Informe escrito de Estadística
"Segunda Fase"

Carlos Aryam Martínez Molina c.molina@estudiantes.matcom.uh.cu
Grupo C411

Eziel Ramos Piñón e.ramos@estudiantes.matcom.uh.cu
Grupo C411

Ariel Plasencia Díaz a.plasencia@estudiantes.matcom.uh.cu
Grupo C411

Índice

1 Introducción

El objetivo de este trabajo es, a partir de un conjunto de datos que coincide con las 50 mejores canciones de la aplicación *Spotify*, realizar un estudio estadístico aplicando las técnicas de regresión lineal simple y múltiple, análisis de varianza y de reducción de dimensiones para llegar a conclusiones precisas sobre características y propiedades de las canciones antes mencionadas.

En nuestro conjunto de datos predominan los datos cuantitativos, por lo que nos dimos la libertad de insertar algunas variables cualitativas con el fin de tener mayores argumentos a la hora de comparar nuestras canciones. Entre las observaciones agregadas y usadas principalmente a partir del capítulo 5 tenemos la *Colaboración* que representa la existencia o no de múltiples cantantes, el *Año de Nacimiento*, el *Género Musical* constituido por los valores pop, reggaeton y otros y el *Tamaño de la canción* que denota si el tema es corto o largo en dependencia de si la duración es menor al promedio de los segundos de todas las canciones.

Para llevar a cabo lo explicado anteriormente nos apoyaremos en el lenguaje de programación *R*, el cual fue diseñado por Ross Ihaka y Robert Gentleman en 1933 y es muy utilizado en nuestros días para la investigación por la comunidad estadística, en el campo de la minería de datos, la investigación biomédica, la bioinformática y las matemáticas financieras.

Para todo este análisis tomaremos un nivel de significación de 0.05 y es necesario mencionar que nos apoyamos en librerías que nos provee el lenguaje *R*, entre las que se encuentran *lmtest*, *ggplot2*, *car*, *factoextra*, *cluster*, *purrr*, *gridExtra*, *lsr* y *psych* utilizadas fundamentalmente para el análisis de los supuestos, para plotear los gráficos y la confección de clústeres.

2 Ejercicio

2.1 Problema

Realice un estudio de sus datos usando las técnicas de regresión lineal (simple y múltiple), de reducción de dimensiones y de análisis de varianza (ANOVA).

a) Escoja las variables a las cuáles les aplicará cada técnica y explique por qué.

b) Realice el análisis de los supuestos y explique si es válida la aplicación de la técnica en esa variable.

3 Regresión Lineal Simple

Para el estudio de la técnica de regresión lineal simple escogimos las observaciones *golpes por minutos* (*beats per minutes*) como variable dependiente y *cantidad de texto hablado* (*speechiness*) como variable independiente. Consideramos que estas observaciones son fundamentales para el análisis de las canciones porque nos ayudan a interpretar su contraste y monotonía.

3.1 Representación gráfica de las observaciones

El primer paso antes de generar un modelo de regresión es representar los datos para poder intuir si existe una relación y cuantificar dicha relación mediante un coeficiente de correlación. Si en este paso no se detecta la posible relación lineal, no tiene sentido seguir adelante generando un modelo lineal (se tendrían que probar otros modelos).

```
ggplot(data = datos, mapping = aes(x = datos$Speechiness,
       y = datos$Beats_Per_Minute)) +
   geom_point(color = "firebrick", size = 2) +
   labs(title = "Diagrama de dispersión", x = "cantidad de texto hablado",
        y = "golpes por minutos") +
   theme_bw() +
   theme(plot.title = element_text(hjust = 0.5))
cor.test(x = datos$Speechiness, y = datos$Beats_Per_Minute, method = "pearson")
```

