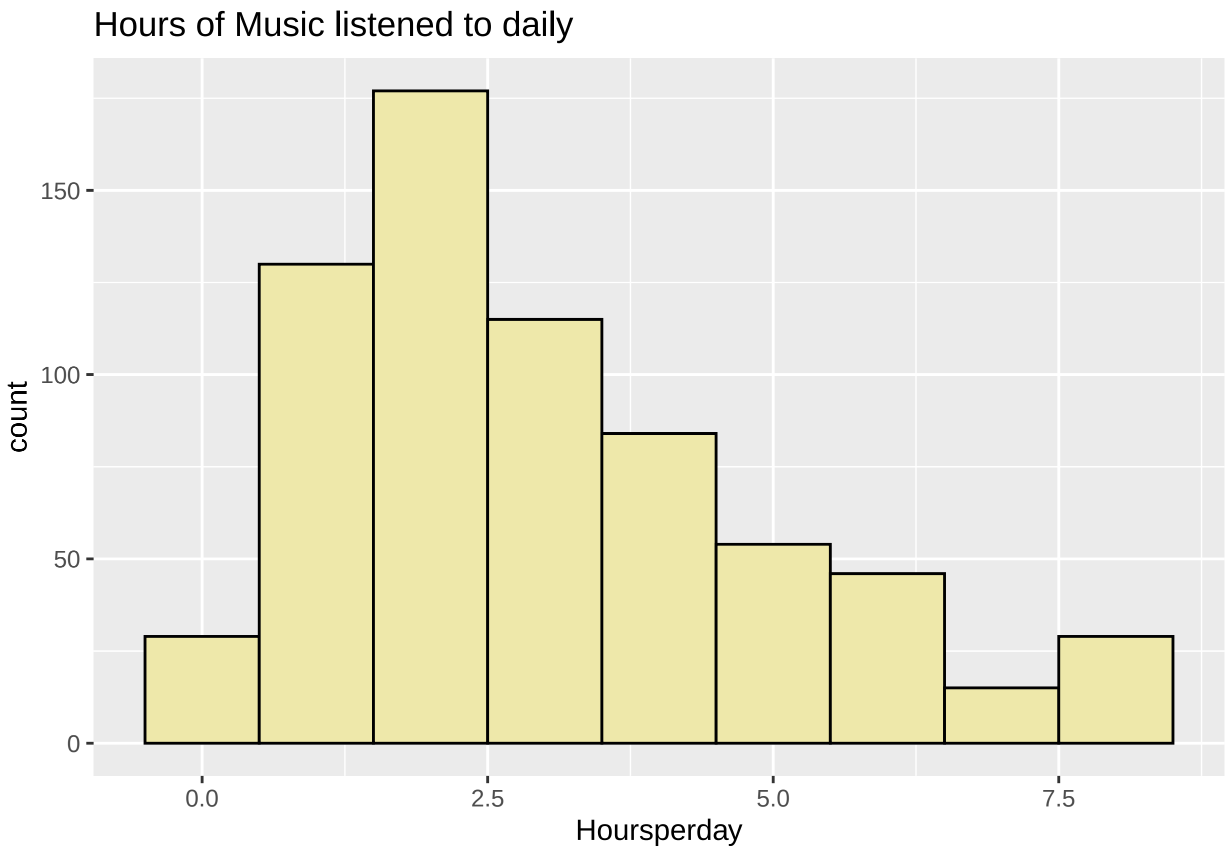
```
head(df1)
```

```
## # A tibble: 6 x 11
##     Age Hoursp~1 Instr~2 Compo~3 Favge~4 Frequ~5 Frequ~6 Frequ~7 Anxiety Depre~8
##   <dbl>    <dbl> <chr>   <chr>   <chr>   <chr>   <chr>   <chr>     <dbl>   <dbl>
## 1    18      4   No      No      Video ~ Never   Rarely  Rarely        7       7
## 2    61      2.5 No      Yes     Jazz    Someti~ Someti~ Never         9       7
## 3    18      4   No      No      R&B     Never   Someti~ Never         7       2
## 4    18      5   Yes     Yes     Jazz    Rarely  Very f~ Very f~       8       8
## 5    18      3   Yes     No      Video ~ Someti~ Rarely  Never         4       8
## 6    21      1   No      No      K pop   Never   Someti~ Never         5       3
## # ... with 1 more variable: Musiceffects <chr>, and abbreviated variable names
## #   1: Hoursperday, 2: Instrumentalist, 3: Composer, 4: Favgenre,
## #   5: ‘Frequency[Classical]‘, 6: ‘Frequency[Pop]‘, 7: ‘Frequency[Rock]‘,
## #   8: Depression
```

```
ggplot(df1, aes(x = Age)) + geom_histogram(binwidth = 3, fill = 'lightpink', color = 'white') + labs(ti
```



