Projet Proba : Analyse des facteurs influençant la popularité des chansons sur Spotify

1. Introduction

La grande popularité des plateformes de streaming musical comme Spotify a généré une multitude de données sur les habitudes d'écoute et la popularité des chansons. L'analyse de ces données nous aide à comprendre ce qui fait le succès des chansons et peut éclairer les décisions de l'industrie musicale, les systèmes de recommandation et les expériences utilisateur personnalisées.

Ce projet vise à explorer les facteurs qui influencent la popularité des chansons sur Spotify à l'aide de Hypothesis testing et Regression analysis. On utilise un jeu de données de Kaggle pour examiner les relations entre les caractéristiques des chansons, les informations sur l'artiste, la date de sortie et différentes mesures de popularité.

2. Le jeu de données

Les données utilisées dans ce projet proviennent des sources suivantes :

Spotify - All Time Top 2000s Mega Dataset (kaggle.com)

- À propos du jeu de données :

Ce jeu de données contient des statistiques audio des 2000 meilleures pistes sur Spotify. Les données contiennent environ 15 colonnes décrivant chacune la piste et ses qualités. Les chansons sorties de 1956 à 2019 sont incluses par certains artistes notables et célèbres comme Queen, The Beatles, Guns N' Roses, etc.

```
> str(data)
'data.frame':   1994 obs. of  15 variables:
 $ Index                 : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Title                 : chr  "Sunrise" "Black Night" "Clint Eastwood" "The Pretender" ...
 $ Artist                : chr  "Norah Jones" "Deep Purple" "Gorillaz" "Foo Fighters" ...
 $ Top.Genre             : chr  "adult standards" "album rock" "alternative hip hop" "alternative metal" ...
 $ Year                  : int  2004 2000 2001 2007 2002 2004 2002 2006 2004 2002 ...
 $ Beats.Per.Minute..BPM.: int  157 135 168 173 106 99 102 137 148 112 ...
 $ Energy                : int  30 79 69 96 82 46 71 96 92 67 ...
 $ Danceability          : int  53 50 66 43 58 54 71 37 36 91 ...
 $ Loudness..dB.         : int  -14 -11 -9 -4 -5 -9 -6 -5 -4 -3 ...
 $ Liveness              : int  11 17 7 3 10 14 13 12 10 24 ...
 $ Valence               : int  68 81 52 37 87 14 54 21 23 66 ...
 $ Length..Duration.     : chr  "201" "207" "341" "269" ...
 $ Acousticness          : int  94 17 2 0 1 0 6 0 0 0 ...
 $ Speechiness           : int  3 7 17 4 3 2 3 14 8 7 ...
 $ Popularity            : int  71 39 69 76 59 45 74 69 77 82 ...
```

Contenu

- **Index :** ID
- **Title :** Nom de la piste
- **Artist :** Nom de l'artiste
- **Top Genre :** Genre de la chanson
- **Year :** Année de sortie de la chanson
- **Beats per Minute (BPM) :** Le tempo global estimé d'une piste en battements par minute (BPM). Dans la terminologie musicale, le tempo est la vitesse ou le rythme d'un morceau donné et dérive directement de la durée moyenne du battement.
- **Energy :** L'énergie est une mesure de 0 à 100 et représente une mesure perceptuelle de l'intensité et de l'activité.
- **Danceability :** La capacité à danser décrit dans quelle mesure une chanson est adaptée à la danse en fonction d'une combinaison d'éléments musicaux, notamment le tempo, la stabilité du rythme, la force du rythme et la régularité globale. Une valeur de 10 est la moins dansante et 100 est la plus dansante.
- **Loudness :** Volume global d'une piste en décibels (dB). Les valeurs d'intensité sonore sont moyennées sur l'ensemble de la piste et sont utiles pour comparer l'intensité sonore relative des pistes. L'intensité sonore est la qualité d'un son qui est le principal corrélat psychologique de la force physique (amplitude). Les valeurs varient généralement entre -60 et 0 dB.
- **Liveness :** Détecte la présence d'un public dans l'enregistrement. Des valeurs de vivacité plus élevées représentent une probabilité accrue que le chanson ait été joué en direct
- **Valence:** Une mesure de 0 à 100 décrivant la positivité musicale véhiculée par un morceau. Les pistes à valence élevée semblent plus positives (par exemple joyeuses, joyeuses, euphoriques), tandis que les pistes à faible valence semblent plus négatives (par exemple tristes, déprimées, en colère).
- **Length :** La durée de la chanson en secondes.

- **Acoustic :** Mesure de confiance de 0 à 100 indiquant si la piste est acoustique. 100 représente une grande confiance dans le fait que la piste est acoustique.
- **Speechiness :** Speechiness détecte la présence de mots prononcés dans une piste. Plus l'enregistrement ressemble exclusivement à de la parole (par exemple, talk-show, livre audio, poésie), plus la valeur de l'attribut est proche de 100. Les valeurs supérieures à 66 décrivent des pistes probablement entièrement composées de paroles. Les valeurs comprises entre 33 et 66 décrivent des pistes pouvant contenir à la fois de la musique et de la parole, soit en sections, soit en couches, y compris dans des cas tels que la musique rap. Les valeurs inférieures à 33 représentent très probablement de la musique et d'autres pistes non vocales.
- **Popularity :** Plus la valeur est élevée, plus la chanson est populaire.

3. L'analyse exploratoire des données (EDA)

L'EDA est une étape cruciale dans la compréhension des caractéristiques de l'ensemble de données, l'identification des problèmes potentiels et l'obtention d'informations qui guident une analyse plus approfondie. Tout d'abord, on vérifie les valeurs manquantes dans l'ensemble de données à l'aide de la fonction is.na(data) :

```
> total_missing <- sum(is.na(data))
> print(paste0("Total missing values in the dataset: ", total_missing))
[1] "Total missing values in the dataset: 0"
```

Il n'existe pas des NA valeurs dans l'ensemble de données.
Ensuite, on renomme les colonnes "Durée", "BPM", "Loudness" pour un traitement plus facile ultérieurement :

```
> data <- rename(data,
+                "Duration" = "Length..Duration.",
+                "BPM" = "Beats.Per.Minute..BPM.",
+                "Loudness_db" = "Loudness..dB.")
>
```

La "Duration" est lue comme un type de données différent (caractère ou facteur), ce qui serait inapproprié pour les calculs numériques, on utilise la fonction as.integer() pour convertir cette colonne en type de données entier. Il existe des valeurs NA après la conversion, on les supprime donc de l'ensemble de données.

```
> data$Duration <- as.integer(data$Duration)
Warning message:
NAs introduced by coercion
> total_missing <- sum(is.na(data))
> print(paste0("Total missing values in the dataset: ", total_missing))
[1] "Total missing values in the dataset: 4"
> data <- na.omit(data)
```

