Projet statistiques

BARKAOUI Meriam et MAUBLANC Albane

2023-06-16

Contents


```
library(ggplot2)
```

```
resultat = read.table("music-mental-health.csv", sep=",",header=TRUE)
attach(resultat)
```

1 - Introduction

1 - Description du jeux de données

Dans le cadre de ce projet de statistiques, nous avons sélectionné un ensemble de données portant sur la musique et la santé mentale. Ce sujet nous intéresse particulièrement et constitue une excellente opportunité d'étude.

Nos données proviennent d'un sondage réalisé via un formulaire Google. Notre ensemble de données contient plusieurs informations, notamment:
- *Timestamp* : l'heure de soumission du formulaire.
- *Age* : l'âge du répondant (nombre entier).
- *Primary streaming service* : la plateforme de streaming musical préférée du répondant (chaîne de caractères).

- *Hours per day* : le nombre d'heures d'écoute de musique par jour (nombre décimal).
- *While working* : indique si la personne écoute de la musique en travaillant (oui/non). - *Instrumentalist* : indique si la personne joue d'un instrument de musique (oui/non). - *Composer* : indique si la personne compose de la musique (oui/non).
- *Exploratory* : indique si la personne explore de nouveaux genres musicaux ou artistes (oui/non).
- *Foreign languages* : indique si le répondant écoute de la musique avec des paroles dans une langue qu'il ne maîtrise pas (oui/non).
- *BPM* : le nombre de battements par minute du genre musical préféré (nombre entier). - *Frequency [Classical], Frequency [Country], Frequency [EDM], Frequency [Folk], Frequency [Gospel], Frequency [Hip hop], Frequency [Jazz], Frequency [K pop], Frequency [Latin], Frequency [Lofi], Frequency [Metal], Frequency [Pop], Frequency [R&B], Frequency [Rap], Frequency [Rock], Frequency [Video game music]* : la fréquence à laquelle la personne écoute de la musique dans chaque genre (très fréquemment, parfois, rarement, jamais).
- *Anxiety, Depression, Insomnia, OCD* : la personne doit s'auto-évaluer et choisir une valeur entre 0 et 10 pour déterminer le degré de présence de chaque trouble.
- *Music effect* : l'effet de la musique sur la personne (améliore, aggrave, aucun effet).
- *Permissions* : l'autorisation donnée par la personne pour publier ses données (I understand).

2 - Description des modifications de la base

Nous avons choisi de supprimer la colonne contenant l'heure de soumission du formulaire ainsi que la colonne des permissions, car ces informations ne fournissent pas de données pertinentes pour notre étude.

```
resultat <- resultat[, -which(names(resultat) == "Timestamp")]
resultat <- resultat[, -which(names(resultat) == "Permissions")]
```

Nous avons supprimé les individus qui n'ont pas renseigné leurs ages.

```
resultat <- subset(resultat, !is.na(Age) & Age != "")
```

Nous avons supprimé les indidus qui qui n'ont pas renseigné le BPM.

```
resultat <- subset(resultat, !is.na(BPM) & BPM != "")
```

```
resultat <- subset(resultat, !is.na(Primary.streaming.service) & Primary.streaming.service != "")
```

3 - Description de la population étudiée

```
nrow(resultat)
```

```
## [1] 628
```

Après nettoyage, notre dataframe est de taille 628 et notre unité statistique est l'individu.

1.4 Description des variables d'intérêt :

4 - Description des variables d'intérêt

Variables qualitatives

- *Frequency [Rock]* : Représente la fréquence à laquelle la personne écoute de la musique Rock.
  –> Il s'agit d'une variable qualitative ordonnée qui a pour valeur : (très fréquemment, parfois, rarement, jamais).

- *Primary streaming service* : Représente la plateforme de streaming musical préférée du répondant.
  –> Il s'agit d'une variable qualitative nominale de type chaine de carctères.

- *Exploratory* : indique si la personne explore de nouveaux genres musicaux ou artistes. –> Il s'agit d'une variable qualitative nominale qui a pour valeur (oui ou non).

Variables quantitatives

- *Age* : Représente l'âge du répondant. –> Il s'agit d'une variable quantitative continue de type entier
- *Depression* : la personne doit s'auto-évaluer et choisir une valeur entre 0 et 10 pour déterminer le degré de dépression qu'elle ressent –> Il s'agit d'une variable quantitative discrète, de type entier et de valeur entre 0 et 10.

- *BPM* : le nombre de battements par minute du genre musical préféré (nombre entier).
  –> Il s'agit d'une variable quantitative discrète, de type entier.

5 - Liste de questions

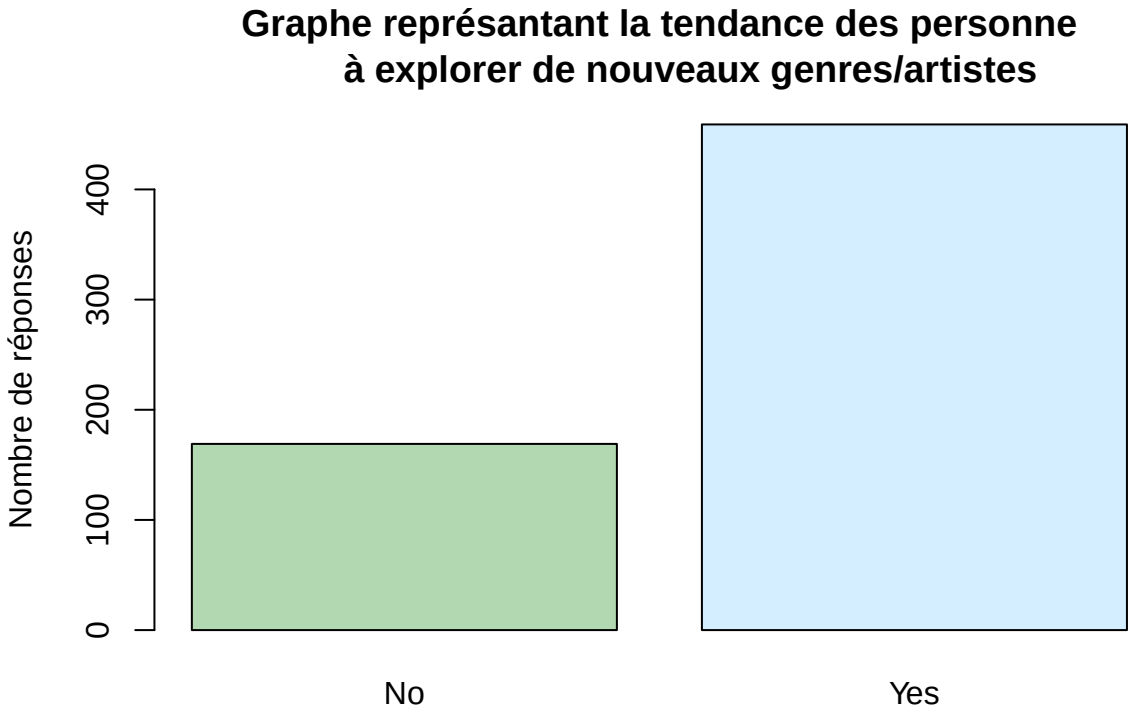
- Est-ce que le niveau de dépression des auditeurs dépend de l'écoute fréquente du genre musical Rock ?
- Existe-t-il une relation entre la plateforme de streaming musicale préférée des individus et leur ouverture à explorer de nouveaux genres musicaux ou artistes ?
- Existe-t-il une corrélation entre l'âge des répondants et le nombre de battements par minute (BPM) du genre musical préféré ?