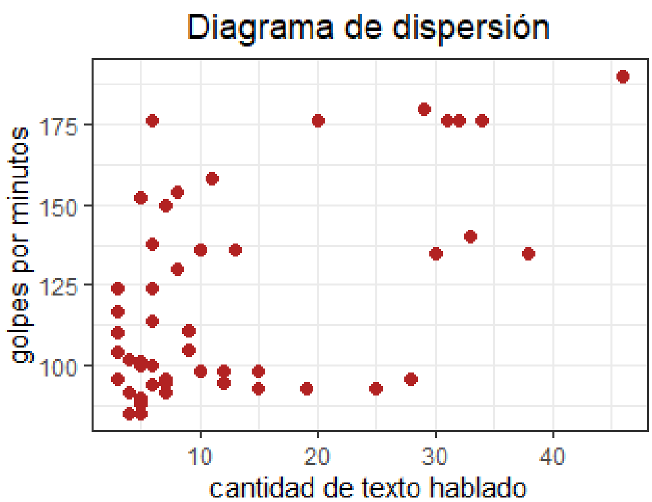


Figura 1: Diagrama de dispersión

```
        Pearson's product-moment correlation

data:  datos$Speechiness and datos$Beats_Per_Minute
t = 4.6472, df = 48, p-value = 2.65e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3298485 0.7232558
sample estimates:
      cor
0.5570519
```

Figura 2: Prueba de correlación mediante el método de Pearson

El gráfico de dispersión (figura 1) y el test de correlación (figura 2) muestran una relación lineal de intensidad normal, aunque no tan significativa, porque $r = 0.5570519 \approx 0.56$ y el $p - valor = 2.65 \cdot 10^{-5} = 0.0000265$. En conferencias estudiadas vimos que si el coeficiente de correlación, r, se encuentra entre 0.4 y 0.7 no podemos afirmar que exista correlación lineal, pero tampoco podemos decir que no haya, por lo que intentaremos generar un modelo de regresión lineal simple que permita predecir el número de golpes por minuto en función de la cantidad de texto hablado de una canción.

3.2 Cálculo del modelo de regresión lineal simple

A continuación, calcularemos el modelo de regresión lineal simple.

```
regresion_lineal <- lm(datos$Beats_Per_Minute ~ datos$Speechiness, data = datos)
summary(regresion_lineal)
```

```
Call:
lm(formula = datos$Beats_Per_Minute ~ datos$Speechiness, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max 
-47.993 -18.453  -8.025  18.482  65.933 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       100.8149     5.5311  18.227  < 2e-16 ***
datos$Speechiness   1.5421     0.3318   4.647 2.65e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 25.93 on 48 degrees of freedom
Multiple R-squared:  0.3103,    Adjusted R-squared:  0.2959 
F-statistic:  21.6 on 1 and 48 DF,  p-value: 2.65e-05
```

Figura 3: Resumen del modelo de regresión lineal simple

La primera columna (*Estimate*) devuelve el valor estimado para los dos parámetros de la ecuación del modelo lineal (β_0 *y* β_1) que equivalen a la ordenada en el origen y la pendiente respectivamente. Se muestran los *errores estándares*, el *valor del estadístico t* y el *p-valor* de cada uno de los dos parámetros. Esto permite determinar si los parámetros son significativamente distintos de 0, es decir, que tienen importancia en el modelo. En los modelos de regresión lineal simple, el parámetro más informativo suele ser la pendiente. Para el modelo generado, tanto la ordenada en el origen como la pendiente son significativas ya que $p - valor = 0.0000265 < 0.05$. El valor de *R-squared* indica que el modelo calculado explica el 31.03 % de la variabilidad presente en la variable respuesta (*golpes por minuto*) mediante la variable independiente (*cantidad de texto hablado*). El $p - valor$ obtenido en el *test F* determina que sí es significativamente superior la varianza explicada por el modelo en comparación a la varianza total. Este parámetro determina que el modelo es significativo y por lo tanto se puede aceptar. El modelo lineal generado sigue como de mejor ajuste a la ecuación (1) donde y es el número de golpes por minuto de la canción o variable dependiente y x equivale a la cantidad de texto hablado de una canción o variable independiente.

