Projet Calcul Financier & Statistique : En avant la musique !

MAHI Riad & Jérôme GAMBIEZ

Mise à jour, le 25 Mai 2023

Contents

1. Introduction

La source de notre jeu de donnée provient du site Kaggle.com:
https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube

Dernière mise à jour du jeu de donnée: Mars 2023
Création: Février 2023

Nous avons sélectionné, dans le cadre du projet Calcul Financier & Statistique, un jeu de données provenant des plateformes de streaming musical Spotify et YouTube. Nous avons spécifiquement choisi ce jeu de données en raison de l'impact qu'ont ces applications sur la vie quotidienne de millions d'utilisateurs à travers le monde. Étant donné que ces applications sont utilisées par une population provenant de divers pays, notre échantillon ne représente pas une zone géographique spécifique.

Question à mettre en place: - Est-ce l'energie et la dansabilité d'une musique ont une relation ? - Est-ce l'énergie d'une musique à un impact sur la sensibilité d'un utilisateur à aimer la musique. - Même question pour la dansabilité?

Mise en place du jeu de donnée:

```
dataPath <- "./Spotify_Youtube.csv"
data <- read.csv(dataPath, header=TRUE, stringsAsFactors=FALSE, nrows=50)
```

Les variables qualitatives sont les variables: - Si la musique à été publiée comme un album ou un single - La fréquence d'écoute d'un titre

1. Peu écouté : Cette catégorie désigne les chansons qui ont reçu un nombre de vues inférieur ou égal à 100 millions. Cela signifie que ces chansons ont été relativement moins populaires ou ont été moins diffusées en comparaison avec d'autres chansons.
2. Moyen écouté : Cette catégorie concerne les chansons qui ont reçu un nombre de vues compris entre 100 millions et 500 millions. Ces chansons peuvent être considérées comme ayant une popularité moyenne ou une diffusion modérée par rapport à d'autres chansons.

3. Très écouté : Cette catégorie comprend les chansons qui ont accumulé un nombre de vues supérieur à 500 millions. Ces chansons sont considérées comme ayant une grande popularité ou une diffusion élevée, attirant un grand nombre de spectateurs ou d'auditeurs.

Les variables quantitatives sont:

- La dancabilité d'une musique, qui est un indicateur sur la probabilité de dancer sur une musique

- L'énergie est une mesure allant de 0.0 à 1.0 et représente une mesure perceptuelle de l'intensité et de l'activité dans le domaine de la musique. Typiquement, les morceaux énergiques sont ressentis comme étant rapides, bruyants et animés. Par exemple, le death metal est caractérisé par une énergie élevée, avec ses rythmes rapides, ses guitares lourdes et ses voix agressives, tandis qu'un prélude de Bach est souvent considéré comme ayant une énergie plus basse, avec ses lignes mélodiques douces et son atmosphère plus calme.