Ensuite, nous traitons avec “Top Genre”, qui est composé de valeurs catégoriques et de nombreux types (album rock, art pop, british folk, adults standards, etc.)

```
> top_genre_counts <- table(data$Top.Genre) %>%
+     sort(decreasing = TRUE)
>
> print(top_genre_counts)

               album rock          adult standards                dutch pop          alternative rock
                      411                      123                       88                       86
              dance pop                dutch indie          alternative metal              classic rock
                       83                       75                       70                       51
             dance rock              dutch cabaret                glam rock             modern rock
                       51                       51                       49                       49
                      pop                art rock           permanent wave          british invasion
                       47                       40                       38                       36
             irish rock              british soul                 europop          classic uk pop
                       34                       31                       27                       22
                   disco              dutch rock               glam metal             neo mellow
                       18                       18                       17                       17
       alternative dance                art pop               blues rock          dutch hip hop
                       15                       14                       14                       13
                    funk          dutch americana                 big beat                britpop
                       13                       12                       11                       11
           classic soul              mellow gold           carnaval limburg          arkansas country
                       11                       11                       10                        9
                chanson               german pop             australian pop             belgian rock
                        9                        9                        8                        8
                  blues                     folk                   reggae       alternative pop rock
                        8                        8                        8                        7
               big room              canadian folk              celtic rock             chamber pop
                        7                        7                        7                        7
         detroit hip hop                 electro           modern folk rock             baroque pop
                        7                        7                        7                        6
               boy band          brill building pop          east coast hip hop          canadian pop
                        6                        6                        6                        5
                 celtic         classic country pop                dutch prog                 g funk
                        5                        5                        5                        5
           acoustic pop                belgian pop             british folk                downtempo
                        4                        4                        4                        4
             dutch metal                 folk-pop              garage rock             metropopolis
                        4                        4                        4                        4
          reggae fusion           australian rock             barbadian pop          bubblegum pop
```

On regroupe les genres dans les 7 genres les plus populaires et une catégorie “Others” en utilisant les fonctions mutate() et case_when(). Si un genre correspond à l'une des chaînes spécifiées, il est remplacé par un nouveau nom de genre correspondant (comme "adult.standards", "soul", etc.). Tous les autres genres sont regroupés dans la catégorie “Others”.

```
> data <- mutate(data,
+                Top.Genre = case_when(
+                    str_detect(Top.Genre, "adult standards") ~ "adult standards",
+                    str_detect(Top.Genre, "soul") ~ "soul",
+                    str_detect(Top.Genre, "alternative") ~ "alternative",
+                    str_detect(Top.Genre, "dance") ~ "dance",
+                    str_detect(Top.Genre, "indie") ~ "indie",
+                    str_detect(Top.Genre, "hip hop|rap") ~ "hip hop/rap",
+                    str_detect(Top.Genre, "rock|prog") ~ "rock",
+                    str_detect(Top.Genre, "pop") ~ "pop",
+                    TRUE ~ "Others"
+                )
+ )
> top_genre_counts <- table(data$Top.Genre) %>%
+     sort(decreasing = TRUE)
>
> print(top_genre_counts)

          rock          Others             pop     alternative           dance adult standards           indie
           710             376             302             187             138             123              81
          soul     hip hop/rap
            45              28
> |
```

On convertit la colonne “Top Genre” en facteur et la renivelle pour que “Others” soit le niveau de référence. Cela se fait à l'aide des fonctions factor() et relevel(). On a décidé de choisir "Others" comme le référence pour les variables dummies. La raison est que "Others" est un groupe de genres mineurs, ce qui peut conduire à des résultats incorrects pour le modèle, que j'ai regroupé en un seul. Cette étape prépare à la création de variables dummies, car elle permet de spécifier quel niveau doit être la référence par rapport à laquelle les autres niveaux sont comparés.

```
#Create dummies variables for "Top Genre" by choosing "others" as the reference level
data$Top.Genre <- relevel(factor(data$Top.Genre), ref = "others")

# Create the model matrix
genre_dummies <- model.matrix(~ Top.Genre, data = data)[,-1]

# Add dummy variables to the main data frame
data <- cbind(data, genre_dummies)

data <- data[data$Top.Genre != "others", ]
```

Après cela, on crée une matrice de modèle de variables dummies pour la colonne “Top Genre”. La première colonne de la matrice, qui représente le niveau de référence “Others”, est supprimée. Les variables dummies sont ensuite ajoutées à l'ensemble de données principal.

Pour l'année de sortie, on crée les nouvelles colonnes pour calculer les années depuis la sortie.

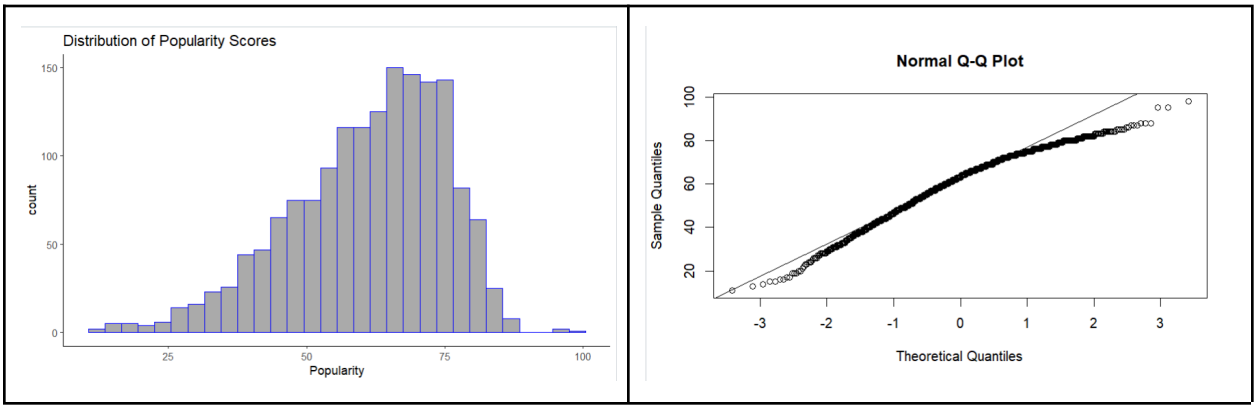
```
> followers<-read.csv(file.choose(),header = TRUE,dec = ".")
> data <- merge(data, followers, by = "Artist", all.x = TRUE)
```

Pour l'artiste, on importe le nombre de followers sur Spotify à partir d'un autre jeu de données et les ai fusionnés par noms d'artiste.

```
> followers<-read.csv(file.choose(),header = TRUE,dec = ".")
> data <- merge(data, followers, by = "Artist", all.x = TRUE)
```

4. Tests statistiques

Après avoir effectué un nettoyage des données, j'ai étudié la variable cible : Popularity scores. On analyse la distribution de la variable de popularité dans les données en utilisant l'histogramme, le tracé QQ (quantile-quantile) et le test de Shapiro Wilk.




```
> shapiro_test_result <- shapiro.test(data$Popularity)
> print(shapiro_test_result)

        Shapiro-Wilk normality test

data:  data$Popularity
W = 0.96753, p-value < 2.2e-16
```

Les résultats montrent que, avec la valeur p était trop petite (valeur p < 2,2e-16), il est peu probable que les données soient distribuées normalement.
Pour résoudre ce problème, on prend un échantillon de 100 observations de la popularité et cela suggère que les données de l'échantillon suivent la distribution normale (valeur p = 0.0747 > 0.05). On utilise cet échantillon pour le reste du projet.

```
> shapiro_test_result <- shapiro.test(data$Popularity)
>
> # Print the test results
> print(shapiro_test_result)

        Shapiro-Wilk normality test

data:  data$Popularity
W = 0.9768, p-value = 0.0747
```