2 - Analyse univariée

1- Variables qualitatives

1- Graphique répresentatif et résumés numériques
Dans cette partie de notre analyse, nous souhaitons évaluer la propension des individus à découvrir de nouveaux genres de musique ainsi que de nouveaux artistes.
- *Exploratory*

```
# Compter le nombre de réponses "oui" et "non"
exploratory_counts <- table(resultat$Exploratory)
exploratory_counts_df<-as.data.frame(exploratory_counts)
# Afficher le graphique
barplot(exploratory_counts, col=c("#B2D8B2", "#D4EEFF"),
        xlab="Si la personne explore de nouveaux genres musicaux/artistes",
        ylab = "Nombre de réponses" )
title("Graphe représantant la tendance des personne
      à explorer de nouveaux genres/artistes")
```



Comme nous pouvons le voir les personnes ont tendance à explorer de nouveaux genres/artistes. Les résumés numériques :

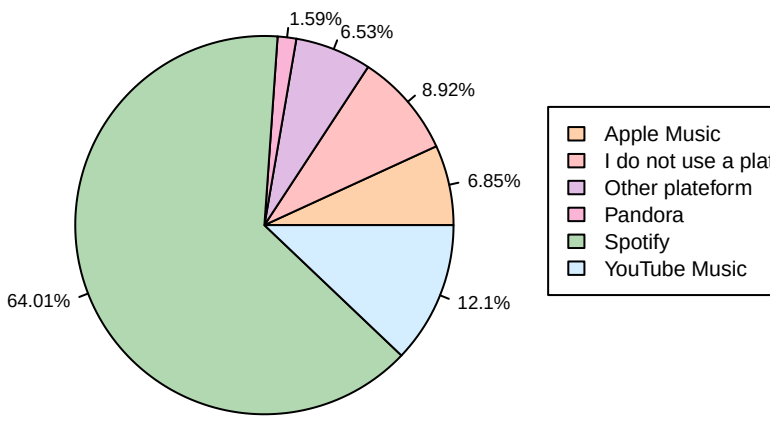
```
summary(exploratory_counts_df)
```

```
##   Var1        Freq
##  No :1   Min.   :169.0
##  Yes:1   1st Qu.:241.5
##          Median :314.0
##          Mean   :314.0
##          3rd Qu.:386.5
##          Max.   :459.0
```

- *Primary streaming service* :
  1- Graphique répresentatif et résumés numériques

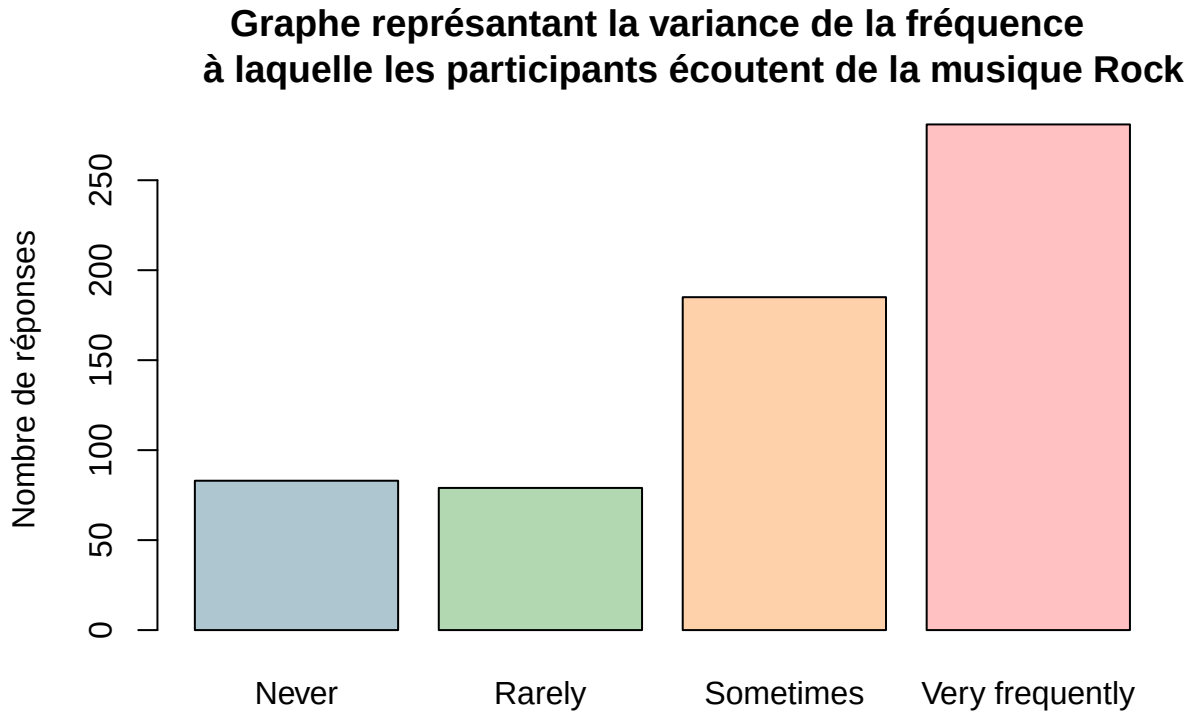
```
streaming_counts <- table(resultat$Primary.streaming.service)
streaming_counts_df <- as.data.frame(streaming_counts)
# Calculer les pourcentages
streaming_counts_df$Percentage <- paste0(round(streaming_counts_df$Freq /
                                               sum(streaming_counts_df$Freq) * 100, 2), "%")

# Créer le graphique en camembert
pie_chart <- pie(streaming_counts, labels = streaming_counts_df$Percentage,
        col=c( "#FFD1AA", "#FFC1C1", "#E0BBE4", "#FAB4D6","#B2D8B2", "#D4EEFF"),
        main="Répartition des services de streaming de musique", cex = 0.6)
```

```
        legend(1.2, .5, c("Apple Music", "I do not use a plateform." ,
              "Other plateform", "Pandora", "Spotify", "YouTube Music"),
        cex = 0.7, fill = c( "#FFD1AA", "#FFC1C1", "#E0BBE4", "#FAB4D6","#B2D8B2", "#D4EEFF"))
```



Spotify est la plateforme la plus utilisée, représentant 64% des réponses. Peut être en raison de sa vaste bibliothèque musicale, de son interface conviviale, de sa disponibilité multiplateforme et de son modèle freemium. - *Frequency [Rock]* :

```
frequency_rock_counts<- table(resultat$Frequency..Rock.)
frequency_rock_counts_df<-as.data.frame(frequency_rock_counts)
barplot(frequency_rock_counts, col=c("#AEC6CF", "#B2D8B2", "#FFD1AA", "#FFC1C1"),
        xlab="La fréquence à laquelle la personne écoute de la musique Rock",
        ylab="Nombre de réponses")
title("Graphe représantant la variance de la fréquence
      à laquelle les participants écoutent de la musique Rock")
```



La majorité des personnes qui ont répondu à ce sondage écoutent fréquemment du rock.

```
summary(frequency_rock_counts_df)
```

```
##                Var1        Freq
##  Never           :1   Min.   : 79
##  Rarely          :1   1st Qu.: 82
##  Sometimes       :1   Median :134
##  Very frequently:1   Mean   :157
##                      3rd Qu.:209
##                      Max.   :281
```

2- Variables quantitatives

1- Graphique répresentatif et résumés numériques
- *Age* : continue

```
min(resultat$Age)
```

```
## [1] 10
```

```
max(resultat$Age)
```

```
## [1] 89
```

```r
nombre_personnes_entre18et22 <- sum(resultat$Age >= 18 & resultat$Age <22)
print(nombre_personnes_entre18et22)
```