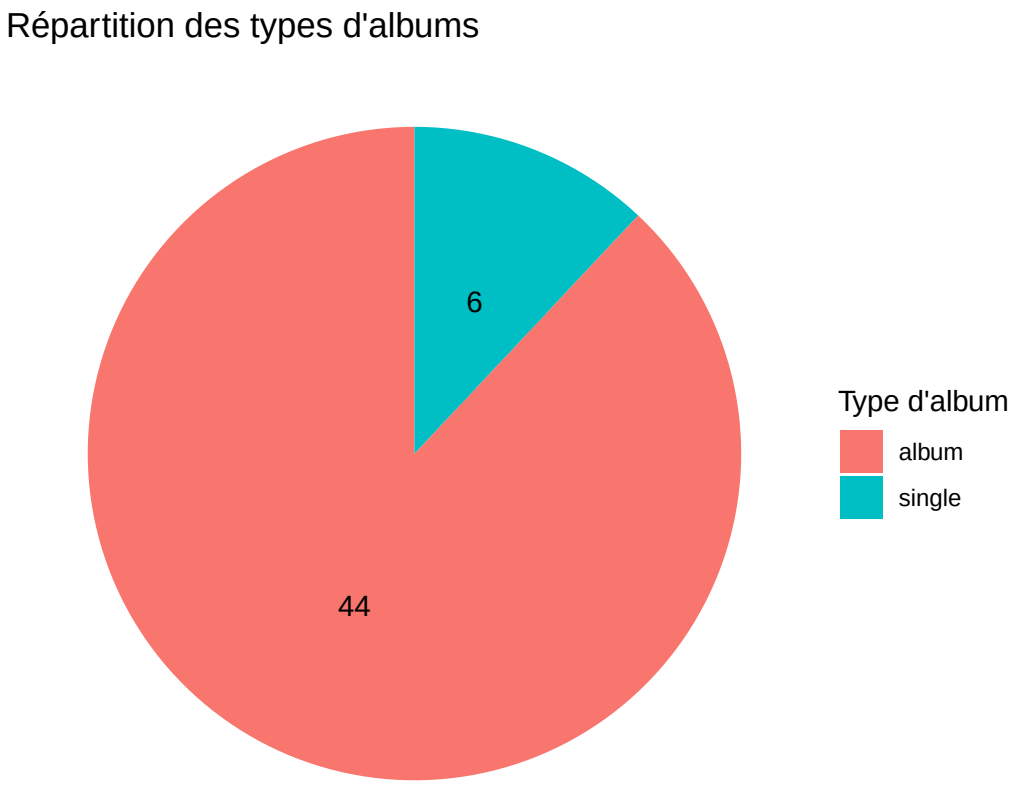
Les caractéristiques perceptuelles qui contribuent à cet attribut d'énergie incluent la plage dynamique, c'est-à-dire la variation entre les niveaux sonores forts et faibles, la loudness perçue, qui correspond à la sensation de volume sonore, le timbre qui donne la qualité sonore spécifique d'un instrument ou d'une voix, le taux de début qui fait référence à la rapidité avec laquelle les sons commencent et se développent, et la densité qui décrit la quantité et l'occupation de l'espace sonore par différents éléments.

En somme, l'énergie musicale est une mesure subjective qui nous aide à évaluer et à caractériser le niveau d'intensité et d'activité ressenties dans un morceau de musique.

2. Analyse univariée

Répartitions de la variable qualitative "type album"

```
library(ggplot2)

album_type <- table(data$Album_type)
df <- as.data.frame(album_type)
df <- df[order(df$Freq, decreasing = TRUE), ]
# Générer le pie chart
pie_chart <- ggplot(df, aes(x = "", y = Freq, fill = Var1)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Répartition des types d'albums", fill = "Type d'album") +
  theme_void() +
  geom_text(aes(label = Freq), position = position_stack(vjust = 0.5), size = 4)

# Afficher le pie chart
print(pie_chart)
```



Répartitions de la variable qualitative noms des artistes

```
library(dplyr)
```