Les statistiques descriptives pour chaque variable de l'ensemble de données, en fonction du type de variable :

```
> summary(data)
      BPM            Energy       Danceability    Loudness_db        Liveness        Valence
 Min.   : 68.0   Min.   : 5.00   Min.   :20.00   Min.   :-21.00   Min.   : 3.00   Min.   : 3.00
 1st Qu.:100.8   1st Qu.:40.50   1st Qu.:45.00   1st Qu.:-12.00   1st Qu.:10.00   1st Qu.:32.00
 Median :122.5   Median :63.50   Median :56.00   Median : -9.00   Median :14.00   Median :54.50
 Mean   :120.8   Mean   :57.81   Mean   :54.47   Mean   : -9.41   Mean   :21.89   Mean   :52.51
 3rd Qu.:136.5   3rd Qu.:77.00   3rd Qu.:64.25   3rd Qu.: -6.00   3rd Qu.:28.00   3rd Qu.:76.00
 Max.   :200.0   Max.   :97.00   Max.   :95.00   Max.   : -3.00   Max.   :99.00   Max.   :98.00
    Duration      Acousticness     Speechiness      Popularity    Top.Genreadult.standards
 Min.   :122.0   Min.   : 0.00   Min.   : 2.00   Min.   :17.00   Min.   :0.00
 1st Qu.:212.8   1st Qu.: 3.75   1st Qu.: 3.00   1st Qu.:49.00   1st Qu.:0.00
 Median :256.0   Median :24.50   Median : 4.00   Median :62.00   Median :0.00
 Mean   :276.0   Mean   :32.98   Mean   : 5.62   Mean   :60.07   Mean   :0.13
 3rd Qu.:298.8   3rd Qu.:64.50   3rd Qu.: 5.00   3rd Qu.:70.00   3rd Qu.:0.00
 Max.   :811.0   Max.   :98.00   Max.   :38.00   Max.   :95.00   Max.   :1.00
 Top.Genrealternative Top.Genredance Top.Genrehiphop.rap Top.Genreindie  Top.Genrepop  Top.Genrerock
 Min.   :0.00         Min.   :0.00   Min.   :0.00        Min.   :0.00   Min.   :0.00   Min.   :0.00
 1st Qu.:0.00         1st Qu.:0.00   1st Qu.:0.00        1st Qu.:0.00   1st Qu.:0.00   1st Qu.:0.00
 Median :0.00         Median :0.00   Median :0.00        Median :0.00   Median :0.00   Median :0.00
 Mean   :0.09         Mean   :0.06   Mean   :0.05        Mean   :0.04   Mean   :0.13   Mean   :0.48
 3rd Qu.:0.00         3rd Qu.:0.00   3rd Qu.:0.00        3rd Qu.:0.00   3rd Qu.:0.00   3rd Qu.:1.00
 Max.   :1.00         Max.   :1.00   Max.   :1.00        Max.   :1.00   Max.   :1.00   Max.   :1.00
 Top.Genresoul  Years_since_release Total.Followers
 Min.   :0.00   Min.   : 5.00       Min.   :   11979
 1st Qu.:0.00   1st Qu.:17.75       1st Qu.:  559808
 Median :0.00   Median :36.50       Median : 1928984
 Mean   :0.02   Mean   :34.16       Mean   : 5734801
 3rd Qu.:0.00   3rd Qu.:50.00       3rd Qu.: 5689252
 Max.   :1.00   Max.   :65.00       Max.   :62743762
```

Ensuite, on applique un test t de Welch Two Sample, qui est un test statistique utilisé pour déterminer si deux moyennes de population sont égales, à deux ensembles de données, “pop_popularity” et “rock_popularity” dans ce cas.

- Hypothèse nulle (H0) : La popularité moyenne des chansons pop est égale à la popularité moyenne des chansons rock.
- Hypothèse alternative (H1) : La popularité moyenne des chansons pop n’est pas égale à la popularité moyenne des chansons rock.

```
> t_test_result <- t.test(pop_popularity, rock_popularity)
> t_test_result

        Welch Two Sample t-test

data:  pop_popularity and rock_popularity
t = -0.72909, df = 15.804, p-value = 0.4766
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -14.382475   7.026705
sample estimates:
mean of x mean of y
 56.38462  60.06250
```

t = -0,72909 : Il s'agit de la statistique t, qui est une mesure de la différence entre les deux moyennes par rapport à la variation des données. Le signe négatif indique que la moyenne du premier groupe (pop_popularity) est inférieure à la moyenne du deuxième groupe (rock_popularity).

Valeur p = 0,4766 > 0,05 : La valeur p signifie que nous ne rejetons pas l'hypothèse nulle. Dans ce cas, la valeur p est de 0,4766, nous n'avons donc pas suffisamment de preuves pour affirmer que les moyennes de pop_popularity et rock_popularity sont différentes.

Intervalle de confiance à 95 % : -14,382475 7,026705 : Il s'agit de la plage de valeurs dans laquelle nous pouvons être sûrs à 95 % que se situe la véritable différence entre les moyennes des deux populations. Puisque cet intervalle contient zéro, il conforte la conclusion selon laquelle les deux moyennes ne sont pas significativement différentes.

moyenne de x 56,38462 et moyenne de y 60,06250 : ce sont les exemples de moyennes de pop_popularity et rock_popularity, respectivement. La popularité moyenne pour le genre « pop » est d'environ 56,38, et pour le genre « rock », elle est d'environ 60,06.

On utilise le test d'analyse de variance (ANOVA), qui est un test statistique utilisé pour déterminer s'il existe des différences significatives entre les moyennes de trois groupes ou plus. Dans ce cas, le test est appliqué à la variable Popularité dans différents Top.Genre.

- Hypothèse nulle (H0) : la popularité moyenne est égale selon les différents genres.
- Hypothèse alternative (H1) : La popularité moyenne n'est pas égale selon les genres.

```
> anova_result <- aov(Popularity ~ Top.Genre, data = data_sample)
> summary(anova_result)
             Df Sum Sq Mean Sq F value Pr(>F)
Top.Genre     7   3744   534.9   3.346 0.0032 **
Residuals    92  14708   159.9
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Dans ce cas, la valeur p = 0,0032 < 0.05, on disposait donc de suffisamment de preuves pour affirmer que la popularité moyenne n'est pas égale selon les genres.
On teste les hypothèses de ANOVA :
A l'aide de Shapiro-Wilk test et Breusch Pagan test, il semble que les hypothèses de normalité et constante variance sont remplies.

```
> shapiro.test(anova_result$residuals)

        Shapiro-Wilk normality test

data:  anova_result$residuals
W = 0.98459, p-value = 0.2966
```

```
> bptest(anova_result)

        studentized Breusch-Pagan test

data:  anova_result
BP = 8.4547, df = 7, p-value = 0.2942
```

5. Construction de modèles

On supprime les variables inutiles du modèle :

```
data <- data %>% select(-Artist, -Index, -Title, -Top.Genre, -X, -Year)
```

5.1. Modèle 1 : Simple Linear Regression

On crée le premier modèle de régression linéaire simple, On a d'abord étudié quelle fonction avait la relation linéaire la plus forte avec "Popularity" :