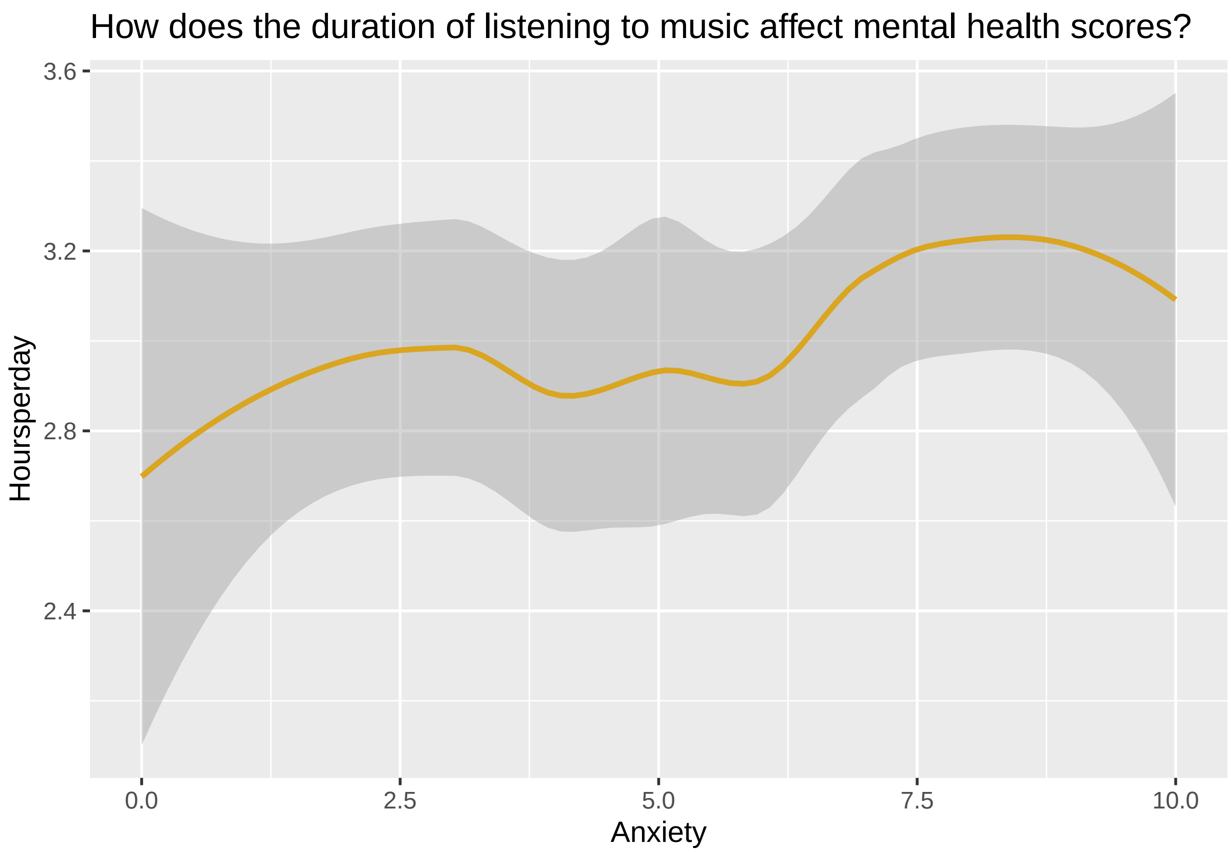
```
ggplot(df1, aes(x = Hoursperday)) + geom_histogram(binwidth = 1, fill = "palegoldenrod", colour = "blac
```



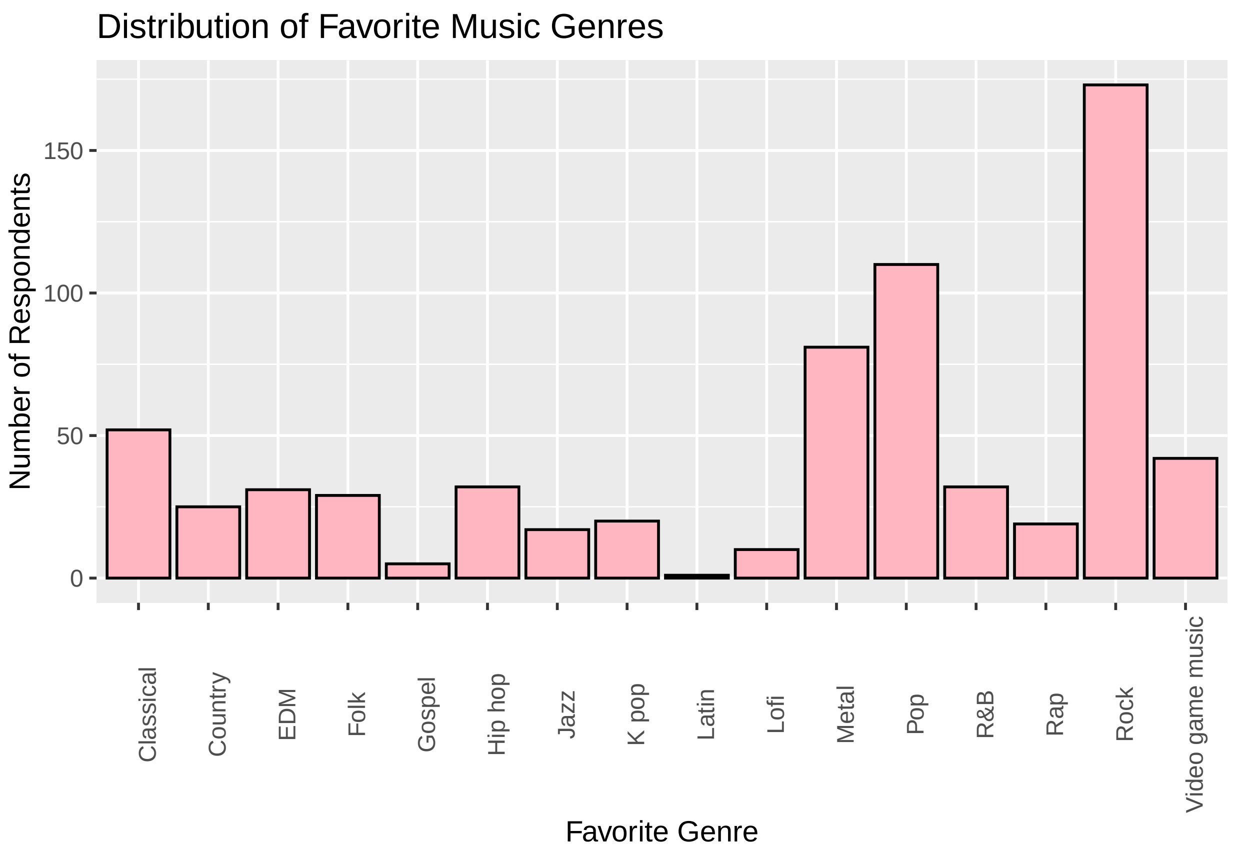
# Is there a relationship between the number of hours listened to music with age?

```
ggplot(data = df1, aes( x = Anxiety, y = Hoursperday)) + geom_smooth (method = 'loess', colour = "golden
  ggtitle("How does the duration of listening to music affect mental health scores?")
```

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



```
genre_count = dplyr::count(df1, Favgenre, sort = TRUE)
ggplot(genre_count, aes(x = Favgenre, y = n)) +
  geom_bar(stat = "identity", fill = "lightpink", color = 'black') +
  xlab("Favorite Genre") +
  ylab("Number of Respondents") +
  ggtitle("Distribution of Favorite Music Genres")+ theme(axis.text.x = element_text(angle = 90))
```