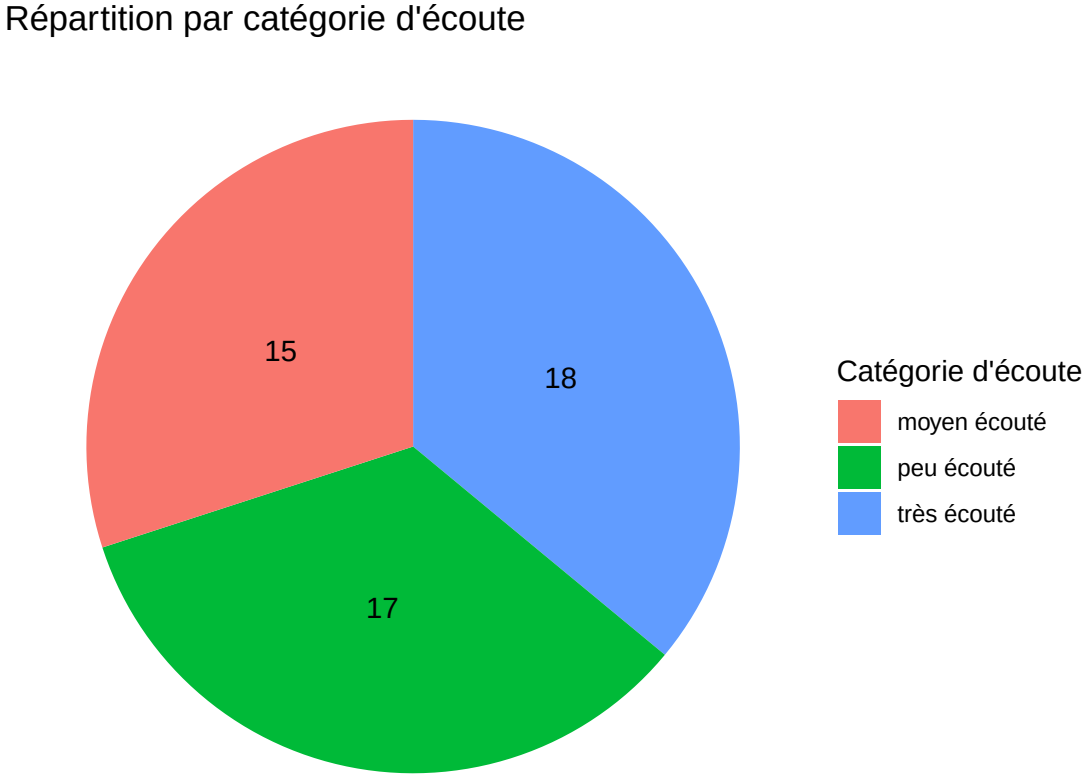
```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

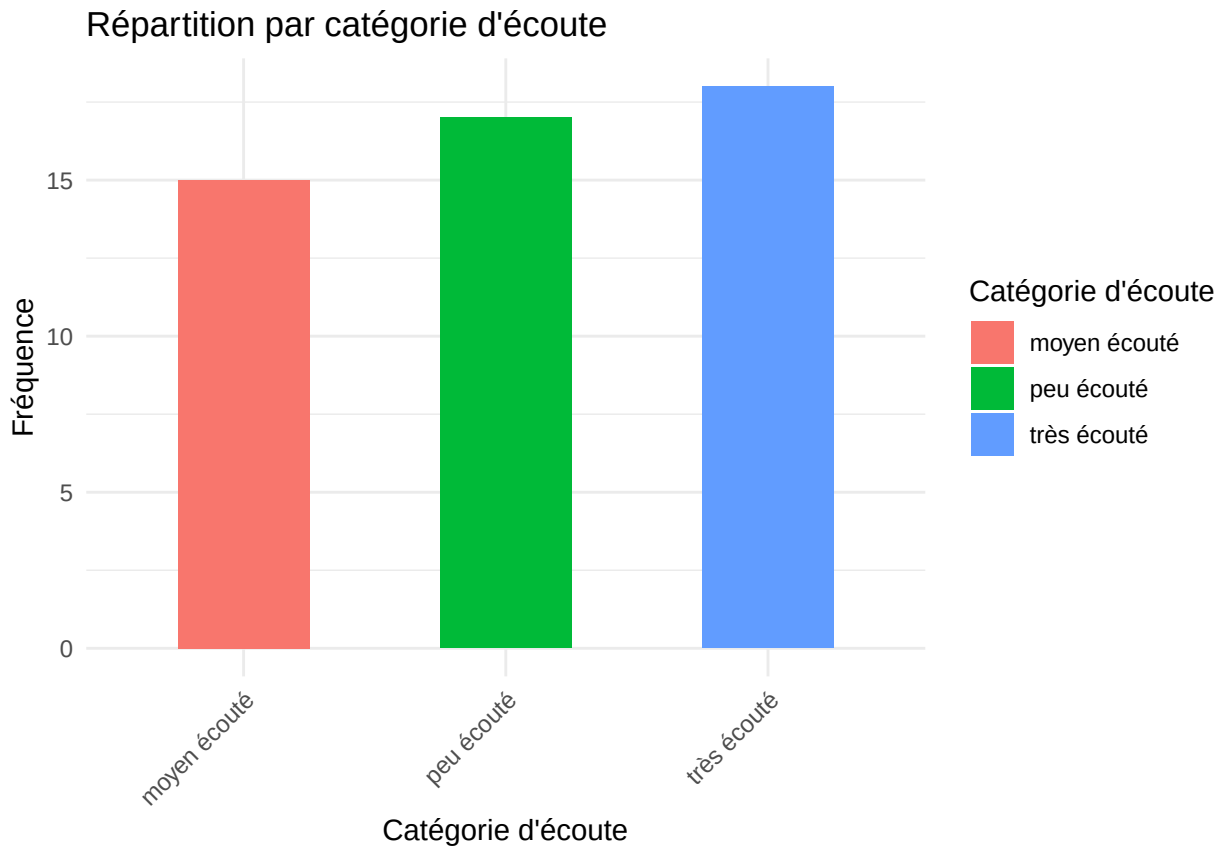
```
# Créer des catégories en fonction du nombre de vues
data <- data %>%
  mutate(Freq_listen = case_when(
    data$Views <= 100000000 ~ "peu écouté",
    data$Views > 100000000 & data$Views <= 500000000 ~ "moyen écouté",
    data$Views > 500000000 ~ "très écouté",
    TRUE ~ NA_character_
  ))