$$y = 100.81 + 1.54x \tag{1}$$

3.3 Representación gráfica del modelo

A continuación mostraremos el gráfico del modelo, en el cual señalaremos tanto la recta de mejor ajuste como los datos de las observaciones escogidas. Además de la línea de mínimos cuadrados, es recomendable incluir los límites superior e inferior del intervalo de confianza. Esto permite identificar la región en la que, según el modelo generado y para un determinado nivel de confianza, se encuentra el valor promedio de la variable dependiente.

```
ggplot(data = datos, mapping = aes(x = datos$Speechiness,
       y = datos$Beats_Per_Minute)) +
   geom_point(color = "firebrick", size = 2) +
   labs(title = "Diagrama de dispersión", x = "cantidad de texto hablado",
       y = "golpes por minutos") +
   geom_smooth(method = "lm", se = FALSE, color = "black", formula = "y ~ x") +
   theme_bw() +
   theme(plot.title = element_text(hjust = 0.5))
ggplot(data = datos, mapping = aes(x = datos$Speechiness,
       y = datos$Beats_Per_Minute)) +
   geom_point(color = "firebrick", size = 2) +
   labs(title = "Diagrama de dispersión", x = "cantidad de texto hablado",
       y = "golpes por minuto") +
   geom_smooth(method = "lm", se = TRUE, color = "black", formula = "y ~ x") +
   theme_bw() +
   theme(plot.title = element_text(hjust = 0.5))
```

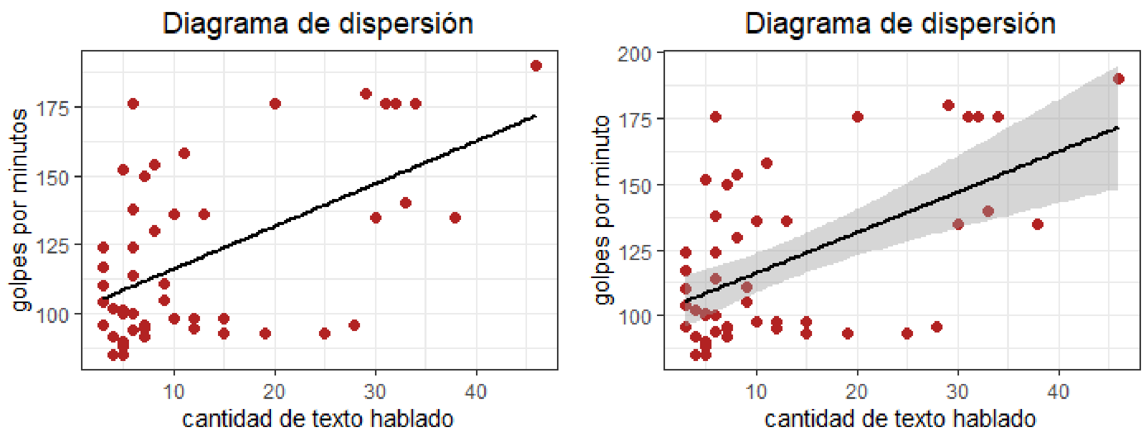


Figura 4: Modelos de regresión lineal simple

3.4 Verificar condiciones para poder aceptar un modelo lineal

3.4.1 Relación lineal entre la variable dependiente e independiente

Se calculan los residuos para cada observación y se representan. Si las observaciones siguen la línea del modelo, los residuos se deben distribuir aleatoriamente entorno al valor 0.

```
datos$Prediccion <- regresion_lineal$fitted.values
datos$Residuos <- regresion_lineal$residuals
ggplot(data = datos, aes(x = datos$Prediccion, y = datos$Residuos)) +
  geom_point(aes(color = datos$Residuos)) +
  scale_color_gradient2(low = "blue3", mid = "grey", high = "red") +
  geom_hline(yintercept = 0) +
  geom_segment(aes(xend = datos$Prediccion, yend = 0), alpha = 0.2) +
  labs(title = "Distribución␣de␣los␣residuos", x = "predicción␣del␣modelo",
       y = "residuos") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5), legend.position = "none")
```