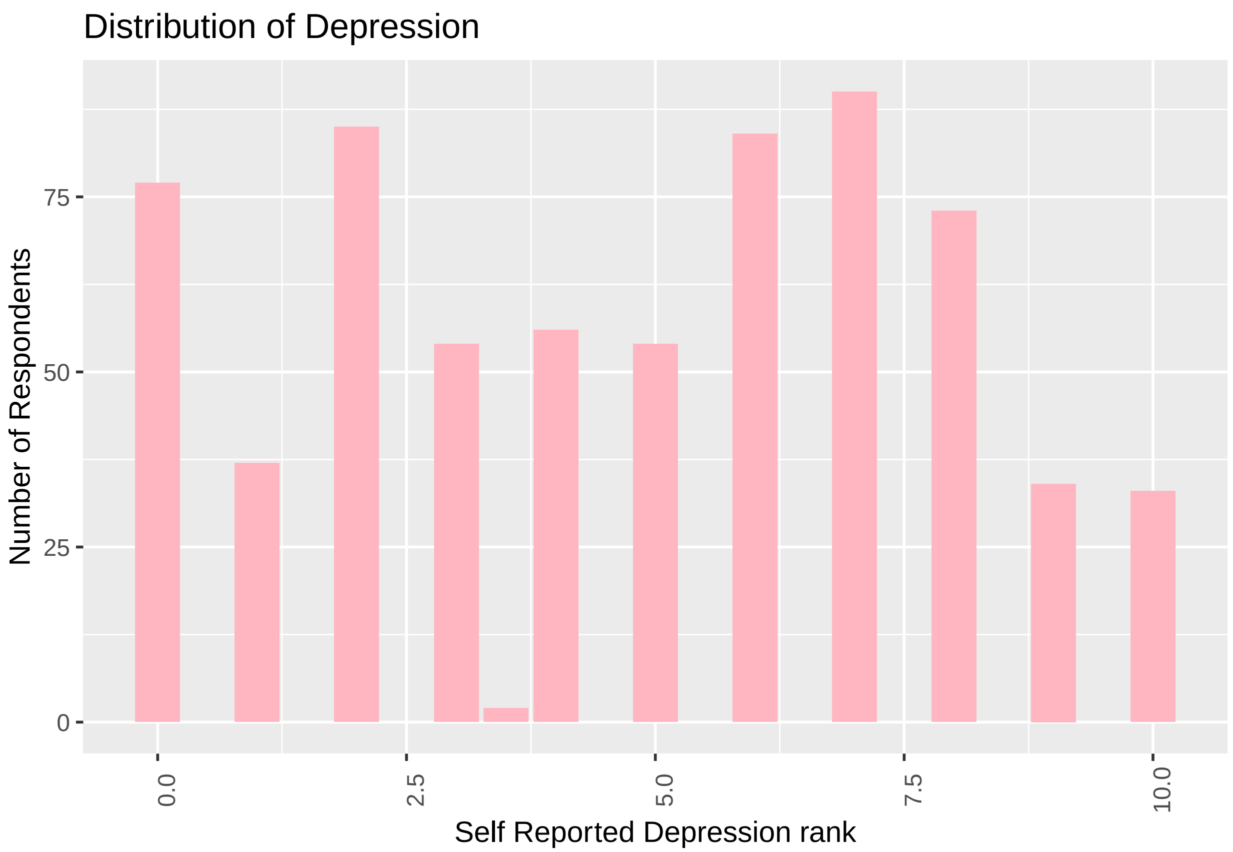
```
Composer_count = dplyr::count(df1, Composer, sort = TRUE)
Instrumentalist_count = dplyr::count(df1, Instrumentalist, sort = TRUE)
Composer_count
```

```
## # A tibble: 2 x 2
##   Composer     n
##   <chr>    <int>
## 1 No         569
## 2 Yes        110
```

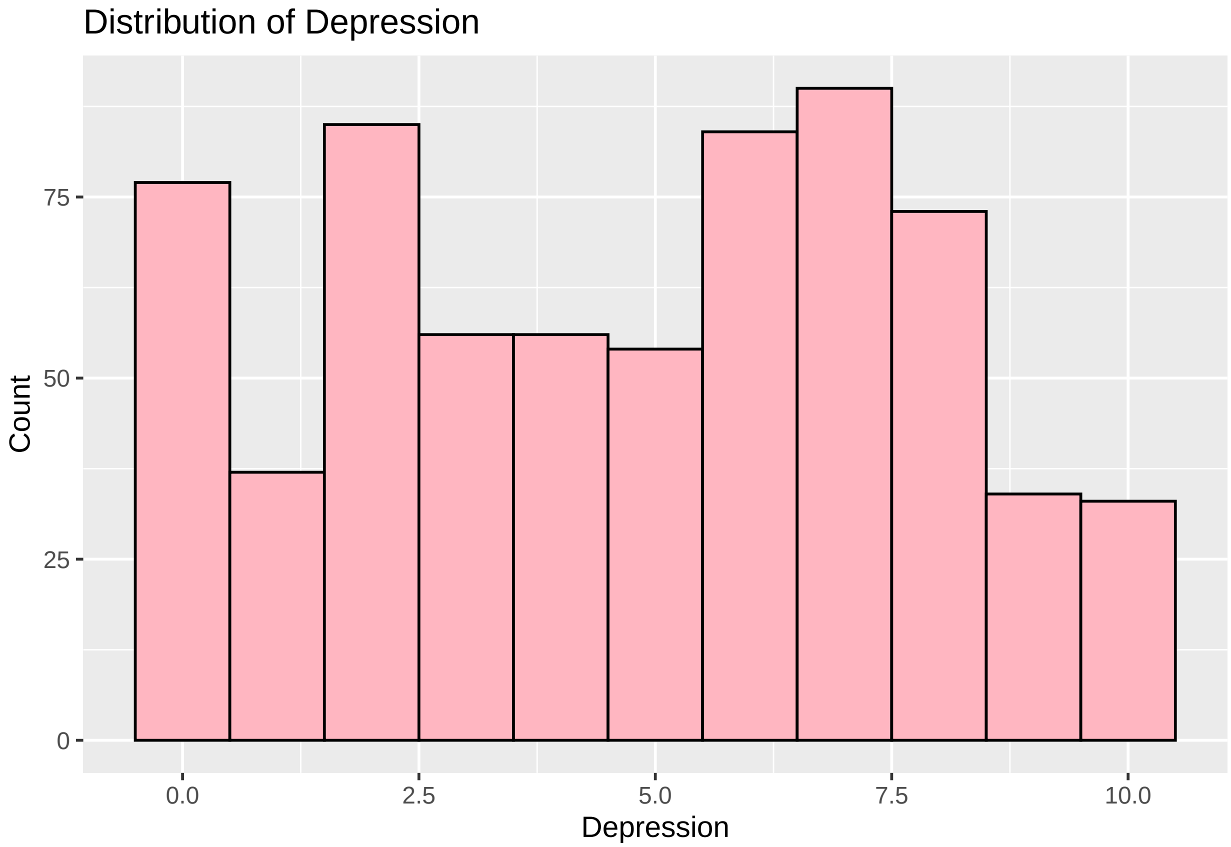
```
Instrumentalist_count
```

```
## # A tibble: 3 x 2
##   Instrumentalist     n
##   <chr>           <int>
## 1 No                459
## 2 Yes               218
## 3 <NA>                2
```