# Créer un tableau de comptage par catégorie d'écoute
```

```r
ecoute_counts <- table(data$Freq_listen)

# Convertir le tableau en data frame
ecoute_df <- as.data.frame(ecoute_counts)
ecoute_df <- ecoute_df[order(ecoute_df$Freq, decreasing = TRUE), ]

# Générer le bar plot
bar_plot <- ggplot(ecoute_df, aes(x = "", y = Freq, fill = Var1)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Répartition par catégorie d'écoute", fill = "Catégorie d'écoute") +
  theme_void() +
  geom_text(aes(label = Freq), position = position_stack(vjust = 0.5), size = 4)

# Afficher le bar plot
print(bar_plot)
```




```r
# Générer l'histogramme à barres
bar_histogram <- ggplot(ecoute_df, aes(x = Var1, y = Freq, fill = Var1)) +
  geom_bar(stat = "identity", width = 0.5) +
  labs(title = "Répartition par catégorie d'écoute", x = "Catégorie d'écoute", y = "Fréquence") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_discrete(name = "Catégorie d'écoute")
```

```
# Afficher l'histogramme à barres
print(bar_histogram)
```

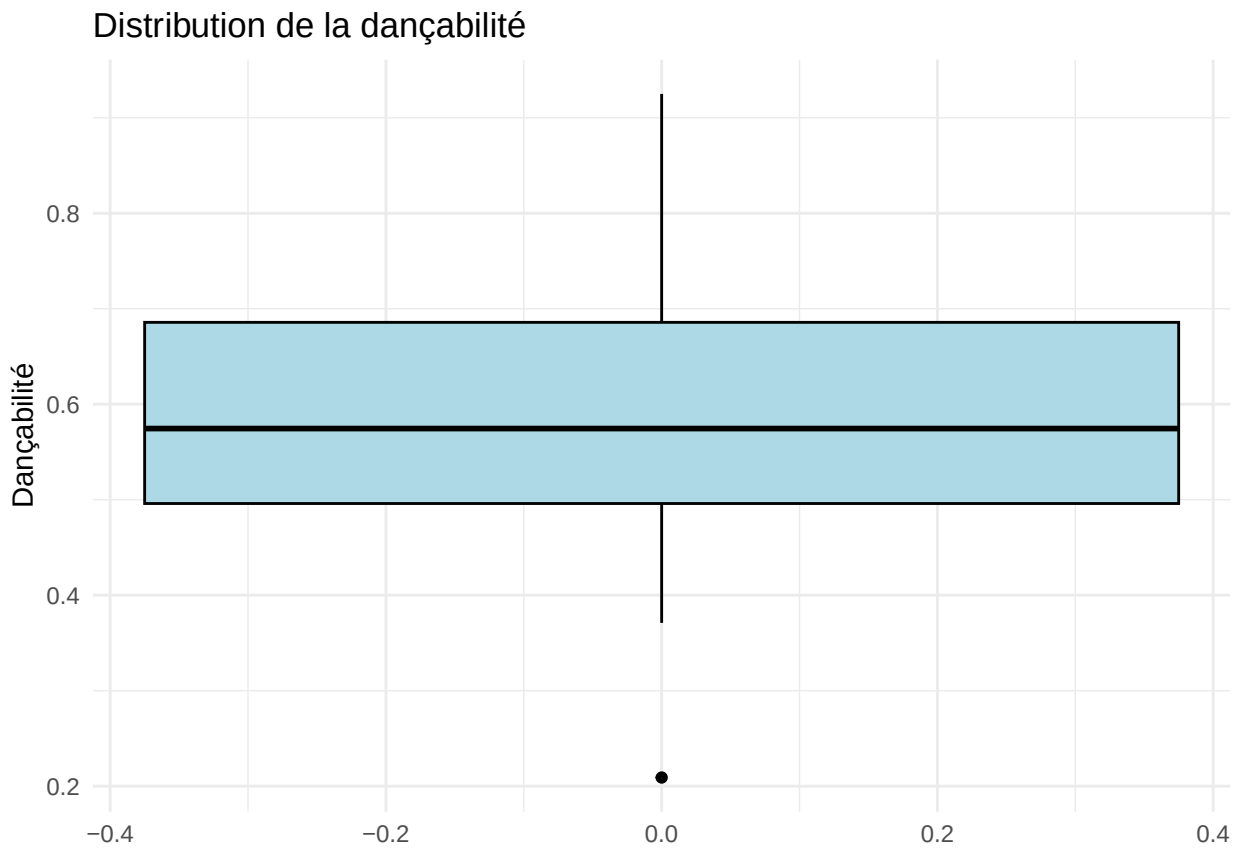


Représentation graphique de la répartition

Dancabilité

```
boxplot <- ggplot(data, aes(y = data$Danceability)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Distribution de la dançabilité", y = "Dançabilité") +
  theme_minimal()
print(boxplot)
```