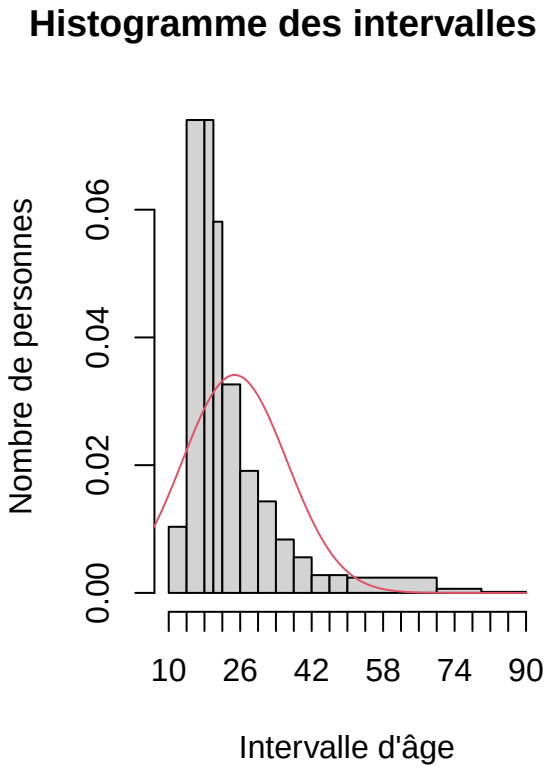
```
## [1] 209
```

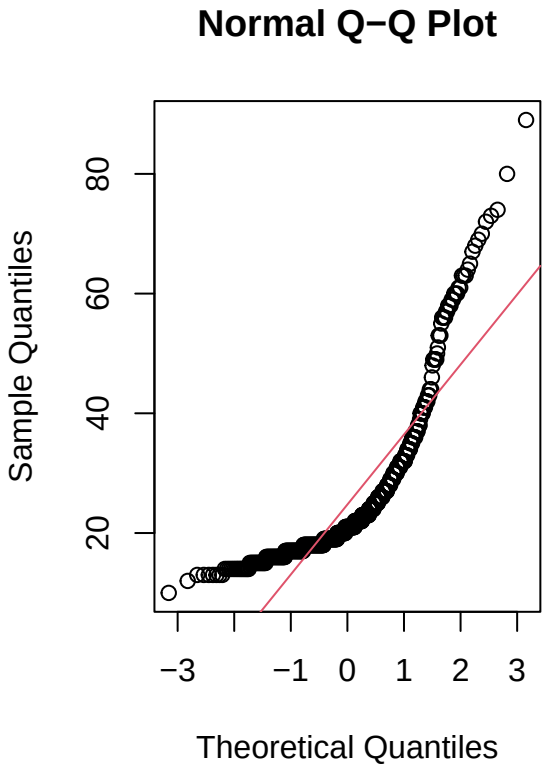
Au départ, nous avons choisi de regrouper les âges par intervalles de 4 ans, mais nous avons constaté que nous avions une population de 209 personnes âgées entre 18 et 22 ans, ce qui représente un tiers de notre population. Par conséquent, nous avons décidé de rediviser ces intervalles en deux, de 18 à 20 ans et de 20 à 22 ans.

```r
par(mfrow=c(1,2))
# Créer des intervalles d'âges personnalisés
intervalles <- cut(resultat$Age,
              breaks = c(10, 14, 18,20, 22, 26, 30, 34, 38, 42, 46, 50, 70, 80, 90),
              right = FALSE, include.lowest = TRUE)

tableau_intervalles <- table(intervalles)

h <- hist(resultat$Age, breaks = c(10, 14, 18,20, 22, 26, 30, 34, 38, 42, 46, 50, 70, 80, 90),
     main = "Histogramme des intervalles d'âges",
     prob=TRUE,
     xlab = "Intervalle d'âge", ylab = "Nombre de personnes",xaxt="n")
axis(side=1, at=seq(10,90,4),labels = seq(10,90,4))
points(seq(0,90,0.5),dnorm(seq(0,90,0.5),mean(resultat$Age),sd(resultat$Age)),col=2,type="l")
print(resultat$Age)
```

```
##   [1] 18 63 18 61 18 18 18 21 19 18 19 19 18 16 16 17 15 15 17 19 18 16 18 14 18
##  [26] 17 17 19 17 16 18 21 26 20 23 18 37 17 18 17 36 24 18 19 17 16 23 23 22 18
##  [51] 23 23 19 22 15 25 32 36 24 29 41 36 19 26 22 16 17 19 18 18 19 19 22 16 19
##  [76] 37 14 26 32 30 43 24 36 19 22 20 31 19 15 18 25 17 28 20 21 41 20 34 23 16
## [101] 19 22 19 23 23 22 17 28 19 17 21 29 22 18 13 24 35 29 28 32 32 21 18 19 25
## [126] 17 16 27 53 25 34 22 17 26 19 21 32 21 25 28 41 18 22 18 49 21 27 19 27 27
## [151] 19 42 60 19 20 28 18 21 21 18 16 19 31 16 17 20 30 19 33 28 25 22 19 44 26
## [176] 23 17 30 35 20 18 19 37 31 29 20 36 18 18 25 18 31 17 30 40 64 14 26 35 33
## [201] 30 31 27 53 23 25 16 31 31 38 17 25 20 15 16 20 23 22 15 17 22 21 19 34 18
## [226] 21 20 19 21 31 20 19 19 24 20 15 17 20 14 24 14 23 22 25 27 17 40 18 16 18
## [251] 15 20 16 20 22 21 21 29 56 17 22 26 30 49 22 32 32 22 17 21 17 25 15 13 23
## [276] 17 59 22 17 27 23 15 22 23 19 32 40 21 21 23 35 18 21 23 13 19 33 18 17 16
## [301] 18 19 15 17 25 24 23 16 16 20 17 20 21 16 43 10 27 24 18 17 19 16 33 32 15
## [326] 27 20 23 16 17 14 32 20 18 14 13 19 14 14 18 15 17 44 21 23 18 26 13 15 57
## [351] 12 26 17 69 17 32 22 38 18 14 14 19 20 18 30 31 72 73 16 19 16 39 15 14 21
## [376] 38 23 16 21 13 16 21 17 24 14 22 20 23 61 24 34 49 33 22 38 23 34 40 16 23
## [401] 22 41 63 28 58 58 67 70 60 15 18 58 28 22 49 20 55 48 14 18 25 18 19 74 36
## [426] 14 20 15 19 29 24 18 21 19 25 27 60 43 51 63 24 19 56 20 19 20 65 28 18 42
## [451] 80 56 20 15 57 22 17 19 50 24 18 15 17 18 18 18 18 17 34 18 18 19 25 25 18
## [476] 24 16 18 18 16 68 16 21 53 29 18 18 22 19 26 14 27 26 59 30 27 18 23 18 40
## [501] 24 23 29 16 18 21 32 22 17 18 18 24 21 46 18 16 56 21 20 21 25 33 20 17 24
## [526] 26 17 16 27 28 19 21 21 15 20 22 22 20 49 42 26 17 30 23 30 31 42 16 34 21
## [551] 27 18 26 37 26 19 22 24 21 28 23 20 17 19 32 18 20 27 15 24 20 16 21 27 17
## [576] 21 17 21 36 18 18 18 18 18 18 20 19 17 23 18 19 89 20 16 17 16 30 37 44 21
## [601] 19 26 18 35 16 19 17 29 21 17 17 22 17 19 19 16 19 13 18 26 14 21 21 17 18
## [626] 19 19 29
```

```
qqnorm(resultat$Age)
abline(mean(resultat$Age),sd(resultat$Age),col=2)
```






```
summary(resultat$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   10.00   18.00   21.00   24.76   27.00   89.00
```