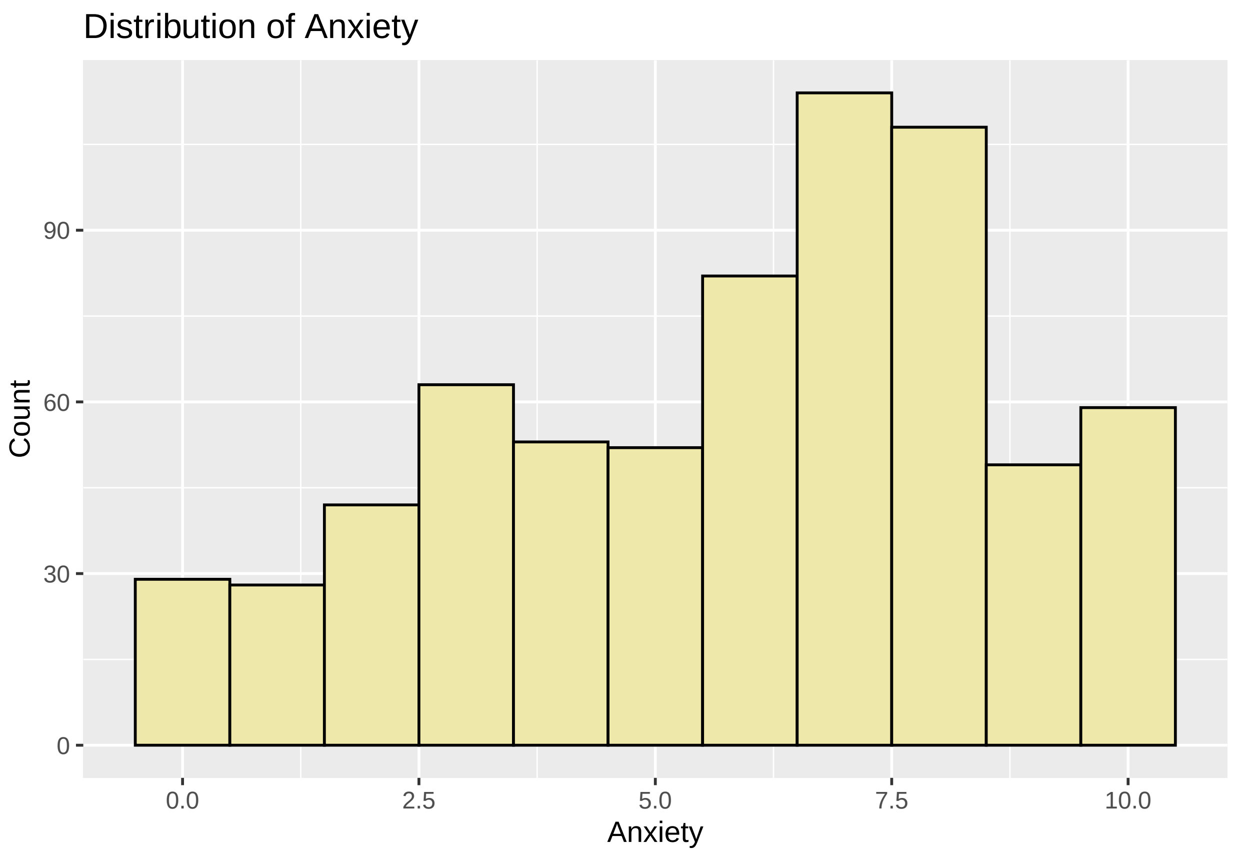
```
Depression_spread = dplyr::count(df1, Depression, sort = TRUE)
ggplot(Depression_spread, aes(x = Depression, y = n)) +
  geom_bar(stat = "identity", fill = "lightpink") +
  xlab("Self Reported Depression rank") +
  ylab("Number of Respondents") +
  ggtitle("Distribution of Depression")+ theme(axis.text.x = element_text(angle = 90))
```



```
ggplot(df1, aes(x = Depression)) +
  geom_histogram(binwidth = 1, color = "black", fill = "lightpink") +
  labs(x = "Depression", y = "Count", title = "Distribution of Depression")
```