```
## Warning: Use of ‘data$Danceability‘ is discouraged.
## i Use ‘Danceability‘ instead.
```



En examinant le diagramme, on peut constater que la majorité des musiques analysées, soit environ 58 %, ont une danceability élevée. Cela indique que ces morceaux sont adaptés à la danse et ont un potentiel élevé pour inciter les gens à se déplacer et à bouger sur la piste de danse.

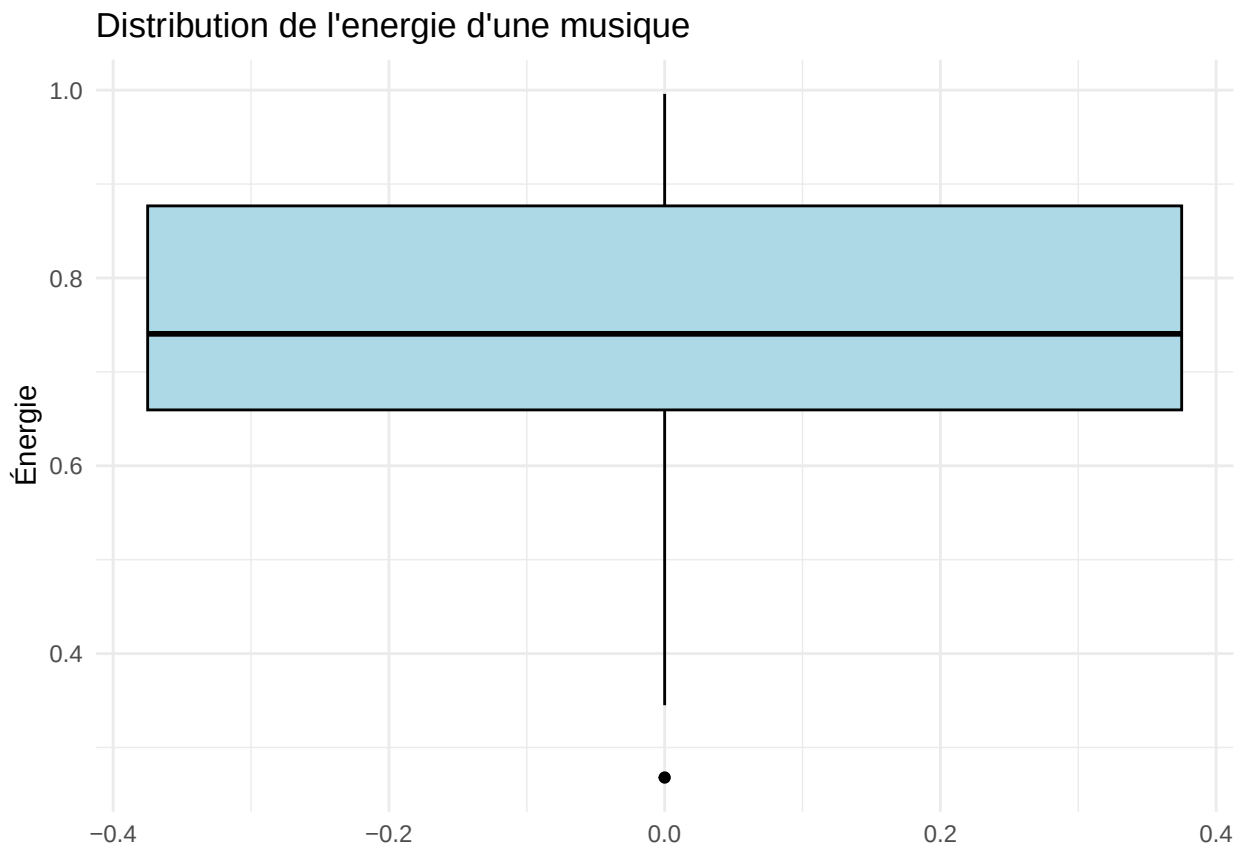
De plus, le diagramme révèle que 25 % des musiques ont un effet dansant sur environ 70 % de la population. Cela suggère que ces morceaux sont particulièrement accrocheurs et entraînants, capables d'engager un large public et de susciter l'envie de danser chez une majorité de personnes.

Enfin, il est intéressant de noter que 50 % des musiques ont un effet dansant sur une personne sur deux. Cela signifie que la danceability de ces morceaux est plus subjective et peut varier d'une personne à l'autre. Certains individus peuvent ressentir une forte envie de danser en les écoutant, tandis que d'autres peuvent ne pas être aussi réceptifs à leur effet dansant.

Énergie

```
boxplot <- ggplot(data, aes(y = data$Energy)) +
  geom_boxplot(fill = "lightblue", color = "black") +
  labs(title = "Distribution de l'energie d'une musique", y = "Énergie") +
  theme_minimal()
print(boxplot)
```

```
## Warning: Use of ‘data$Energy‘ is discouraged.
## i Use ‘Energy‘ instead.
```



L'analyse du diagramme en boîte à moustaches révèle plusieurs informations importantes concernant l'énergie des morceaux de musique. Plus l'énergie est proche de 1.0, plus elle est rapides, fortes et bruyantes.