2- Etude du caractère gaussien

ça suit la loi normale . **3- Les estimateurs de moyenne et variance**
Moyenne d'âges:

```
mean(resultat$Age)
```

```
## [1] 24.76115
```

Variance :

```
var(resultat$Age)
```

```
## [1] 136.6829
```

4- Intervalles de confiance de la moyenne
Intervalles de confiance de la moyenne:

```
interval=t.test(resultat$Age, conf.level = 0.5)
interval$conf.int
```

```
## [1] 24.4463 25.0760
## attr(,"conf.level")
## [1] 0.5
```

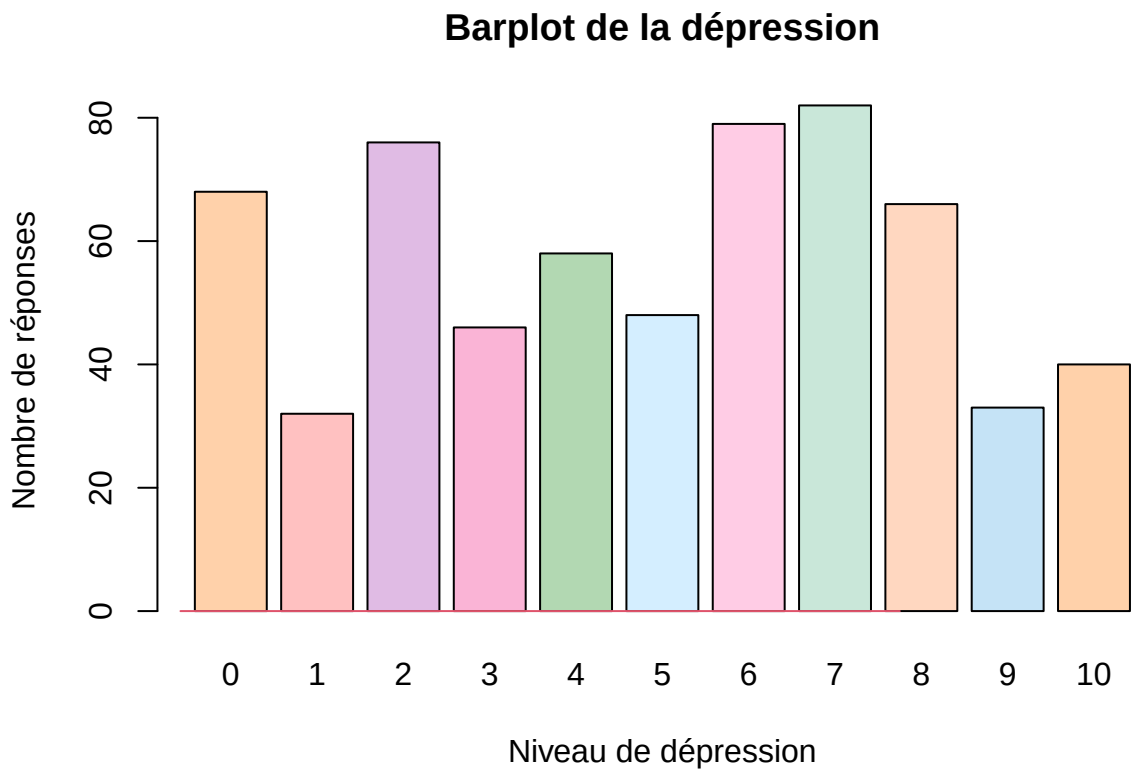
Le résultat obtenu [24.4463 25.0760] correspond à l'intervalle de confiance à 50% pour la moyenne de la variable d'âge. Cela signifie que l'on estime avec 50% de confiance que la vraie moyenne de l'âge de la population se situe entre 24.4463 et 25.0760.

```
interval=t.test(resultat$Age, conf.level = 0.95)
interval$conf.int
```

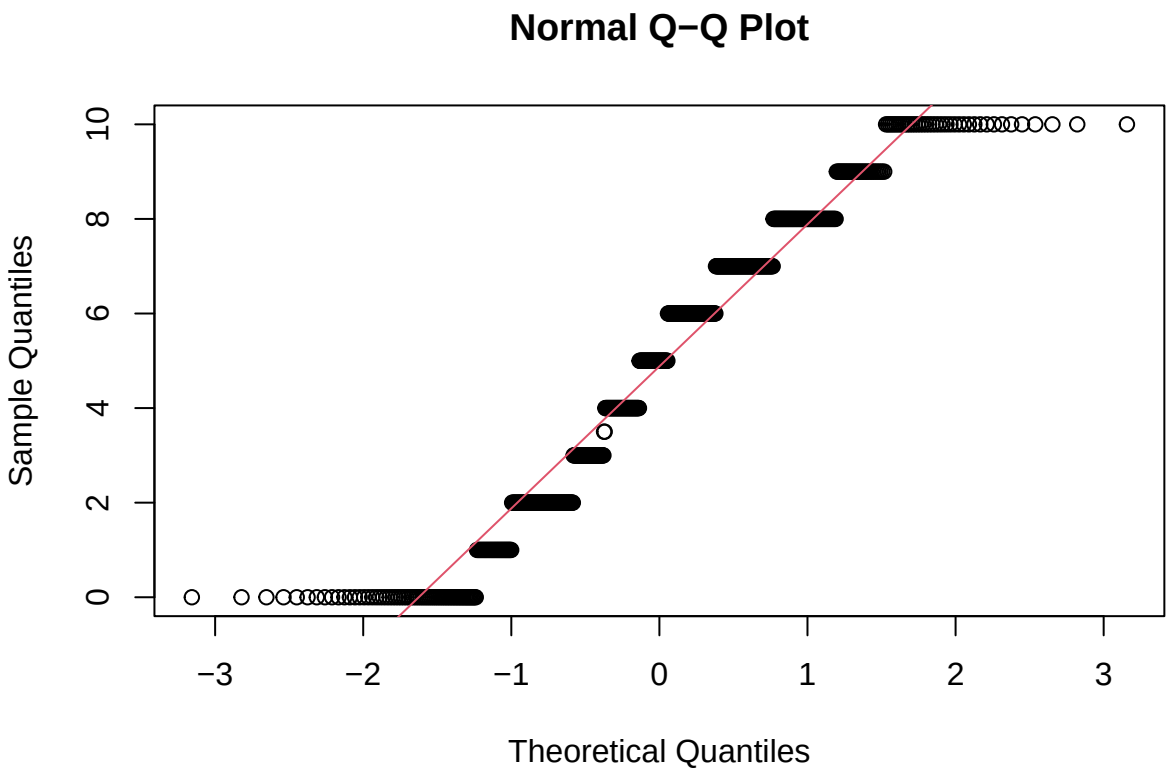
```
## [1] 23.84500 25.67729
## attr(,"conf.level")
## [1] 0.95
```