```
> # Calculate correlation coefficients
> correlations <- sapply(data[numeric_cols], function(x) cor(x, data$Popularity, use = "pairwise.complete.ob
s"))
>
> # Sort correlations in descending order
> sorted_correlations <- sort(correlations, decreasing = TRUE)
>
> # Print sorted correlations
> print(sorted_correlations)
          Popularity      Total.Followers  Years_since_release              Valence
        1.0000000000         0.3314673105         0.2029844751         0.1736854050
         Speechiness          Loudness_db Top.Genreadult.standards        Top.Genresoul
        0.1585202667         0.1431248094         0.1359135114         0.1149461414
      Top.Genredance         Danceability  Top.Genrehiphop.rap Top.Genrealternative
        0.0885920901         0.0728677236         0.0629947427         0.0575435046
              Energy       Top.Genrerock                  BPM         Acousticness
        0.0008071290        -0.0005304598        -0.0062981454        -0.0442692848
        Top.Genrepop             Duration             Liveness       Top.Genreindie
       -0.1048738336        -0.1283688228        -0.1399646155        -0.3955055110
>
```

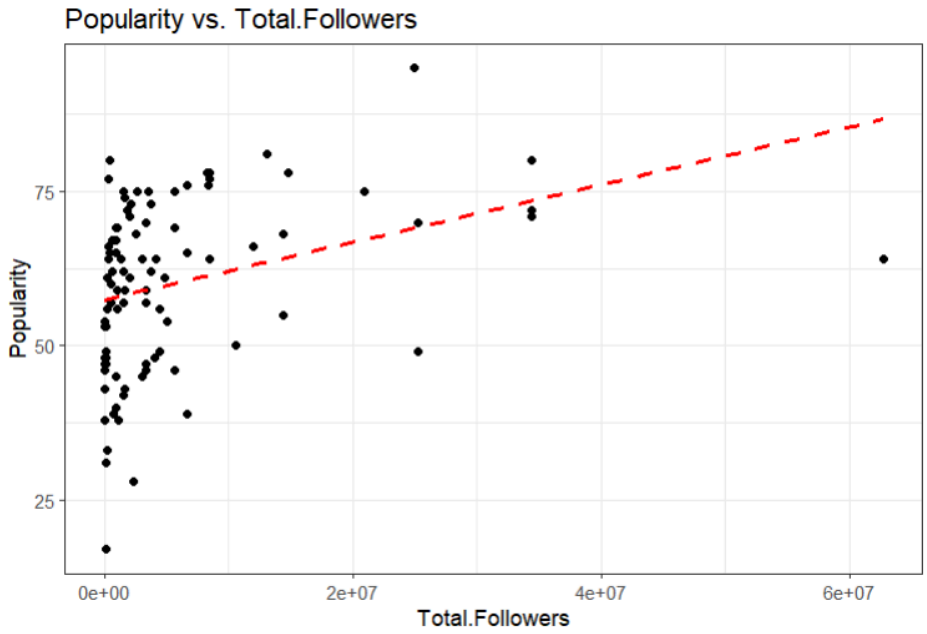
À partir de ce résultat, on peut voir que "Total.Followers" a la relation linéaire positive la plus forte avec "Popularity" (coefficient de corrélation = 0,331) et que "Top.Genreindie" a la relation linéaire négative la plus forte avec "Popularity" (coefficient de corrélation = -0,395).

```
> # Create the simple linear regression model
> model_1 <- lm(Popularity ~ Total.Followers, data = data)
>
> # Print the summary of the model
> summary(model_1)

Call:
lm(formula = Popularity ~ Total.Followers, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-40.440  -9.711   2.353   9.184  25.939

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     5.739e+01  1.506e+00  38.110  < 2e-16 ***
Total.Followers 4.666e-07  1.342e-07   3.478 0.000755 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 12.95 on 98 degrees of freedom
Multiple R-squared:  0.1099,    Adjusted R-squared:  0.1008
F-statistic:  12.1 on 1 and 98 DF,  p-value: 0.0007551
```

La valeur R^2 est de 0,1099, ce qui signifie qu'environ 11 % de la variabilité de la “Popularité” peut être expliquée par le “Total.Followers”. La valeur R^2 est assez faible, ce qui indique que "Total.Followers" n'explique qu'une petite partie de la variabilité de "Popularity".

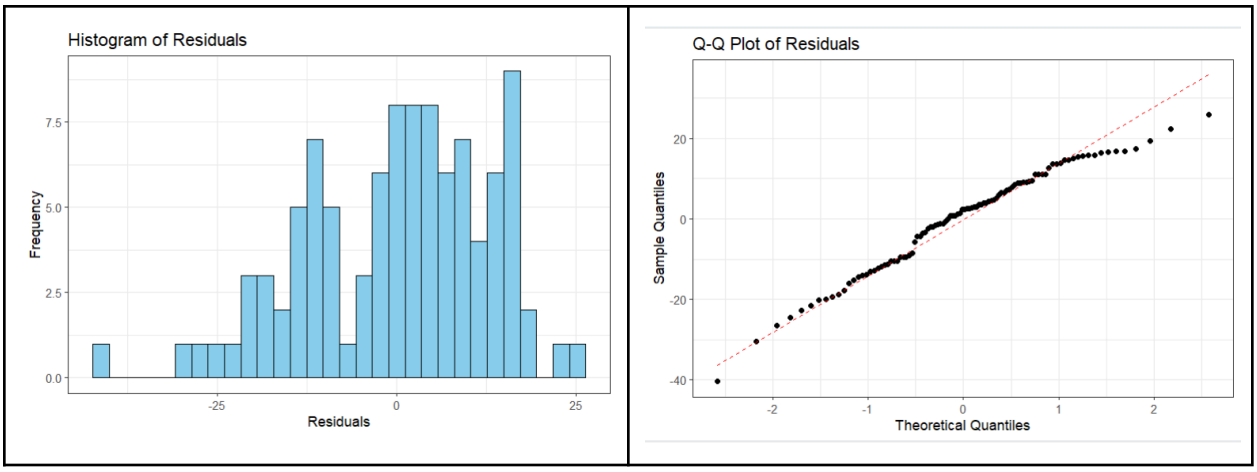
On teste les hypothèses du modèle :

1. **Linéarité**



En observant la figure, on fait une remarque que la relation linéaire uniquement entre la popularité et la variable indépendante 'Total Followers" n'est pas très claire et précise. Il semble que l'hypothèse de Linéarité est violée.

2. **Normalité des résidus**



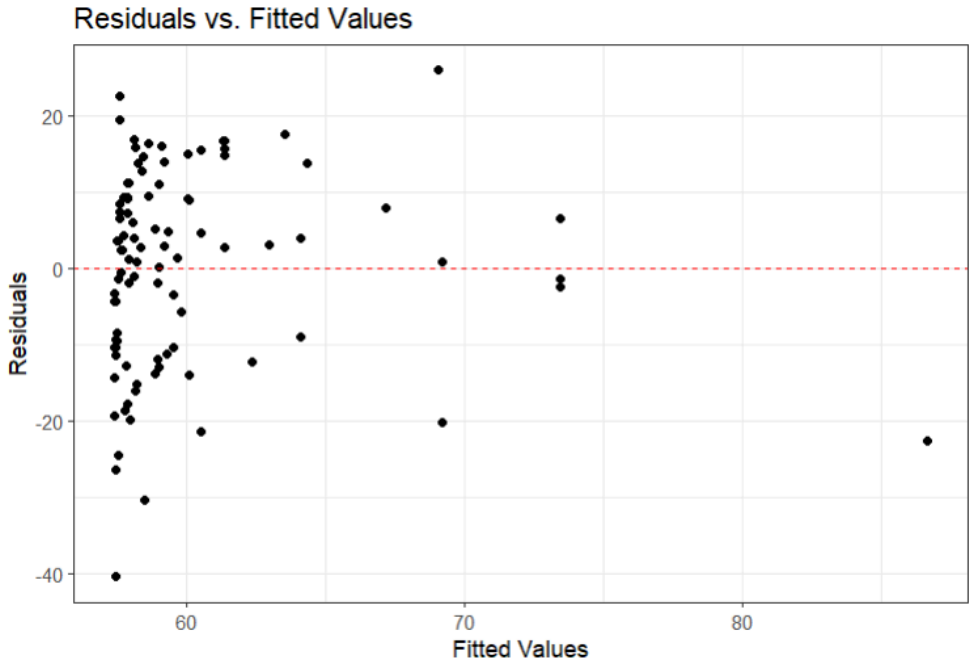
```
> # Perform the Shapiro-Wilk test
> shapiro.test(residuals)

        Shapiro-Wilk normality test

data:  residuals
W = 0.97333, p-value = 0.03999
```

On voit bien sur les figures et le test Shapiro Wilk, avec le valeur p = 0.03999 < 0.05, que les résiduels ne suivent pas la loi normale. Il suggère que l'hypothèse de normalité ne soit pas remplie pour ce modèle.

3. Indépendance

L'hypothèse d'observation indépendante stipule que chaque observation de l'ensemble de données est indépendante. Comme chaque variable est indépendante les unes des autres, l'hypothèse d'indépendance n'est pas violée.

4. Constant variance