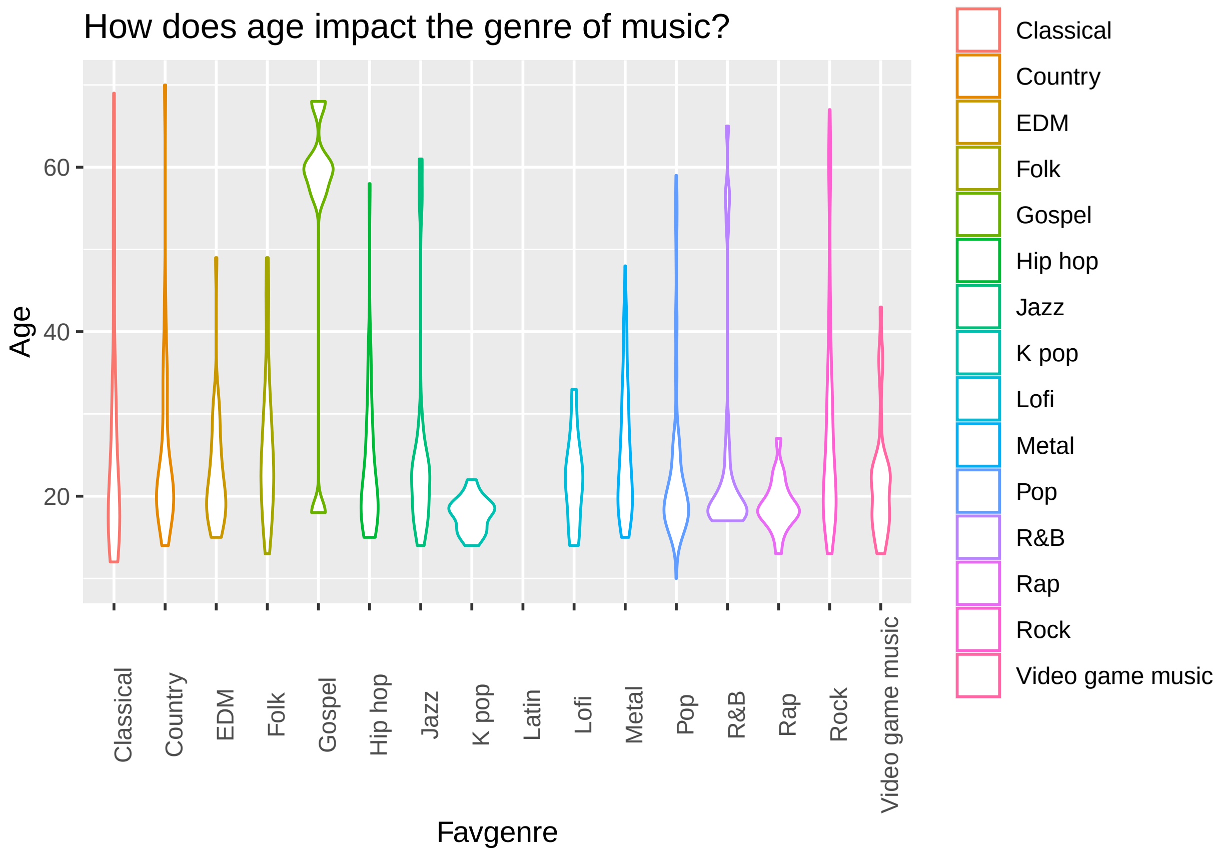


```
ggplot(df1, aes(x = Anxiety)) +
  geom_histogram(binwidth = 1, color = "black", fill = "palegoldenrod") +
  labs(x = "Anxiety", y = "Count", title = "Distribution of Anxiety")
```

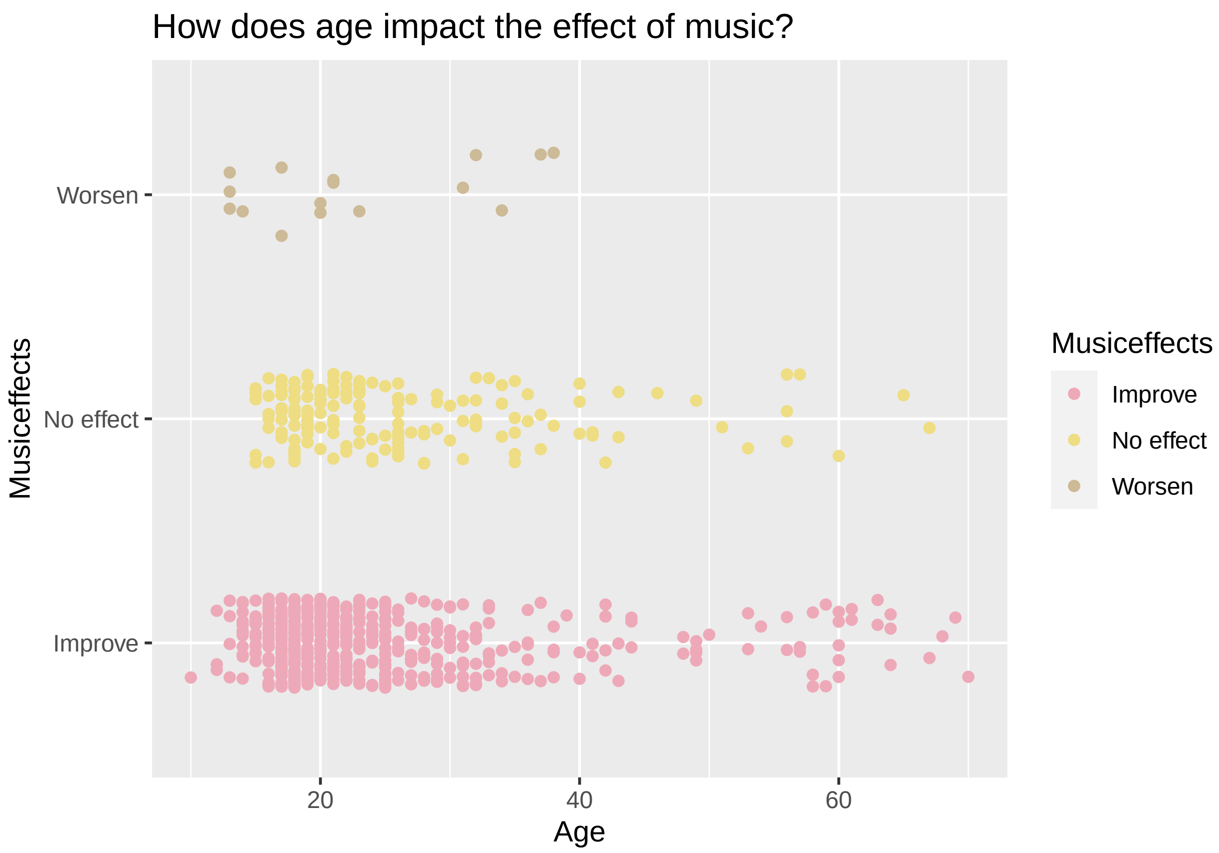


```
ggplot(df1, aes(x = Age, y = Favgenre, color = Favgenre)) +
    geom_violin() + coord_flip() +  ggtitle("How does age impact the genre of music?")+ theme(axis.text
```

```
## Warning: Groups with fewer than two data points have been dropped.
```