Les valeurs [23.84500, 25.67729] correspondent à l'intervalle de confiance à 95% pour la moyenne de la variable analysée. Cela signifie qu'avec un niveau de confiance de 95%, nous estimons que la vraie moyenne de la population se situe entre 23.84500 et 25.67729.
- *Depression* : discrete **1- Graphique répresentatif et résumés numériques**

```
depression_counts<- table(round(resultat$Depression))
depression_counts_df <- as.data.frame(depression_counts)
# Renommer les colonnes du data frame
colnames(depression_counts_df) <- c("Niveau de dépression", "Nombre de personnes")

barplot(depression_counts_df$`Nombre de personnes`,
        col=c("#FFD1AA", "#FFC1C1", "#E0BBE4", "#FAB4D6", "#B2D8B2", "#D4EEFF",
              "#FFCCE5", "#C9E7D9", "#FFD7C0", "#C5E3F6"),
        names.arg = depression_counts_df$`Niveau de dépression`,
        xlab = "Niveau de dépression", ylab = "Nombre de réponses",

        main = "Barplot de la dépression")

points(seq(0,10,.5),dnorm(seq(0,10,0.5),
mean(depression_counts_df$`Nombre de personnes`),sd(depression_counts_df$`Nombre de personnes`)),col=2,
```




```
qqnorm(resultat$Depression)
abline(mean(resultat$Depression),sd(resultat$Depression),col=2)
```




```
summary(resultat$Depression)
```

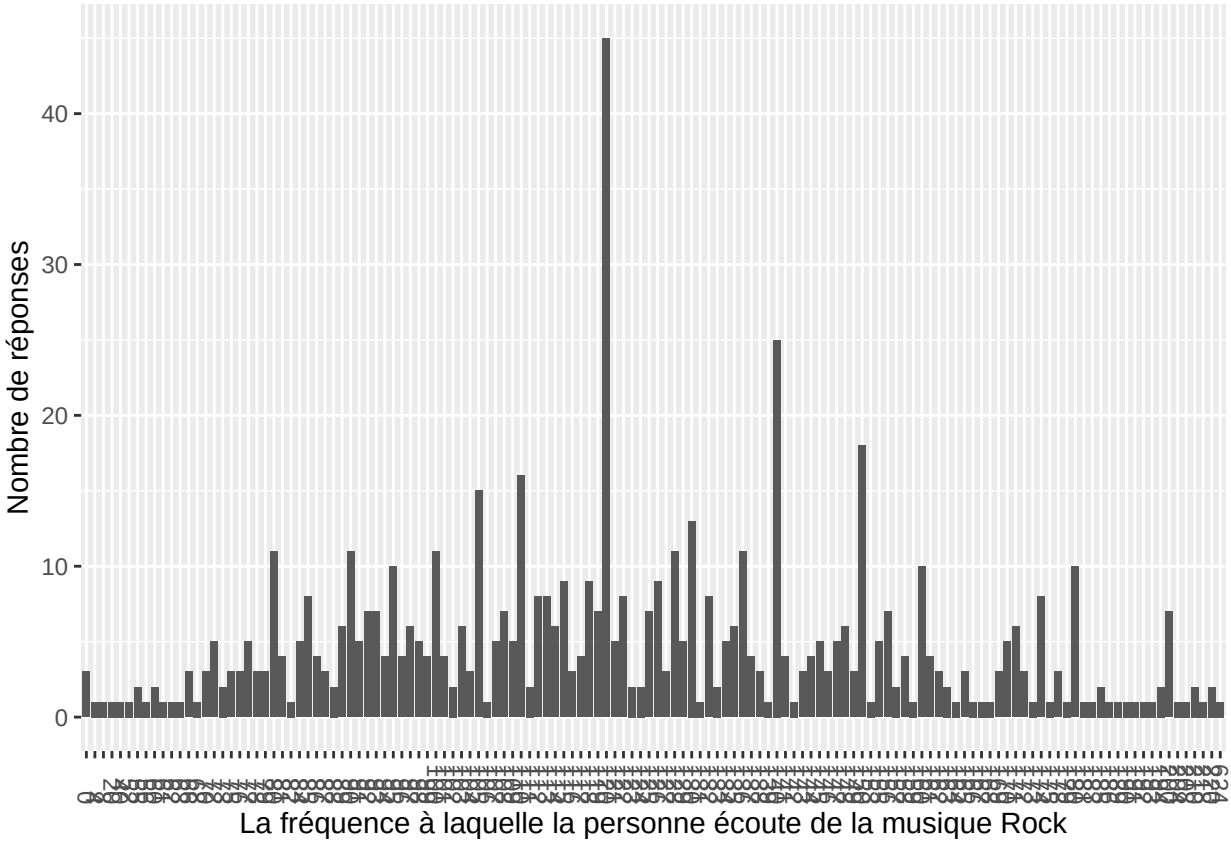
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   2.000   5.000   4.882   7.000  10.000
```

2- Etude du caractère gaussien
3- Les estimateurs de moyenne et variance 4- Intervalles de confiance de la moyenne

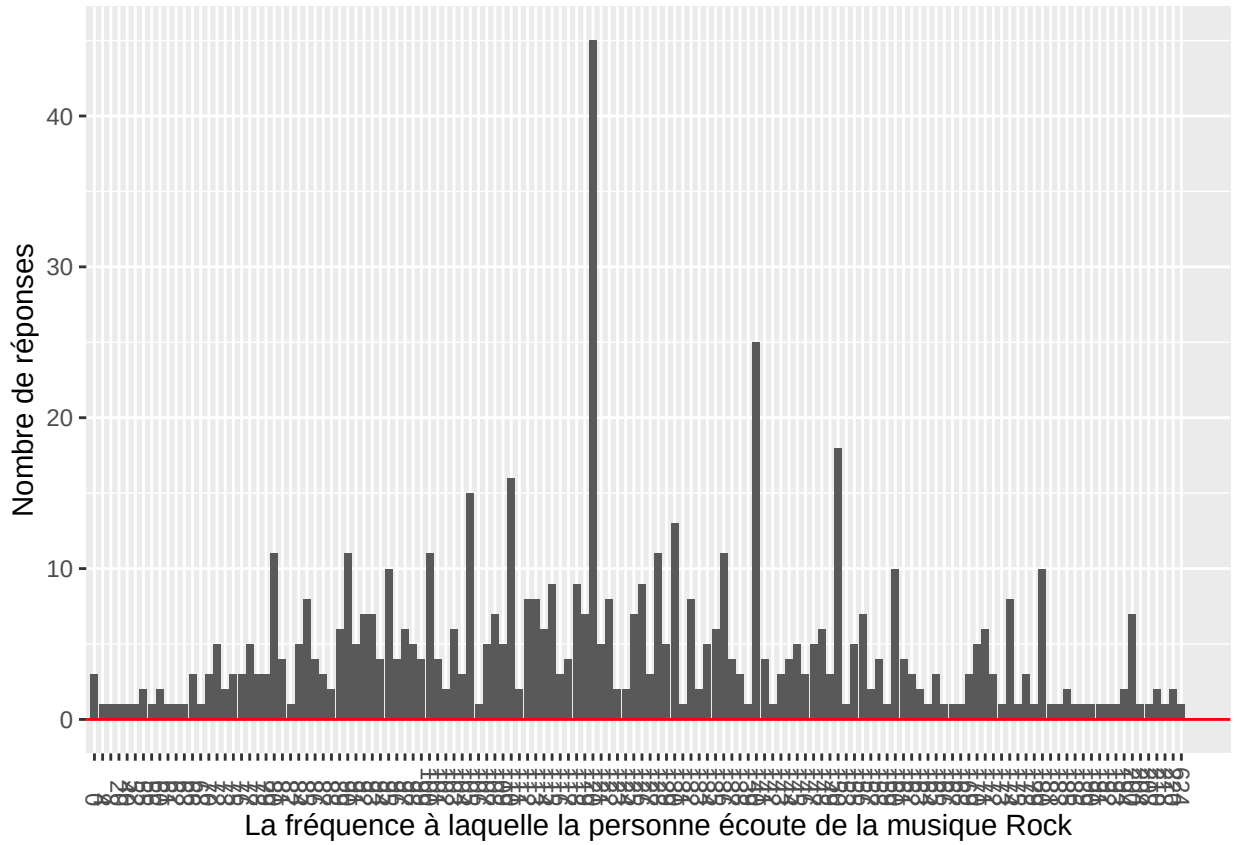
- *BPM* : discrète **1- Graphique répresentatif et résumés numériques**