En examinant les différentes parties du diagramme, nous pouvons constater que la médiane à 0.75. On peut donc conclure 75% des morceaux analysés présentent une énergie relativement élevée supérieur à 0.67.

La boîte à moustaches indique également que 25% des morceaux ont une énergie supérieure à 0,87, ce qui suggère une intensité, une vitesse et un niveau sonore plus élevés. Ces morceaux pourraient correspondre à des genres musicaux tels que le rock, le metal ou d'autres styles de musique agressifs et dynamiques. On retrouve des groupes comme Metallica ou Red Hot Chili Peppers.

D'autre part, 25% des morceaux ont une énergie inférieure à 0,66, ce qui indique une énergie plus basse. Ces morceaux pourraient être plus calmes, plus lents et moins bruyants. On retrouve le musicien Coldlay qui propose des musiques appelé Soft rock.

Conclusion des deux moustaches

En analysant les deux diagrammes ensemble, nous pouvons effectivement observer une corrélation entre la danceability et l'énergie des morceaux de musique. Il semble y avoir une tendance où les morceaux ayant une forte danceability ont également une énergie élevée, supérieure à 0,65.

En se basant sur cette observation, il est intéressant de noter que les musiques les plus dansantes sont celles de 50 Cent, un artiste de hip-hop/rap. Ses morceaux ont une énergie se situant entre 0,65 et 0,75, ce qui correspond à une intensité sonore élevée et une propension à inciter les auditeurs à se déplacer et à danser. ## Estimation et intervalles de confiance

Analyse de la dancabilité