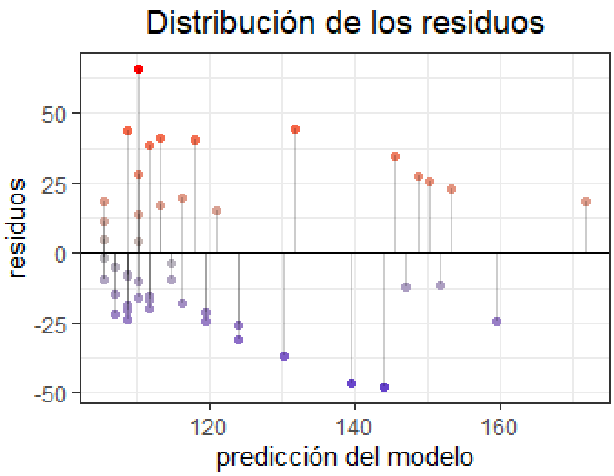


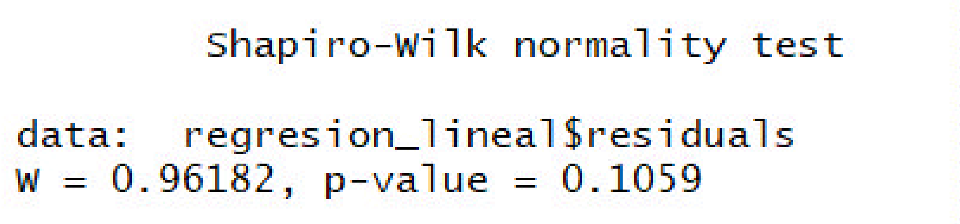
Figura 5: Distribución de los residuos

Los residuos se distribuyen de forma aleatoria entorno al 0, por lo que se acepta la linealidad.

3.4.2 Distribución normal de los residuos

Los residuos se deben distribuir de forma normal con media 0. Para comprobarlo se recurre a histogramas, a los cuantiles normales y a un test de contraste de normalidad.

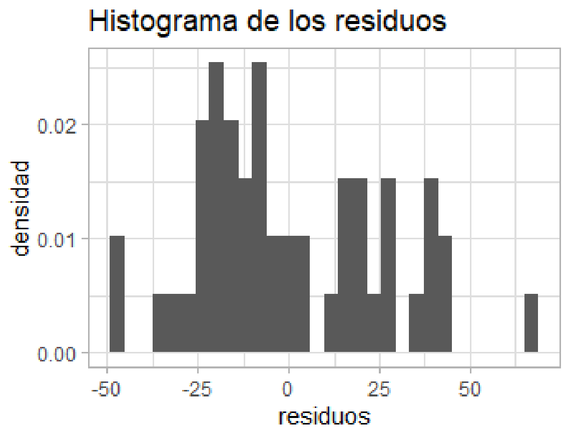
```
shapiro.test(regresion_lineal$residuals)
ggplot(data = datos, aes(x = datos$Residuos)) +
  geom_histogram(aes(y = ..density..)) +
  labs(title = "Histograma␣de␣los␣residuos", x = "residuos", y = "densidad") +
  theme_light()
qqnorm(regresion_lineal$residuals)
qqline(regresion_lineal$residuals)
```



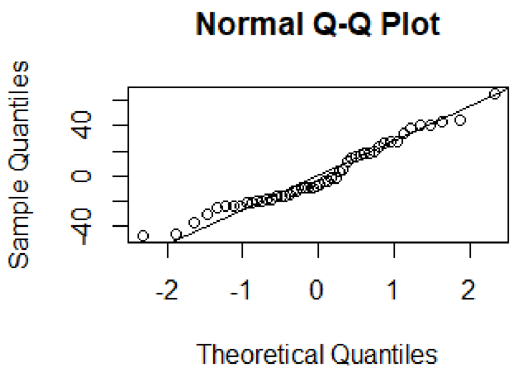
```
Shapiro-Wilk normality test

data:  regresion_lineal$residuals
W = 0.96182, p-value = 0.1059
```

Figura 6: Prueba de normalidad de los residuos



(a) Histograma de los residuos



(b) Gráfico de normalidad

Figura 7: Gráficos de los residuos

Tanto la representación gráfica como la prueba de normalidad confirman la distribución normal de los residuos ya que $p - valor = 0.1059 \approx 0.106 > 0.05$.