```
bpm_counts<-table(resultat$BPM)
bpm_counts_df<- as.data.frame(bpm_counts)
#Ici on supprime la dernière ligne car elle contient 999999 beats / minute ce qui est une valeur imposs
bpm_counts_df <- bpm_counts_df[1:(nrow(bpm_counts_df)-1), ]
# Créer le diagramme en bâtons
colnames(bpm_counts_df) <- c( "Nombre de beats", "Nombres de réponses")

# Créer le graphique à barres
bar_plot <- ggplot(data.frame(bpm_counts_df),
                   aes(x = bpm_counts_df$`Nombre de beats`, y = bpm_counts_df$`Nombres de réponses`)) +
  geom_bar(stat = "identity")+ theme(axis.text.x = element_text(angle = -90, vjust = 0.5, hjust=1))+

  labs(x = "La fréquence à laquelle la personne écoute de la musique Rock", y = "Nombre de réponses", t
         scale_x_continuous(breaks = seq(0, 140, 10)))
```

```
#points(seq(0,140,0.5),dnorm(seq(0,140,0.5),mean(resultat$BPM),sd(resultat$BPM)),col=2,type="l")
print(bar_plot)
```



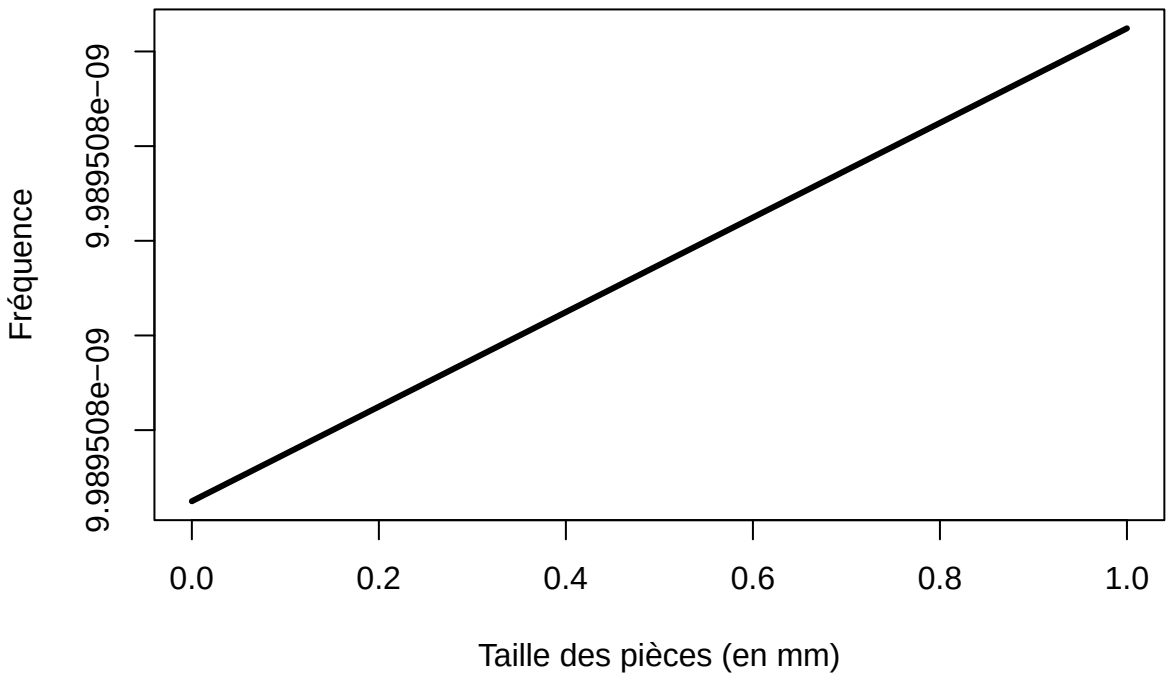

```
bar_plot <- bar_plot +
  geom_line(data = data.frame(x = seq(0, 140, 0.5)),
            aes(x = x, y = dnorm(x, mean(resultat$BPM), sd(resultat$BPM))),
            col = "red")

print(bar_plot)
```



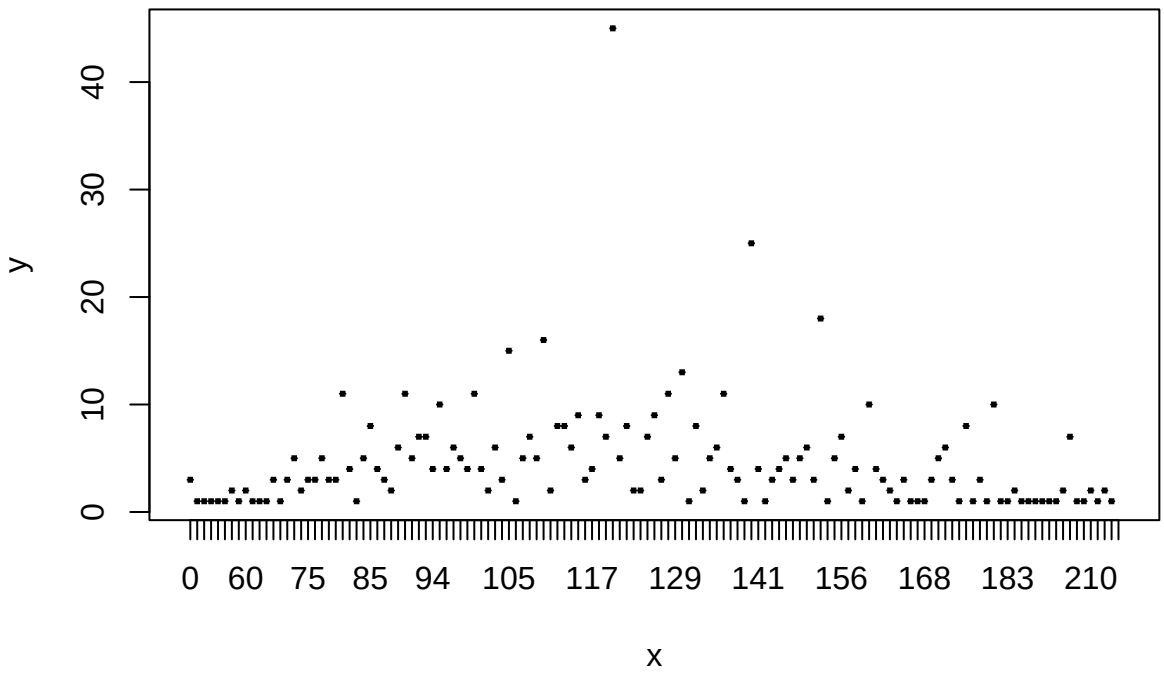

```
moy <-mean(resultat$BPM) # moyenne de la population.
median(bpm_counts_df$`Nombres de réponses`)
```

```
## [1] 3
```

```
std <- sd(resultat$BPM) # écart-type de la population
plot(function(x) dnorm(x,moy,std),main = "",lwd=3,xlab="Taille des pièces (en mm)",ylab="Fréquence")
```