```
dancabilite  <- data$Danceability
energie <- data$Energy
```

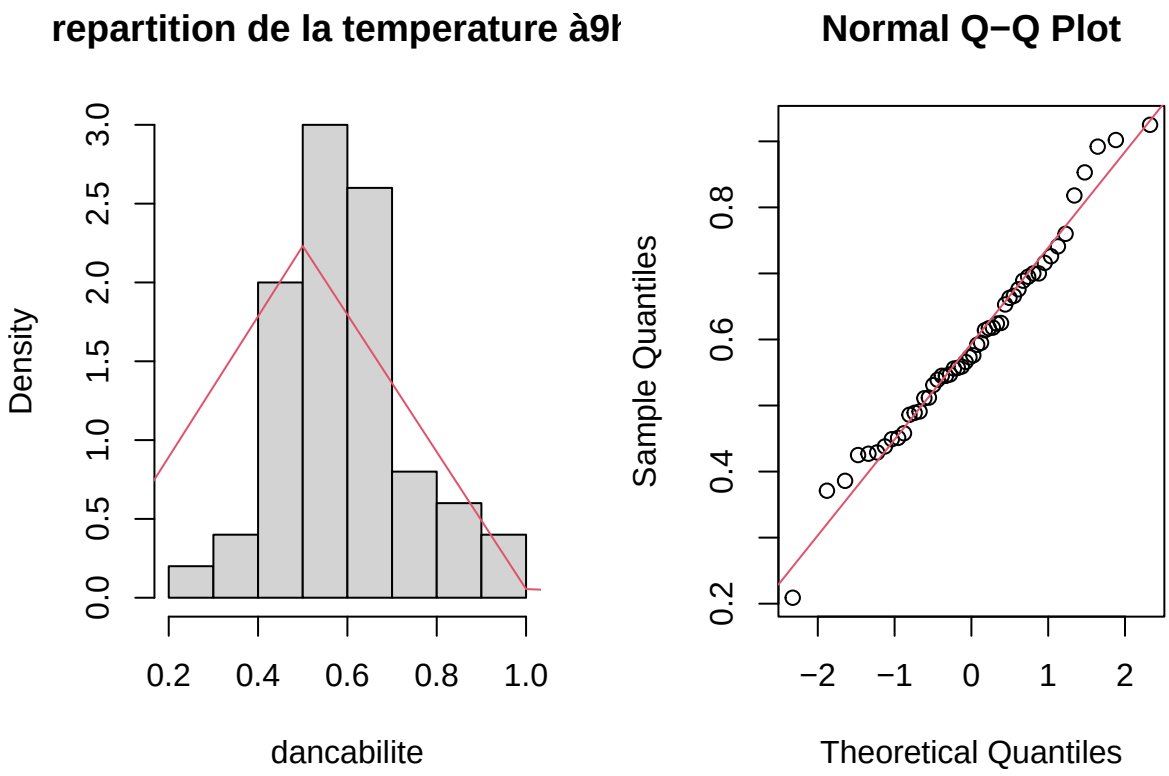
```
var(dancabilite)
```