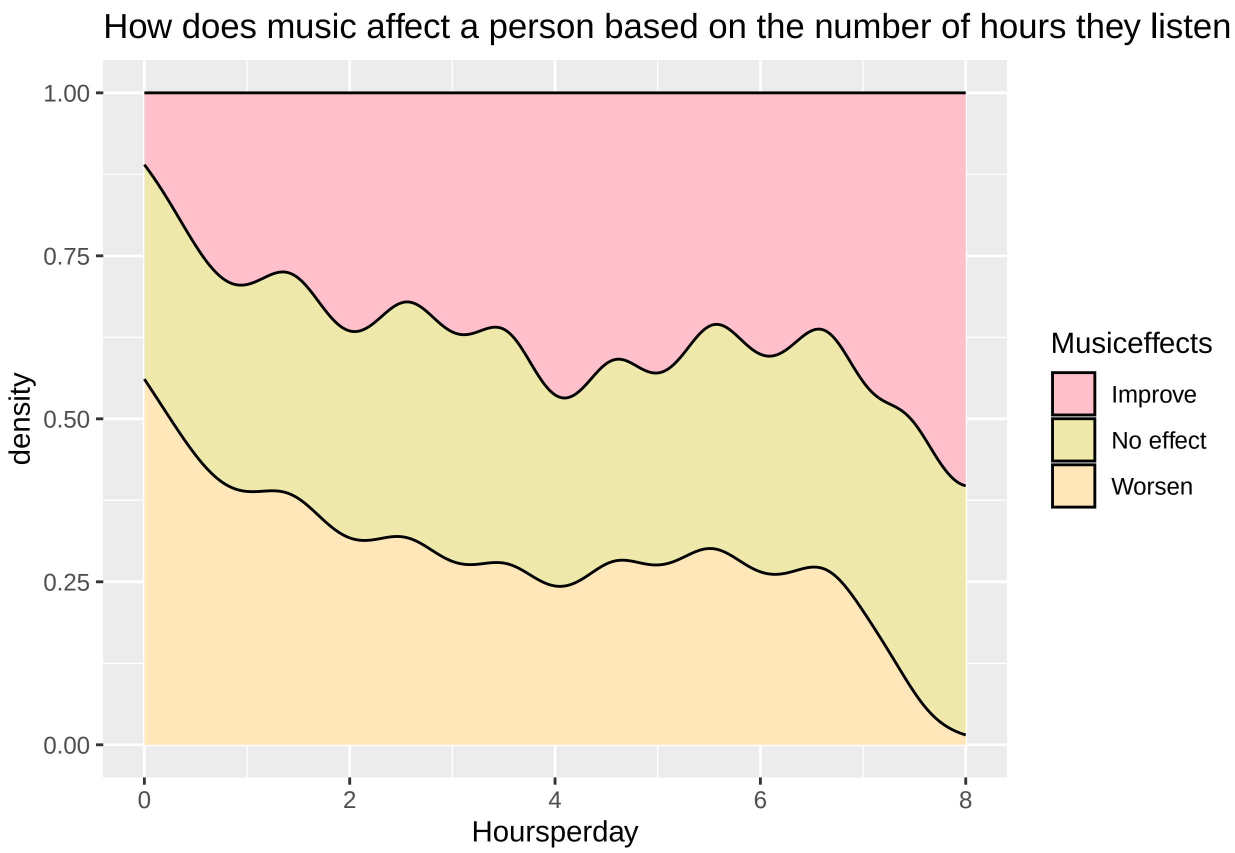


```
ggplot(df1, aes(x = Age, y = Musiceffects, color = Musiceffects)) +
    geom_jitter(width = 0, height = 0.2)+ scale_color_manual(values=c('pink2', 'lightgoldenrod', 'wheat3
```



How does age impact the effect of music?

```
ggplot(df1, aes(x = Hoursperday, fill = Musiceffects)) +
    geom_density(position = "fill") + scale_fill_manual(values=c('pink', 'palegoldenrod', 'wheat1'))+
    ggtitle("How does music affect a person based on the number of hours they listen per day?")
```