3.4.3 Varianza constante de los residuos (homocedasticidad)

La variabilidad de los residuos debe de ser constante a lo largo del eje de las abcisas, sin embargo un patrón cónico es indicativo de falta de homogeneidad en la varianza.

```
ggplot(data = datos, aes(x = datos$Prediccion, y = datos$Residuos)) +
   geom_point(aes(color = datos$Residuos)) +
   scale_color_gradient2(low = "blue3", mid = "grey", high = "red") +
   geom_segment(aes(xend = datos$Prediccion, yend = 0), alpha = 0.2) +
   geom_smooth(se = FALSE, color = "firebrick", method = "loess",
                 formula = "y ~ x") +
   labs(title = "Distribución de los residuos", x = "predicción del modelo",
         y = "residuos") +
   geom_hline(yintercept = 0) +
   theme_bw() +
   theme(plot.title = element_text(hjust = 0.5), legend.position = "none")
bptest(regresion_lineal)
```

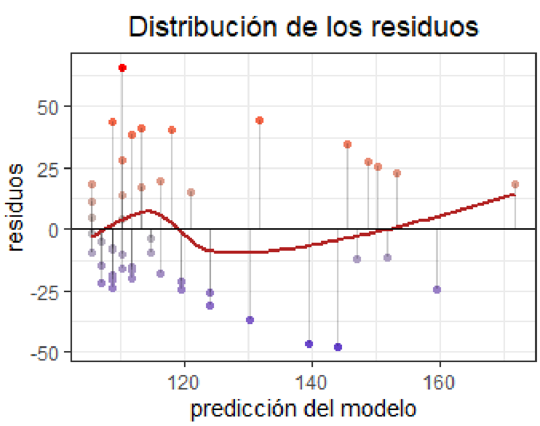


Figura 8: Distribución de los residuos

```
	Durbin-Watson test

data:  regresion_lineal
DW = 1.982, p-value = 0.4721
alternative hypothesis: true autocorrelation is greater than 0
```

Figura 9: Prueba de homocedasticidad de los residuos

Ni la representación gráfica ni el contraste de hipótesis muestran evidencias que haga sospechar falta de homocedasticidad.

3.5 Código

Solución en R

4 Regresión Lineal Múltiple

Para el estudio de la técnica de regresión lineal múltiple haremos un bosqueo por todas las variables cuantitativas propuestas, pero profundizaremos con las que mantienen una estrecha relación con la energía brindada por las canciones, índice que representa la intensidad y las ganas tanto de escucharla como de bailarla.

4.1 Analizar la relación entre variables

El primer paso a la hora de establecer un modelo lineal múltiple es estudiar la relación que existe entre las observaciones. Esta información es crítica a la hora de identificar cuáles pueden ser los mejores predictores para el modelo, qué variables presentan relaciones de tipo no lineal (por lo que no pueden ser incluidas) y para identificar colinialidad entre predictores. A modo complementario, es recomendable representar la distribución de cada variable mediante histogramas. Las dos formas principales de hacerlo son mediante representaciones gráficas (gráficos de dispersión) y el cálculo del coeficiente de correlación de cada par de variables.

```r
round(cor(x = datos, method = "pearson"), 3)
multi.hist(x = datos, dcol = c("blue", "red"), dlty = c("dotted", "solid"), main = "")
```