```
plot(bpm_counts_df$`Nombre de beats`, bpm_counts_df$`Nombres de réponses`, type)
```



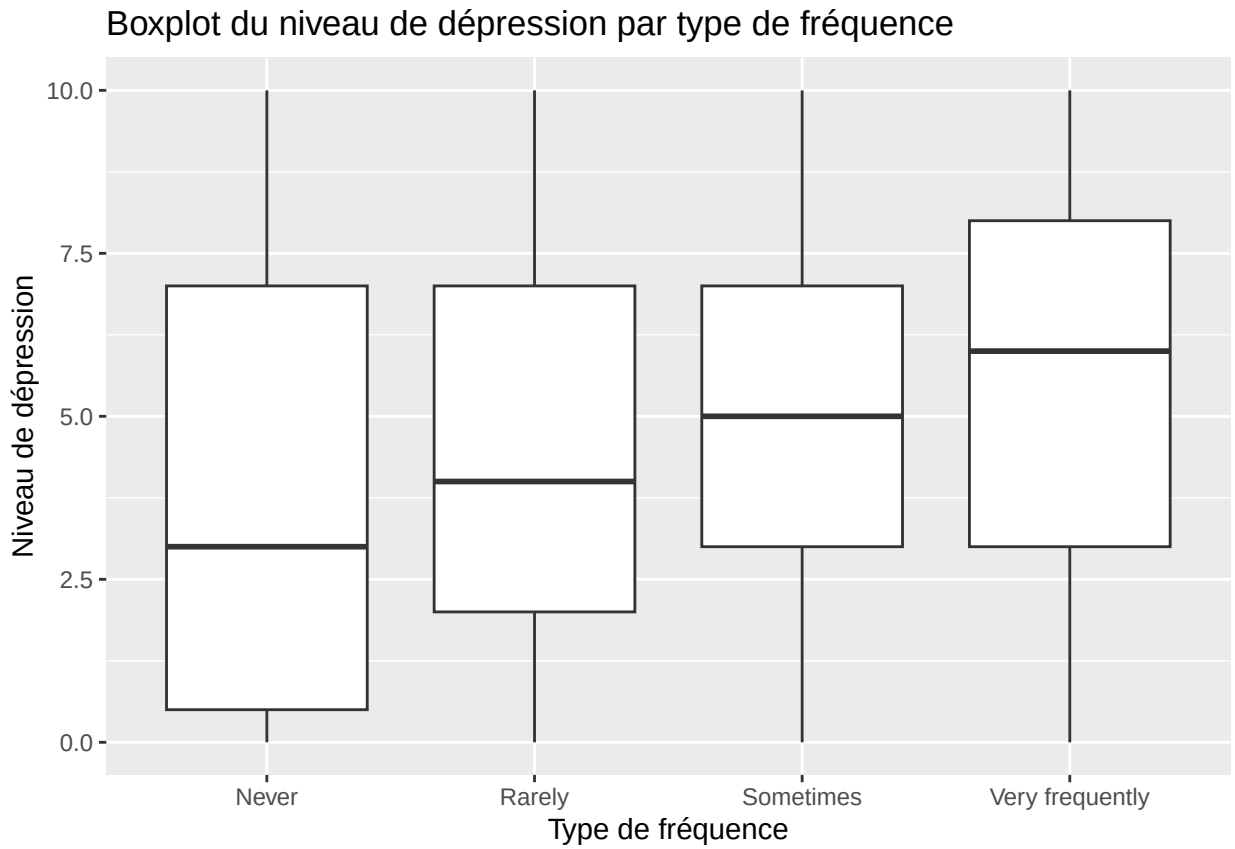
2- Etude du caractère gaussien
3- Les estimateurs de moyenne et variance 4- Intervalles de confiance de la moyenne

3 - Analyse multivariée

*1 - Analyse quanti x quali

1- Box plot Pour cette partie, nous allons répondre à cette question : Est-ce que le niveau de dépression des auditeurs dépend de l'écoute fréquente du genre musical Rock ? Pour cette analyse nous avons besoin d'avoir 4 sous population/groupe de la variable *Fréquence Rock*: - Premier groupe : 'Never' - Deuxième groupe : 'Rarely' - Troisième groupe :'Sometimes' - Quatrième groupe : 'Very frequently'

```
ggplot(resultat, aes(x = resultat$Frequency..Rock., y = resultat$Depression)) +
  geom_boxplot() +
  labs(x = "Type de fréquence", y = "Niveau de dépression",
       title = "Boxplot du niveau de dépression par type de fréquence")
```



Commentaires : pas de valeur extreme,... **2- Rapport de corrélation**

```
#Moyenne de niveau de dépression par groupe/sous population
moyenne1<- mean(resultat$Depression[resultat$Frequency..Rock.=='Never'])
moyenne2<- mean(resultat$Depression[resultat$Frequency..Rock.=='Very frequently'])
moyenne3<- mean(resultat$Depression[resultat$Frequency..Rock.=='Sometimes'])
moyenne4<- mean(resultat$Depression[resultat$Frequency..Rock.=='Rarely'])

#Variance de niveau de dépression par groupe/sous population
var1<- var(resultat$Depression[resultat$Frequency..Rock.=='Never'])
var2<- var(resultat$Depression[resultat$Frequency..Rock.=='Very frequently'])
var3<- var(resultat$Depression[resultat$Frequency..Rock.=='Sometimes'])
var4<- var(resultat$Depression[resultat$Frequency..Rock.=='Rarely'])

frequency_rock_counts<- table(resultat$Frequency..Rock.)
frequency_rock_counts_df <- as.data.frame(frequency_rock_counts)

moyennes<-c(moyenne1,moyenne4,moyenne3,moyenne2)
variances<-c(var1,var4,var3,var2)
frequency_rock_counts_df$Moyennes <- moyennes
frequency_rock_counts_df$`Variance dans chaque groupe` <- variances
colnames(frequency_rock_counts_df)<-c("Frequency","Nombre de reponses",
                                      "Moyennes","Variance dans chaque groupe")
print(frequency_rock_counts_df) #pour avoir le nombre de réponses
```

```
##         Frequency Nombre de reponses Moyennes Variance dans chaque groupe
```

```
## 1           Never                  83 3.698795                    9.895974
## 2          Rarely                  79 4.164557                    8.677702
## 3       Sometimes                 185 5.010811                    8.043361
## 4 Very frequently                 281 5.348754                    8.826150
```

```
# Calculer la moyenne pondérée
moyenne_ponderee <- sum(frequency_rock_counts_df$`Nombre de reponses` * frequency_rock_counts_df$Moyenne
                    sum(frequency_rock_counts_df$`Nombre de reponses`)
# Moyenne pondérée
print(moyenne_ponderee)
```

```
## [1] 4.882166
```

```
#Var(Y) : Variance totale sans utiliser le théorème de décomposition de la variance
variance<-var(resultat$Depression)
print(variance)
```

```
## [1] 9.028358
```

```
#Variance interclasse (B)
B<- sum(frequency_rock_counts_df$`Nombre de reponses` *
          (frequency_rock_counts_df$Moyennes - moyenne_ponderee )^ 2)/sum(frequency_rock_counts_df$`Nomb
print(B)
```

```
## [1] 0.3521484
```

```
#Variance intraclasse (W)
W<- sum(frequency_rock_counts_df$`Nombre de reponses` *
          (frequency_rock_counts_df $`Variance dans chaque groupe` ))/sum(frequency_rock_counts_df$`Nomb
print(W)
```

```
## [1] 8.718271
```

```
#
```

Après avoir calculé les variances intraclasses (W) et et les variances interclasses(B) nous pouvons appliquer le théorème de décomposition de la variance trouver la variance totale (V) de la variable `Depression` qui est égale à W + B:

```
V<- B +W
print(V)
```

```
## [1] 9.070419
```

Commentaires : Nous remarquons que V est égale à la variance calculée précédemment, ce qui prouve la justesse de nos calculs. Nous pouvons, maintenant, calculer le rapport de corrélation. Cet indicateur mesure la part de variabilité globale imputable aux différences de groupe.