```
> bptest(model_1)

        studentized Breusch-Pagan test

data:  model_1
BP = 0.16241, df = 1, p-value = 0.6869
```

Le résultat du test de Breusch Pagan indiquant que la valeur p = 0,68 > 0,05, la variance là où il y a des valeurs ajustées est distribuée de manière similaire, validant que l'hypothèse est remplie.

5.2. Modèle 2 : Multiple linear regression avec toutes les fonctions

```
> model_2 <- lm(Popularity ~ ., data = data)
>
> # Print the summary of the model
> summary(model_2)

Call:
lm(formula = Popularity ~ ., data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-26.5916  -5.4394  -0.4419   6.4508  23.9771

Coefficients: (1 not defined because of singularities)
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               8.869e+01  1.761e+01   5.036 2.82e-06 ***
BPM                      -4.325e-03  5.089e-02  -0.085 0.932484
Energy                   -3.133e-01  1.249e-01  -2.508 0.014121 *
Danceability             -3.646e-02  1.246e-01  -0.293 0.770579
Loudness_db               1.701e+00  5.675e-01   2.998 0.003607 **
Liveness                 -1.486e-02  6.369e-02  -0.233 0.816051
Valence                   1.268e-01  7.260e-02   1.747 0.084429 .
Duration                 -1.559e-02  1.212e-02  -1.286 0.201991
Acousticness             -7.838e-02  6.548e-02  -1.197 0.234790
Speechiness               2.833e-01  2.469e-01   1.148 0.254536
Top.Genreadult.standards -2.198e+00  8.700e+00  -0.253 0.801184
Top.Genrealternative      2.357e+00  9.022e+00   0.261 0.794599
Top.Genredance            2.387e+00  9.309e+00   0.256 0.798245
Top.Genrehiphop.rap      -1.427e+01  1.035e+01  -1.379 0.171640
Top.Genreindie           -2.255e+01  1.021e+01  -2.208 0.030083 *
Top.Genrepop             -6.005e+00  8.953e+00  -0.671 0.504271
Top.Genrerock            -4.270e+00  8.344e+00  -0.512 0.610211
Top.Genresoul                    NA         NA      NA       NA
Years_since_release       2.476e-01  8.538e-02   2.900 0.004800 **
Total.Followers           5.037e-07  1.429e-07   3.525 0.000701 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.03 on 81 degrees of freedom
Multiple R-squared:  0.466,     Adjusted R-squared:  0.3474
F-statistic: 3.927 on 18 and 81 DF,  p-value: 1.075e-05
```

Le variable "Top.Genresoul" et ne sont pas définies en raison de singularités, ce qui signifie qu'elles peuvent être parfaitement corrélées avec une ou plusieurs autres variables du modèle. En utilisant la fonction alias(), le résultat montre que “Top.Genresoul” est un alias de coefficients.

“Top.Genresoul” est alias avec toutes les variables “Top.Genre”, ce qui signifie qu'il peut être exprimé comme une combinaison linéaire des autres variables Top.Genre. En fait, “Top.Genresoul” et les autres variables “Top.Genres” sont des variables dummies créées à partir de la même variable catégorielle, et l'une d'elles (dans ce cas, Top.Genresoul) peut être exprimée comme une combinaison linéaire des autres.

```
> alias_info <- alias(model_2)
>
> # Print the alias information
> print(alias_info)
Model :
Popularity ~ BPM + Energy + Danceability + Loudness_db + Liveness +
    Valence + Duration + Acousticness + Speechiness + Top.Genreadult.standards +
    Top.Genrealternative + Top.Genredance + Top.Genrehiphop.rap +
    Top.Genreindie + Top.Genrepop + Top.Genrerock + Top.Genresoul +
    Years_since_release + Total.Followers

Complete :
              (Intercept) BPM Energy Danceability Loudness_db Liveness Valence Duration Acousticness
Top.Genresoul  1           0   0      0            0           0        0       0        0
              Speechiness Top.Genreadult.standards Top.Genrealternative Top.Genredance
Top.Genresoul  0          -1                       -1                   -1
              Top.Genrehiphop.rap Top.Genreindie Top.Genrepop Top.Genrerock Years_since_release
Top.Genresoul -1                  -1             -1           -1            0
              Total.Followers
Top.Genresoul  0
```

Pour résoudre ce problème, on supprime "Top.Genresoul" du modèle puis le reconstruit

```
> model_2 <- lm(Popularity ~ . -Top.Genresoul, data = data)
>
> # Print the summary of the model
> summary(model_2)

Call:
lm(formula = Popularity ~ . - Top.Genresoul, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-26.5916  -5.4394  -0.4419   6.4508  23.9771

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              8.869e+01  1.761e+01   5.036 2.82e-06 ***
BPM                     -4.325e-03  5.089e-02  -0.085 0.932484
Energy                  -3.133e-01  1.249e-01  -2.508 0.014121 *
Danceability            -3.646e-02  1.246e-01  -0.293 0.770579
Loudness_db              1.701e+00  5.675e-01   2.998 0.003607 **
Liveness                -1.486e-02  6.369e-02  -0.233 0.816051
Valence                  1.268e-01  7.260e-02   1.747 0.084429 .
Duration                -1.559e-02  1.212e-02  -1.286 0.201991
Acousticness            -7.838e-02  6.548e-02  -1.197 0.234790
Speechiness              2.833e-01  2.469e-01   1.148 0.254536
Top.Genreadult.standards -2.198e+00  8.700e+00  -0.253 0.801184
Top.Genrealternative     2.357e+00  9.022e+00   0.261 0.794599
Top.Genredance           2.387e+00  9.309e+00   0.256 0.798245
Top.Genrehiphop.rap     -1.427e+01  1.035e+01  -1.379 0.171640
Top.Genreindie          -2.255e+01  1.021e+01  -2.208 0.030083 *
Top.Genrepop            -6.005e+00  8.953e+00  -0.671 0.504271
Top.Genrerock           -4.270e+00  8.344e+00  -0.512 0.610211
Years_since_release      2.476e-01  8.538e-02   2.900 0.004800 **
Total.Followers          5.037e-07  1.429e-07   3.525 0.000701 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.03 on 81 degrees of freedom
Multiple R-squared:  0.466,     Adjusted R-squared:  0.3474
F-statistic: 3.927 on 18 and 81 DF,  p-value: 1.075e-05
```

Dans le modèle, des variables telles que "Danceability", "BPM", "Liveness", "Duration", “Valance”, “Speechiness”, "Top.Genrealternative", "Top.Genrepop", "Top.Genredance", "Top.Genrerock", et “Top.Genreadult.standards” sont statistiquement significatives au niveau 0,05.

La valeur R^2 est de 0,466, ce qui signifie qu'environ 46.6 % de la variabilité de la “Popularité” peut être expliquée par le modèle.