```
## [1] 0.0210791
```

```
mean(dancabilite)
```

```
## [1] 0.59372
```

```
par(mfrow=c(1,2))
hist(dancabilite , main="repartition de la temperature à9h ",xlab="dancabilite",prob=T)
points(seq(0,30,0.5),dnorm(seq(0,30,0.5),mean(dancabilite),sd(dancabilite)),col=2,type="l")
qqnorm(dancabilite)
abline(mean(dancabilite),sd(dancabilite),col=2)
```




```
interval=t.test(dancabilite)
interval$conf.int
```

```
## [1] 0.5524585 0.6349815
## attr(,"conf.level")
## [1] 0.95
```

Analyse de l'energie

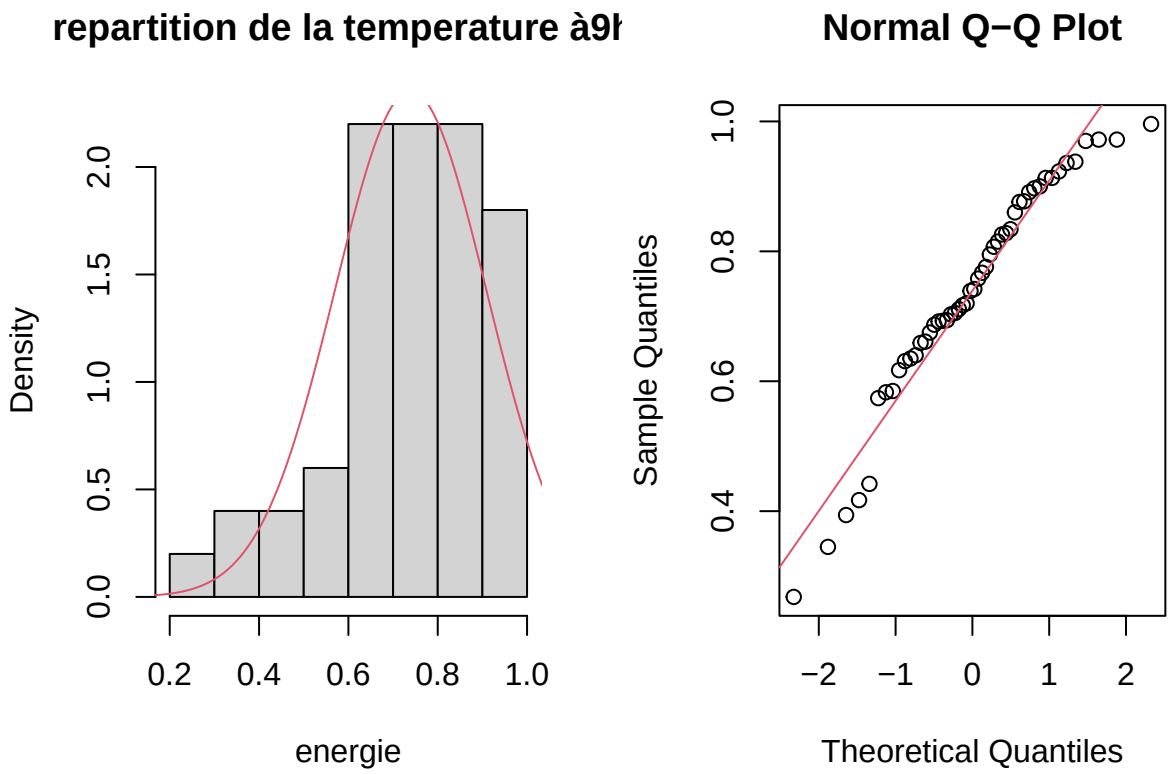
```r
var(energie)
```

```
## [1] 0.02873404
```

```r
mean(energie)
```

```
## [1] 0.73938
```

```r
par(mfrow=c(1,2))
hist(energie , main="repartition de la temperature à9h ",xlab="energie",prob=T)
points(seq(0,30,0.01),dnorm(seq(0,30,0.01),mean(energie),sd(energie)),col=2,type="l")
qqnorm(energie)
abline(mean(energie),sd(energie),col=2)
```




```r
interval=t.test(energie)
interval$conf.int
```

```
## [1] 0.6912055 0.7875545
## attr(,"conf.level")
## [1] 0.95
```