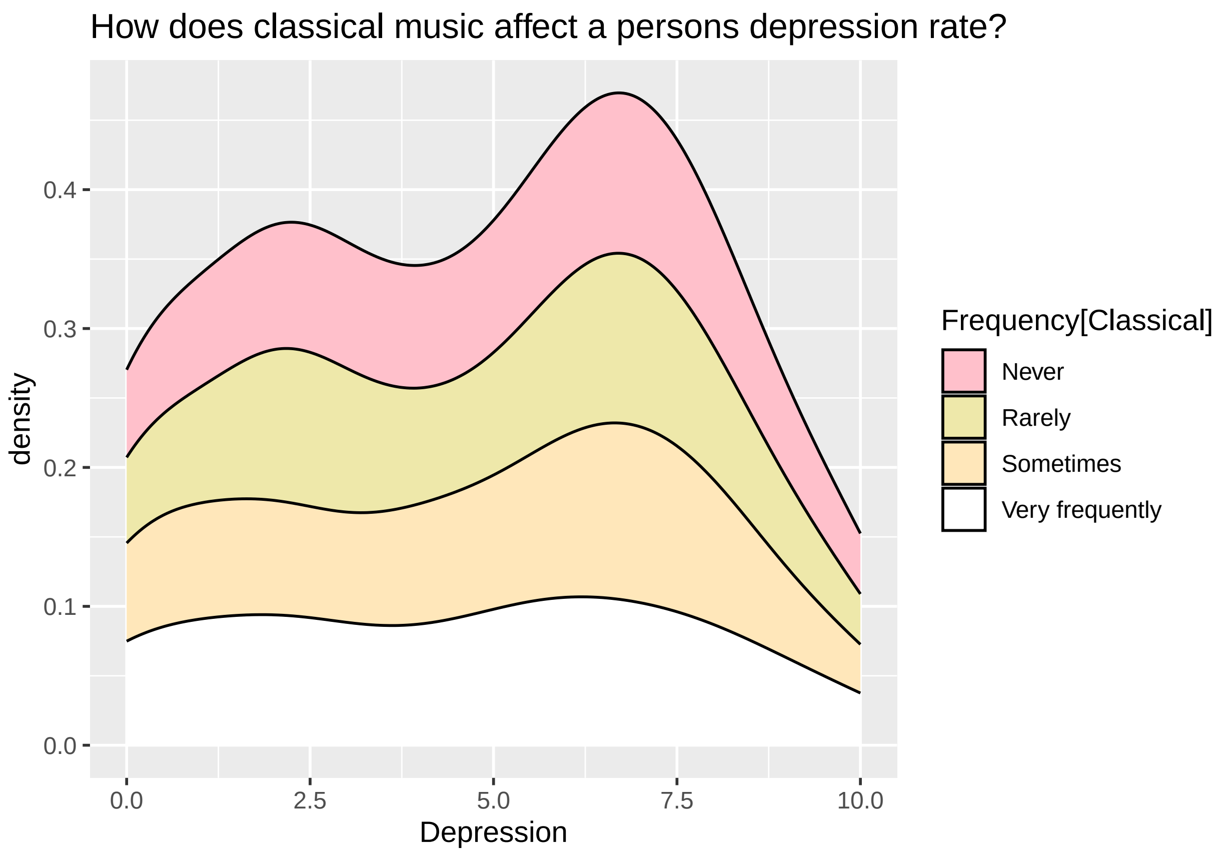


# Particular kinds of music relating to anxiety and depression

```
classical = dplyr::count(df1,`Frequency[Classical]`, sort = TRUE)
classical
```

```
## # A tibble: 4 x 2
##   ‘Frequency[Classical]‘     n
##   <chr>                  <int>
## 1 Rarely                   239
## 2 Sometimes                182
## 3 Never                    155
## 4 Very frequently          103
```

```
ggplot(df1, aes(x = Depression, fill = `Frequency[Classical]`)) +
    geom_density(position = "stack") + scale_fill_manual(values=c('pink', 'palegoldenrod', 'wheat1','wh
    ggtitle("How does classical music affect a persons depression rate?")
```



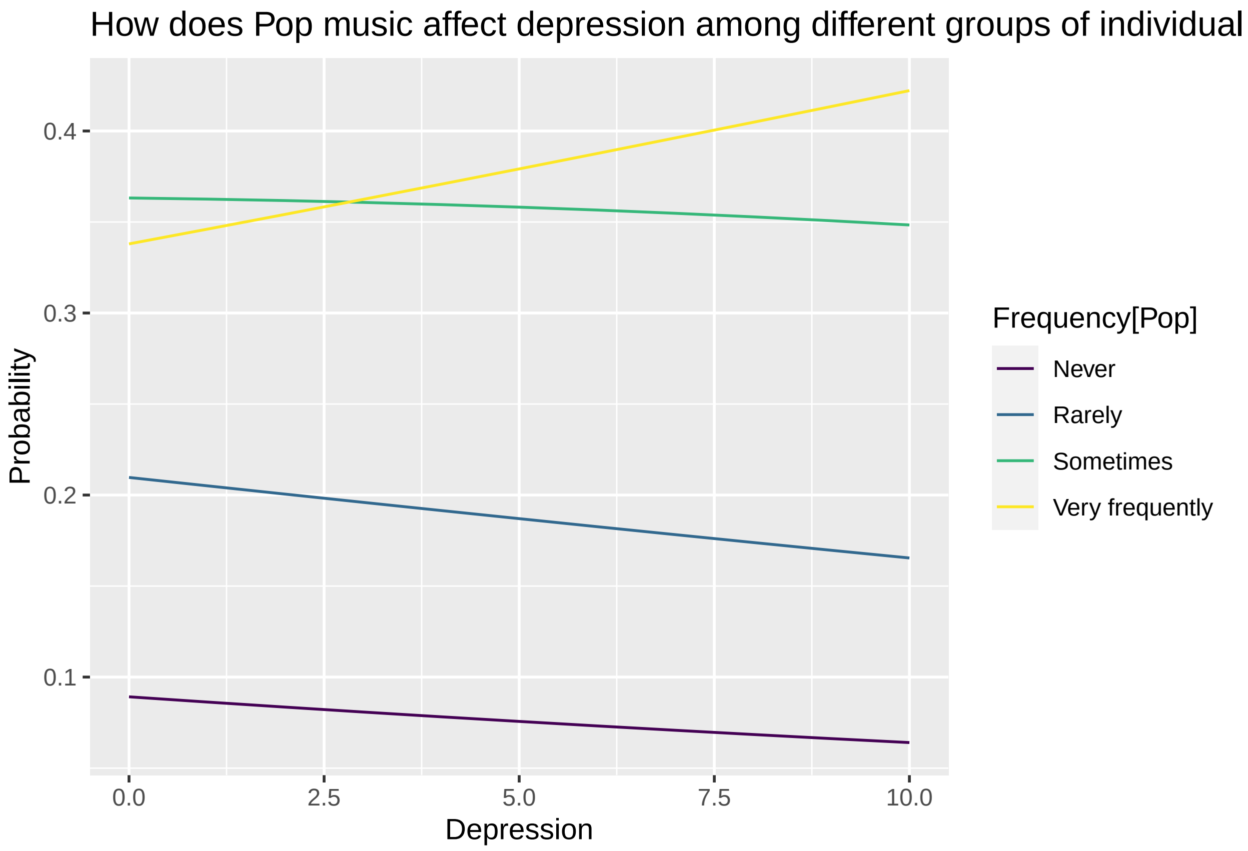
```
library(MASS)
depression.polr = polr(as.factor (`Frequency[Pop]`) ~ Depression, data = df1)
library(arm)
```

```
## 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/sushmithakeerthy/Documents/Working Folder/Indiana University/Semester 4/
```

```
display(depression.polr)
```

```
##
## Re-fitting to get Hessian
## polr(formula = as.factor(‘Frequency[Pop]‘) ~ Depression, data = df1)
##                              coef.est coef.se
## Depression                    0.04     0.02
## Never|Rarely                 -2.32     0.18
## Rarely|Sometimes             -0.85     0.14
## Sometimes|Very frequently     0.67     0.14
## ---
## n = 679, k = 4 (including 3 intercepts)
## residual deviance = 1690.9, null deviance is not computed by polr
```