```
> vif_values <- car::vif(model_2)
>
> # Print the VIF values
> print(vif_values)
                      BPM                    Energy              Danceability               Loudness_db
                 1.415696                  7.397313                  2.894531                  3.834144
                 Liveness                   Valence                  Duration              Acousticness
                 1.316686                  2.997451                  1.527984                  3.510501
              Speechiness Top.Genreadult.standards     Top.Genrealternative            Top.Genredance
                 1.663087                  7.036902                  5.480209                  4.017851
      Top.Genrehiphop.rap           Top.Genreindie             Top.Genrepop             Top.Genrerock
                 4.180408                  3.291808                  7.451953                 14.285624
      Years_since_release          Total.Followers
                 1.799509                  1.563635
```

Le facteur d'inflation de variance (VIF) indique le niveau de multicolinéarité dans le modèle:

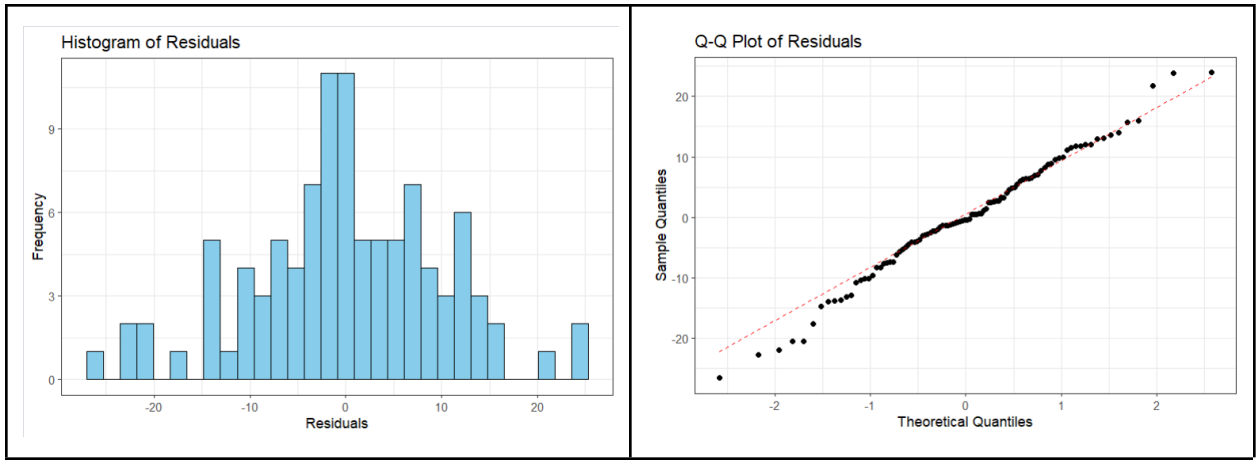
- "BPM", "Liveness", "Duration", "Speechiness", "Years_since_release", "Total.Followers" : ces variables ont des valeurs VIF proches de 1, ce qui suggère qu'elles ne sont pas corrélées avec les autres variables du modèle.
- "Danceability", "Loudness_db", "Valence", "Acousticness", "Top.Genredance", "Top.Genrehiphop.rap", "Top.Genreindie" : Ces variables ont des valeurs VIF comprises entre 1 et 5, ce qui suggère qu'elles ont une corrélation modérée avec d’autres variables du modèle.
- "Energy", "Top.Genreadult.standards", "Top.Genrealternative", "Top.Genrepop" : ces variables ont des valeurs VIF comprises entre 5 et 10, ce qui suggère qu'elles ont une forte corrélation avec d'autres variables du modèle.
- "Top.Genrerock" : Cette variable a une valeur VIF supérieure à 10, ce qui suggère qu'elle a une très forte corrélation avec d'autres variables du modèle. En effet, il s'agit d'une variable factice créée à partir de la même variable catégorielle.

On teste les hypothèses du modèle :

1. **Linéarité**

La plupart des variables sont non linéaires avec "Popularity", il signifie que l'hypothèse de Linéarité est violée.

2. **Normalité des résidus**



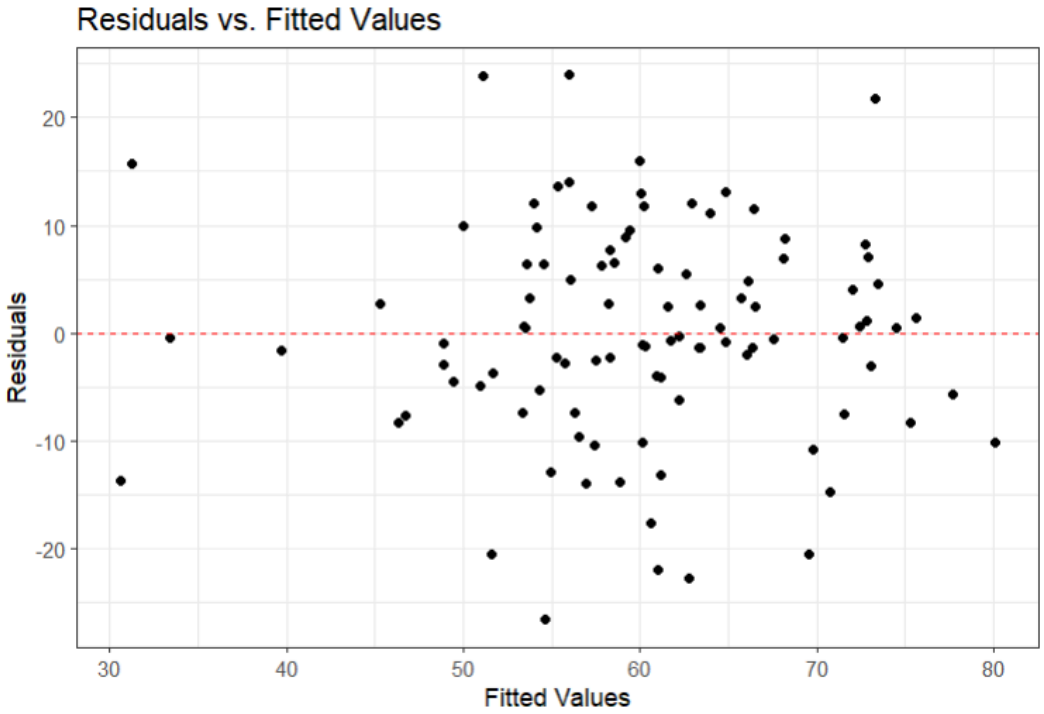

```
> shapiro.test(residuals_2)

        Shapiro-Wilk normality test

data:  residuals_2
W = 0.98927, p-value = 0.6057
```

On voit bien sur les figures et le test Shapiro Wilk, avec le valeur p = 0.6057 > 0.05, que les résiduels suivent la loi normale. Il suggère que l'hypothèse de normalité soit remplie pour ce modèle.

3. Indépendance

L'hypothèse d'observation indépendante stipule que chaque observation de l'ensemble de données est indépendante. Comme chaque variable est indépendante les unes des autres, l'hypothèse d'indépendance n'est pas violée.

4. Constant variance




```
> bptest(model_2)

        studentized Breusch-Pagan test

data:  model_2
BP = 28.612, df = 18, p-value = 0.05333
```

Le résultat du test de Breusch Pagan indiquant que la valeur p = 0,05333 > 0,05, la variance là où il y a des valeurs ajustées est distribuée de manière similaire, validant que l'hypothèse est remplie.

5.3. Modèle 3 : Multiple linear regression avec fonctions sélectionnées

```
# Create a copy of the data
data_transformed <- data

# Create interaction features
data_transformed$Energy_Danceability <- data$Energy * data$Danceability
data_transformed$BPM_Loudness <- data$BPM * data$Loudness_db
data_transformed$Acousticness_Speechiness <- data$Acousticness * data$Speechiness
data_transformed <- data_transformed %>% select(-Energy, -Danceability, -BPM, -Loudness_db, -Acousticness, -Speechiness, -Top.Genresoul)
```

Tout d'abord, on crée trois nouvelles fonctionnalités d'interaction dans l'ensemble de données « data_transformed ». Les fonctionnalités d'interaction sont de nouvelles variables créées en multipliant deux variables existantes ensemble. Ils peuvent être utiles dans un modèle de régression pour capturer l'effet de la combinaison de deux variables sur la variable de réponse.

- "Energy_Danceability" est créé en multipliant "Energy" et "Danceability".
- "BPM_Loudness" est créé en multipliant "BPM" et "Loudness_db".
- "Acousticness_Speechiness" est créé en multipliant "Acousticness" et "Speechiness".

Ensuite, on modifie l'ensemble de données "data_transformed" pour supprimer certaines colonnes. Les colonnes "Énergie", "Danceability", "BPM, Loudness_db", "Acousticness", "Speechiness" et "Top.Genresoul" sont supprimées.