	Beats_Per_Minute	Energy	Danceability	Loudness.dB.	Liveness	Valence	Length	Acousticness
Beats_Per_Minute	1.000	0.044	-0.094	0.017	-0.167	-0.012	-0.139	-0.031
Energy	0.044	1.000	0.018	0.671	0.163	0.439	0.225	-0.340
Danceability	-0.094	0.018	1.000	0.016	-0.150	0.173	0.000	-0.098
Loudness.dB.	0.017	0.671	0.016	1.000	0.259	0.238	0.219	-0.138
Liveness	-0.167	0.163	-0.150	0.259	1.000	0.016	0.132	0.021
Valence	-0.012	0.439	0.173	0.238	0.016	1.000	-0.018	-0.052
Length	-0.139	0.225	0.000	0.219	0.132	-0.018	1.000	-0.076
Acousticness	-0.031	-0.340	-0.098	-0.138	0.021	-0.052	-0.076	1.000
Speechiness	0.557	-0.090	-0.103	-0.272	-0.125	-0.053	0.047	0.008
Popularity	0.196	-0.080	-0.071	-0.043	0.093	-0.318	-0.088	-0.035
YearFromBirth	0.113	-0.073	0.007	-0.209	-0.081	0.112	-0.252	0.218

	Speechiness	Popularity	YearFromBirth
Beats_Per_Minute	0.557	0.196	0.113
Energy	-0.090	-0.080	-0.073
Danceability	-0.103	-0.071	0.007
Loudness.dB.	-0.272	-0.043	-0.209
Liveness	-0.125	0.093	-0.081
Valence	-0.053	-0.318	0.112
Length	0.047	-0.088	-0.252
Acousticness	0.008	-0.035	0.218
Speechiness	1.000	0.239	0.089
Popularity	0.239	1.000	0.072
YearFromBirth	0.089	0.072	1.000

Figura 10: Coeficiente de correlación para cada par de variables

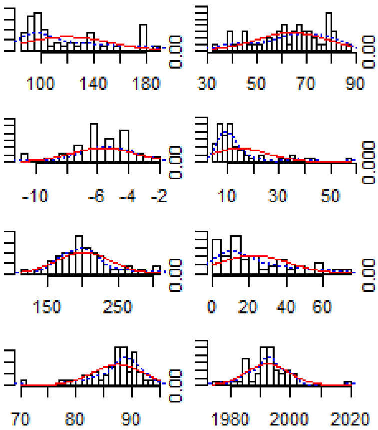


Figura 11: Distribución de cada variable mediante histogramas

Del análisis preliminar se pueden extraer las siguientes conclusiones:

1. Las observaciones que tienen una mayor relación lineal con la energía de la canción son: la sonoridad ($r = 0.671$) y la positividad ($r = 0.439$).
2. El índice que representa a los golpes por minutos y la acústica de una canción están medianamente correlacionados ($r = 0.557$) por lo que posiblemente no sea útil introducir ambos predictores en el modelo.

4.2 Generar el modelo

Como hemos estudiado, hay diferentes formas de llegar al modelo final más adecuado. En este caso se va a emplear el método mixto iniciando el modelo con todas las observaciones como predictores y realizando la selección de los mejores predictores.

```
modelo <- lm(datos$Energy ~ datos$Beats_Per_Minute + datos$Danceability +
             datos$Loudness.dB. + datos$Length + datos$Valence +
             datos$Liveness + datos$Acousticness + datos$Speechiness +
             datos$Popularity, data = datos)
summary(modelo)
```

```
Call:
lm(formula = datos$Energy ~ datos$Beats_Per_Minute + datos$Danceability +
    datos$Loudness.dB. + datos$Length + datos$Valence + datos$Liveness +
    datos$Acousticness + datos$Speechiness + datos$Popularity,
    data = datos)

Residuals:
     Min       1Q   Median       3Q      Max
-17.1912  -5.8420   0.0916   5.6392  23.9678

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            71.100649  36.360736   1.955  0.05754 .
datos$Beats_Per_Minute -0.003527   0.059180  -0.060  0.95277
datos$Danceability     -0.072464   0.122827  -0.590  0.55853
datos$Loudness.dB.      3.906041   0.819225   4.768 2.47e-05 ***
datos$Length            0.030970   0.039023   0.794  0.43208
datos$Valence           0.200639   0.070231   2.857  0.00676 **
datos$Liveness          0.002746   0.137040   0.020  0.98411
datos$Acousticness     -0.183487   0.075424  -2.433  0.01955 *
datos$Speechiness       0.090985   0.170290   0.534  0.59610
datos$Popularity        0.072858   0.349244   0.209  0.83581
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.839 on 40 degrees of freedom
Multiple R-squared:  0.6098,	Adjusted R-squared:  0.522
F-statistic: 6.946 on 9 and 40 DF,  p-value: 5.882e-06
```