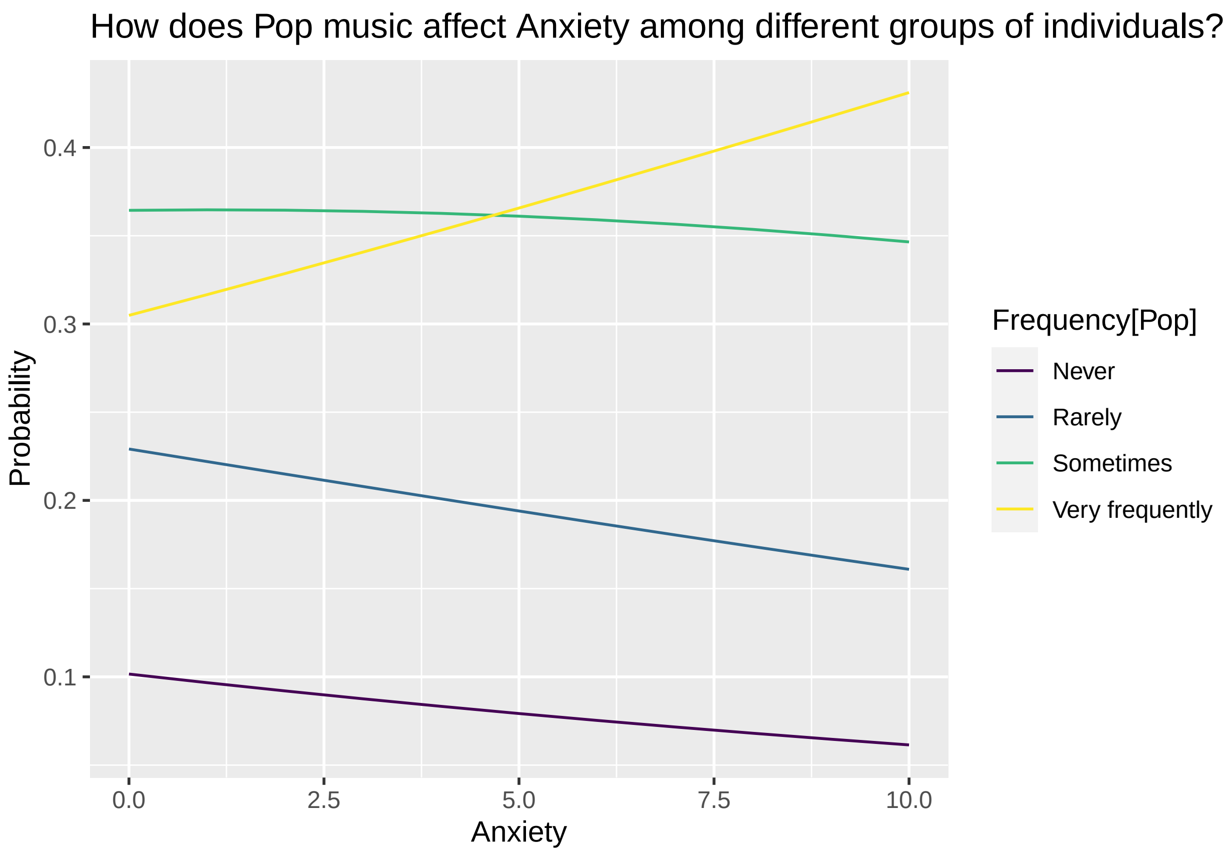
```
Depression = seq(min(df1$Depression), max(df1$Depression), 1)
pop.probs = predict(depression.polr, newdata = data.frame(Depression), type = "prob")
pop.probs.df = data.frame(Depression, pop.probs)
names(pop.probs.df) = c("Depression", "Never", "Rarely", "Sometimes", "Very frequently")
library(tidyr)
pop.probs.long = pop.probs.df %>% gather(`Frequency[Pop]`, Probability, 2:5)
pop.probs.long$`Frequency[Pop]` = factor(pop.probs.long$`Frequency[Pop]`, levels = c("Never", "Rarely",
ggplot(pop.probs.long, aes(x = Depression, y = Probability, group = `Frequency[Pop]`, color = `Frequenc
    geom_line() +
    ggtitle("How does Pop music affect depression among different groups of individuals?")
```



```
Anxiety.polr = polr(as.factor (`Frequency[Pop]`) ~ Anxiety, data = df1)
display(Anxiety.polr)
```

```
##
## Re-fitting to get Hessian
## polr(formula = as.factor(‘Frequency[Pop]‘) ~ Anxiety, data = df1)
##                          coef.est coef.se
## Anxiety                   0.05     0.03
## Never|Rarely             -2.18     0.20
## Rarely|Sometimes         -0.70     0.17
## Sometimes|Very frequently  0.82     0.17
## ---
## n = 679, k = 4 (including 3 intercepts)
## residual deviance = 1688.7, null deviance is not computed by polr
```

```
Anxiety = seq(min(df1$Anxiety), max(df1$Anxiety), 1)
pop.probs = predict(Anxiety.polr, newdata = data.frame(Anxiety), type = "prob")
pop.probs.df = data.frame(Anxiety, pop.probs)
names(pop.probs.df) = c("Anxiety", "Never", "Rarely", "Sometimes", "Very frequently")
library(tidyr)
pop.probs.long = pop.probs.df %>% gather(`Frequency[Pop]`, Probability, 2:5)
pop.probs.long$`Frequency[Pop]` = factor(pop.probs.long$`Frequency[Pop]`, levels = c("Never", "Rarely",
ggplot(pop.probs.long, aes(x = Anxiety, y = Probability, group = `Frequency[Pop]`, color = `Frequency[P
    geom_line() +
    ggtitle("How does Pop music affect Anxiety among different groups of individuals?")
```


How does Pop music affect Anxiety among different groups of individuals?
Probability
Anxiety
Frequency[Pop]
Never
Rarely
Sometimes
Very frequently