```
> model_3 <- lm(formula, data = data_transformed)
> summary(model_3)

Call:
lm(formula = formula, data = data_transformed)

Residuals:
     Min       1Q   Median       3Q      Max
-26.0355  -6.8904   0.3995   7.2626  25.1663

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               5.864e+01  1.144e+01   5.127 1.86e-06 ***
Liveness                 -5.043e-02  6.165e-02  -0.818 0.415690
Valence                   9.089e-02  6.868e-02   1.323 0.189277
Duration                 -1.140e-02  1.183e-02  -0.964 0.337994
Top.Genreadult.standards  2.101e-02  8.789e+00   0.002 0.998098
Top.Genrealternative      3.394e+00  9.137e+00   0.372 0.711189
Top.Genredance            3.935e+00  9.332e+00   0.422 0.674359
Top.Genrehiphop.rap      -8.993e+00  9.688e+00  -0.928 0.355938
Top.Genreindie           -2.398e+01  1.017e+01  -2.358 0.020689 *
Top.Genrepop             -6.815e+00  8.925e+00  -0.764 0.447252
Top.Genrerock            -2.597e+00  8.441e+00  -0.308 0.759144
Years_since_release       2.070e-01  8.217e-02   2.520 0.013637 *
Total.Followers           5.331e-07  1.351e-07   3.945 0.000165 ***
Energy_Danceability      -4.722e-04  1.337e-03  -0.353 0.724885
BPM_Loudness              5.621e-03  3.119e-03   1.802 0.075100 .
Acousticness_Speechiness  1.081e-02  7.527e-03   1.436 0.154659
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 11.19 on 84 degrees of freedom
Multiple R-squared:  0.4299,    Adjusted R-squared:  0.3281
F-statistic: 4.223 on 15 and 84 DF,  p-value: 9.494e-06
```

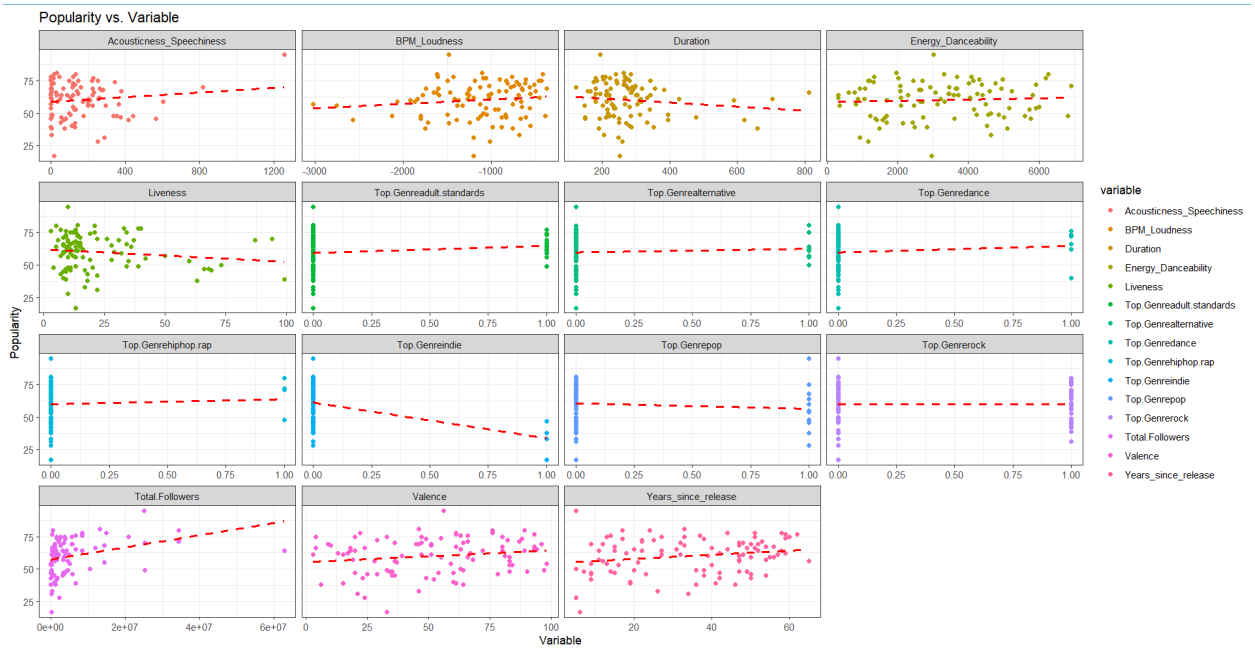
Dans le modèle, les 3 nouvelles fonctionnalités d'interaction "Energy_Danceability", "BPM_Loudness", "Acousticness_Speechiness" sont statistiquement significatives au niveau 0,05.

La valeur R^2 est de 0,4299, ce qui signifie qu'environ 43 % de la variabilité de la “Popularité” peut être expliquée par le modèle.