```
rapport_correlation<- B/V
print(rapport_correlation)
```

```
## [1] 0.03882382
```

Le rapport de corrélation est proche de 0 alors il semble ne pas avoir de lien entre les deux variables ; autrement dit, la fréquence de l'écoute de musique rock ne donne pas d'information sur le niveau de dépression des personnes. **3- Test d'égalité des moyennes**

```
shapiro.test(resultat$Depression)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  resultat$Depression
## W = 0.94558, p-value = 2.024e-14
```

Le test de normalité de Shapiro-Wilk a été effectué sur la variable resultat$Depression. Voici comment interpréter les résultats :

```
W : La statistique de test de Shapiro-Wilk est de 0.94558. Cette valeur se situe entre 0 et 1, où une va
```

```
p-value : La valeur de p est de 2.024e-14, ce qui est extrêmement faible. La p-value est utilisée pour d
```

En résumé, les résultats du test de Shapiro-Wilk indiquent que la variable resultat$Depression ne suit pas une distribution normale, avec une p-value très faible. Cela signifie que les données ne sont pas symétriques et ne sont pas conformes à une distribution gaussienne.

Pour faire le test des moyennes, il faut d'abord calculer la variance pour savoir si on la prend en compte dans le test des moyennes. Nous avons décidé de comparer les moyennes de 'Very frequently' et 'Never'. Les hypothèses sont : - H0 les variances sont égales. - H1 les variances sont différentes.

```
var.test(resultat$Depression[resultat$Frequency..Rock.=='Never'], resultat$Depression[resultat$Frequency
```

```
##
##  F test to compare two variances
##
## data:  resultat$Depression[resultat$Frequency..Rock. == "Never"] and resultat$Depression[resultat$Fre
## F = 1.1212, num df = 82, denom df = 280, p-value = 0.4951
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.8036246 1.6181472
## sample estimates:
## ratio of variances
##           1.121211
```

Les résultats du test de variance indiquent qu'il n'y a pas de différence significative entre les variances des groupes 'Never' et 'Very frequently'et p-valeur élevée. Cela signifie que, pour effectuer un test de comparaison des moyennes entre ces deux groupes, nous pouvez considérer que les variances sont égales et nous rejetons H1.

On va faire le test des moyennes en prennant en compte la variance. Les hypothèses sont : - H0 les moyennes sont égales. - H1 les moyennes sont différentes.

```
t.test(resultat$Depression[resultat$Frequency..Rock.=='Never'], resultat$Depression[resultat$Frequency.
       var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  resultat$Depression[resultat$Frequency..Rock. == "Never"] and resultat$Depression[resultat$Fr
## t = -4.3858, df = 362, p-value = 1.517e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.3897832 -0.9101353
## sample estimates:
## mean of x mean of y
##  3.698795  5.348754
```

La p-value est de 1.517e-05, ce qui est extrêmement faible. Cela indique qu'il y a suffisamment de preuves pour rejeter l'hypothèse nulle selon laquelle les moyennes des deux groupes sont égales. Autrement dit, il y a une différence significative entre les moyennes des groupes 'Never' et 'Very frequently'. Sur la base des analyses effectuées et des résultats significatifs obtenus, il semble y avoir une association entre l'écoute intensive de rock et la santé mentale. *## 2 - Analyse quali x quali* **1- le tableau de contingence, proposer une représentation graphique** quali*quali:

```
#exploratory_plateform<- data.frame(resultat$Primary.streaming.service,resultat$Exploratory)
#tableau_contingence<- table(exploratory_plateform$resultat.Primary.streaming.service,exploratory_plate
#Tableau de contingence
#print(tableau_contingence)
#Représentation graphique
#colnames(df)<- c("Plateform", "Response", "Effectif")
#ggplot(df, aes(x =Plateform, y=Effectif, fill=Response))+
 # geom_bar(stat = "identity", position = "stack")+
  #labs(c="Plateforme de streaming de musique", y="Nombre de réponses", fill="Réponse")+
  #theme_minimal()

#Il faut commenter ça
```

Spotify offre un accès à une vaste bibliothèque de musique comprenant des millions de chansons provenant d'artistes du monde entier. Cette diversité musicale attire de nombreux utilisateurs qui peuvent trouver facilement leur musique préférée ainsi que de nouvelles découvertes. *## 3- Analyse quanti* quanti :*

```
# Créer des intervalles d'âges personnalisés
intervalles <- cut(resultat$Age, breaks = c(10, 14, 18,20, 22, 26, 30, 34, 38, 42, 46, 50, Inf), right =

tableau_intervalles <- table(intervalles)
age_df<- data.frame(tableau_intervalles)
print(age_df)
```

```
##    intervalles Freq
## 1      [10,14)    9
## 2      [14,18)  128
## 3      [18,20)  132
## 4      [20,22)   77
## 5      [22,26)   97
```

```
## 6      [26,30)   54
## 7      [30,34)   40
## 8      [34,38)   24
## 9      [38,42)   14
## 10     [42,46)   10
## 11     [46,50)    7
## 12    [50,Inf]   36
```

```
colnames(age_df) <- c("Ages", "Nombre de personnes")
age_df$Ages <- factor(age_df$Ages, levels = unique(age_df$Ages))
print(age_df$Ages)
```

```
##  [1] [10,14)  [14,18)  [18,20)  [20,22)  [22,26)  [26,30)  [30,34)  [34,38)
##  [9] [38,42)  [42,46)  [46,50)  [50,Inf]
## 12 Levels: [10,14) [14,18) [18,20) [20,22) [22,26) [26,30) [30,34) ... [50,Inf]
```

```
moyenne_bpm <- aggregate(resultat$BPM ~ intervalles, data = resultat, FUN = mean)
colnames(moyenne_bpm) <- c("Intervalles", "Moyenne BPM")
print(moyenne_bpm)
```

```
##    Intervalles  Moyenne BPM
## 1      [10,14)     116.1111
## 2      [14,18) 7812627.4531
## 3      [18,20)     120.1288
## 4      [20,22)     123.8701
## 5      [22,26)     119.8041
## 6      [26,30)     131.8519
## 7      [30,34)     124.3000
## 8      [34,38)     119.1250
## 9      [38,42)     132.5714
## 10     [42,46)     115.5000
## 11     [46,50)     131.1429
## 12    [50,Inf]     115.8056
```

```
#cor(intervalles,moyenne_bpm$`Moyenne BPM`)
```

```
#plot(resultat$Age, resultat$BPM,xlab = "Âge", ylab = "BPM",ylim = c(0, 180))
cor(resultat$Age, resultat$BPM)
```

```
## [1] -0.02995137
```

```
table = data.frame(resultat$Age, resultat$BPM)
colnames(table)[1] <- "Age";
colnames(table)[2] <- "BPM";
ggplot(data = table, aes(x = Age, y = BPM)) +
  ylim(0, 180) +
  geom_point() +
  geom_smooth(method = "lm", color = "orange")
```

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

```
## Warning: Removed 28 rows containing non-finite values (‘stat_smooth()‘).

## Warning: Removed 28 rows containing missing values (‘geom_point()‘).
```