```
> vif_values <- car::vif(model_3)
>
> # Print the VIF values
> print(vif_values)
                Liveness                  Valence                 Duration Top.Genreadult.standards
                1.198408                 2.605316                 1.413633                 6.976620
     Top.Genrealternative             Top.Genredance      Top.Genrehiphop.rap           Top.Genreindie
                5.459471                 3.921957                 3.559866                 3.170990
           Top.Genrepop             Top.Genrerock        Years_since_release          Total.Followers
                7.193583                14.201970                 1.618802                 1.357558
      Energy_Danceability           BPM_Loudness Acousticness_Speechiness
                3.917992                 2.072949                 1.540151
```

Les 3 nouvelles fonctionnalités d'interaction ont des valeurs VIF comprises entre 1 et 5, ce qui suggère qu'elles ont une corrélation modérée avec d’autres variables du modèle.
On teste les hypothèses du modèle :

1. **Linéarité**

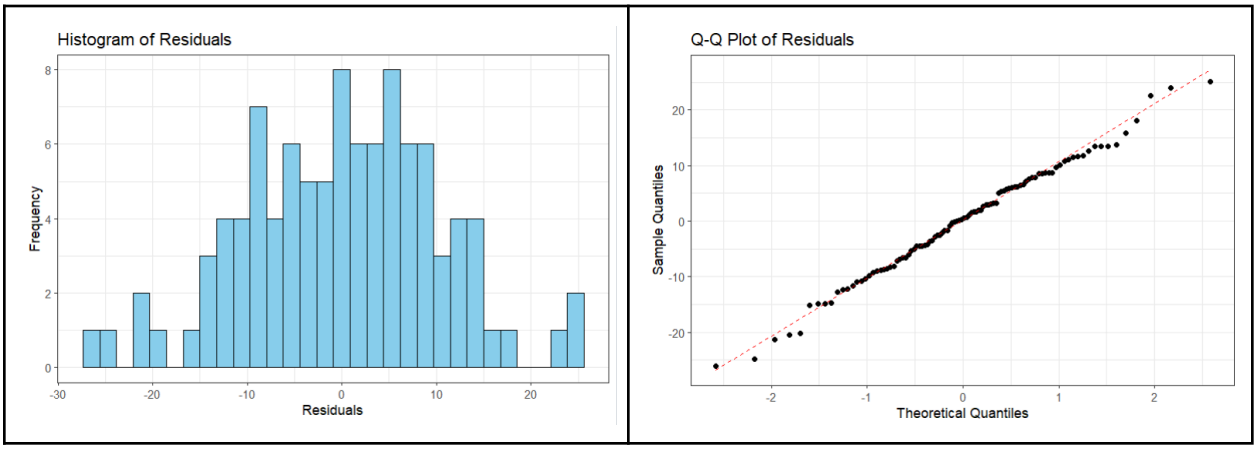


La plupart des variables sont non linéaires avec "Popularity", il signifie que l'hypothèse de Linéarité est violée.

2. **Indépendance**

L'hypothèse d'observation indépendante stipule que chaque observation de l'ensemble de données est indépendante. Comme chaque variable est indépendante les unes des autres, l’hypothèse d’indépendance n’est pas violée.

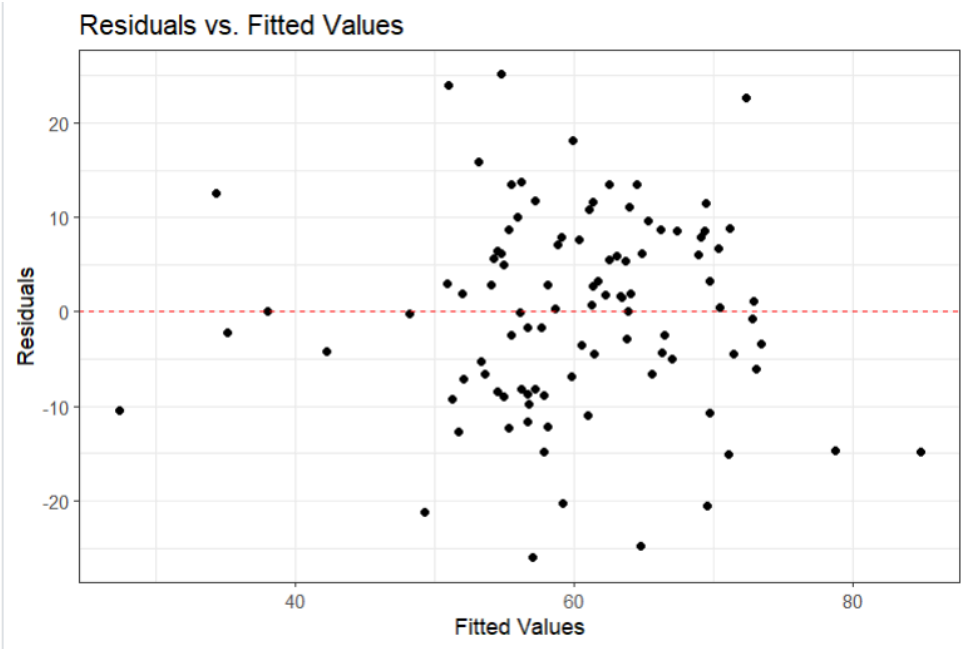
3. Normalité des résidus




```
> shapiro.test(residuals_3)

        Shapiro-Wilk normality test

data:  residuals_3
W = 0.99247, p-value = 0.8544
```

On voit bien sur les figures et le test Shapiro Wilk, avec le valeur p = 0.8544 > 0.05, que les résiduels suivent la loi normale. Il suggère que l'hypothèse de normalité soit remplie pour ce modèle.

4. Constant Variance



```
> bptest(model_3)

        studentized Breusch-Pagan test

data:  model_3
BP = 26.97, df = 15, p-value = 0.02898
```

Le résultat du test de Breusch Pagan indiquant que la valeur p = 0,02898 < 0,05, il suggère que l'hypothèse de Constant Variance ne soit pas remplie pour ce modèle.

6. Évaluation des modèles

L'évaluation des modèles de régression est cruciale pour évaluer leurs performances et déterminer dans quelle mesure ils capturent la relation entre les variables indépendantes et la variable dépendante. Il utilise quelques mesures clés pour évaluer le modèle :

1. **R au carré (coefficient de détermination) :**

- R au carré représente la proportion de variance de la variable dépendante expliquée par les variables indépendantes du modèle.
- Il va de 0 à 1, des valeurs plus élevées indiquant un meilleur ajustement (plus de variance expliquée par le modèle).

2. **Erreur quadratique moyenne (RMSE) :**

- RMSE est la racine carrée de MSE.
- Il est plus facile à interpréter dans les mêmes unités que la variable dépendante, ce qui en fait une mesure plus intuitive de l'erreur de prédiction moyenne.

3. **Erreur absolue moyenne (MAE) :**

- MAE calcule la différence absolue moyenne entre les valeurs prédites et les valeurs réelles.
- Il est moins sensible aux valeurs aberrantes que MSE/RMSE, car il utilise des différences absolues au lieu de carrés.
- MAE représente l'ampleur moyenne des erreurs dans les mêmes unités que la variable dépendante.

```
> predictions <- predict(model_1, newdata = data)
>
> # Calculate the residuals
> residuals <- data$Popularity - predictions
>
> # Calculate RMSE
> rmse <- sqrt(mean(residuals^2))
> print(paste("RMSE: ", rmse))
[1] "RMSE:  12.8160532387548"
>
> # Calculate MAE
> mae <- mean(abs(residuals))
> print(paste("MAE: ", mae))
[1] "MAE:  10.3884002862376"
```

```
> predictions <- predict(model_2, newdata = data)
>
> # Calculate the residuals
> residuals <- data$Popularity - predictions
>
> # Calculate RMSE
> rmse <- sqrt(mean(residuals^2))
> print(paste("RMSE: ", rmse))
[1] "RMSE:  9.92628151172089"
>
> # Calculate MAE
> mae <- mean(abs(residuals))
> print(paste("MAE: ", mae))
[1] "MAE:  7.64614365460246"
```

```
> predictions <- predict(model_3, newdata = data_
>
> # Calculate the residuals
> residuals <- data$Popularity - predictions
>
> # Calculate RMSE
> rmse <- sqrt(mean(residuals^2))
> print(paste("RMSE: ", rmse))
[1] "RMSE:  10.2564570711803"
>
> # Calculate MAE
> mae <- mean(abs(residuals))
> print(paste("MAE: ", mae))
[1] "MAE:  8.22444744691153"
```

- Modèle 1 : RMSE est d'environ 12,82 et MAE est d'environ 10,39. Il signifie qu'en moyenne, les prédictions du modèle sont éloignées d'environ 12,82 unités des valeurs réelles (en termes de RMSE) et d'environ 10,39 unités en termes de MAE.
- Modèle 2 : RMSE est d'environ 9,93 et MAE est d'environ 7,65. Ce modèle fonctionne mieux que le modèle 1 car il a des valeurs RMSE et MAE inférieures.
- Modèle 3 : RMSE est d'environ 10,26 et MAE est d'environ 8,22. Ce modèle est légèrement moins performant que le modèle 2 mais meilleur que le modèle 1.

	RMSE	MAE	R^2
Modèle 1	12.82	10.39	0.11
Modèle 2	9.93	7.65	0.47
Modèle 3	10.26	8.22	0.43

Le modèle 2 offre les meilleures performances parmi les trois modèles car il présente les valeurs RMSE et MAE les plus basses. Cela signifie que les prédictions du modèle 2 sont, en moyenne, plus proches des valeurs réelles que celles des deux autres modèles.

7. Conclusion

- L'analyse exploratoire des données est utile pour sélectionner des variables numériques et catégorielles pour les régressions linéaires modèles.
- L'ajustement de plusieurs modèles de régression linéaire peut nécessiter des essais et des erreurs pour sélectionner les variables qui correspondent à un modèle précis tout en conservant les hypothèses du modèle.
- Le modèle utilisait une régression linéaire multiple expliquant seulement 47 % (43 % avec la transformation de données) de la variation de la popularité des chansons. Il est nécessaire d'explorer d'autres types de modèles mieux adaptés à cet jeu de données.