Figura 12: Resumen del modelo de regresión múltiple

El modelo con todas las variables introducidas como predictores posee un *R-squared* alto (0.6098), que es capaz de explicar el 60, 98 % de la variabilidad observada en la energía de una canción. El $p - valor$ del modelo es significativo ($5.882 \cdot 10^{-6}$) por lo que se puede aceptar que el modelo no es por azar, al menos uno de los coeficientes parciales de regresión es distinto de 0. Muchos de ellos no son significativos, lo que es un indicativo de que podrían no contribuir al modelo.

4.3 Selección de los mejores predictores

En este caso se van a emplear la estrategia de stepwise mixto.

```
step(object = modelo, direction = "both", trace = 1)
```

El resultado de la línea anterior no lo mostraremos por su gran magnitud en cuanto a espacio, pero se puede ver en el código. Como consecuencia de su interpretación obtenemos que el mejor modelo del proceso de selección ha sido:

```
modelo <- lm(datos$Energy ~ datos$Loudness.dB. + datos$Valence + datos$Acousticness,
             data = datos)
summary(modelo)
```

```
Call:
lm(formula = datos$Energy ~ datos$Loudness.dB. + datos$Valence +
    datos$Acousticness, data = datos)

Residuals:
    Min      1Q  Median      3Q     Max
-17.267  -6.059   1.040   6.068  23.658

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        80.25346    5.91280  13.573  < 2e-16 ***
datos$Loudness.dB.  3.92805    0.67660   5.806 5.66e-07 ***
datos$Valence       0.18546    0.06178   3.002  0.00432 **
datos$Acousticness -0.18443    0.07125  -2.589  0.01286 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.381 on 46 degrees of freedom
Multiple R-squared:  0.5921,    Adjusted R-squared:  0.5655
F-statistic: 22.26 on 3 and 46 DF,  p-value: 4.727e-09
```

Figura 13: Modelo de regresión múltiple

Es recomendable mostrar el intervalo de confianza para cada uno de los coeficientes parciales de regresión.

```
confint(lm(formula = datos$Energy ~ datos$Loudness.dB. + datos$Valence +
           datos$Acousticness, data = datos))
```

```
                         2.5 %      97.5 %
(Intercept)        68.35161944 92.15530816
datos$Loudness.dB.  2.56612060  5.28998317
datos$Valence       0.06110736  0.30982058
datos$Acousticness -0.32784806 -0.04101683
```

Figura 14: Intervalos de confianza

4.4 Validación de condiciones para la regresión múltiple lineal

4.4.1 Relación lineal entre los predictores numéricos y la variable respuesta

Esta condición se puede validar bien mediante diagramas de dispersión entre la variable dependiente y cada uno de los predictores o con diagramas de dispersión entre cada uno de los predictores y los residuos del modelo. Si la relación es lineal, los residuos deben de distribuirse aleatoriamente en torno a 0 con una variabilidad constante a lo largo del eje de las abcisas. Esta última opción suele ser más indicada ya que permite identificar posibles datos atípicos.

```
plot1 <- ggplot(data = datos, aes(datos$Loudness.dB., modelo$residuals)) +
   geom_point() +
   labs(title = "", x = "sonoridad", y = "residuos") +
   geom_smooth(formula = "y ~ x", method = "loess", color = "firebrick") +
   geom_hline(yintercept = 0) +
   theme_bw()
plot2 <- ggplot(data = datos, aes(datos$Valence, modelo$residuals)) +
   geom_point() +
   labs(title = "", x = "positividad", y = "residuos") +
   geom_smooth(formula = "y ~ x", method = "loess", color = "firebrick") +
   geom_hline(yintercept = 0) +
   theme_bw()
plot3 <- ggplot(data = datos, aes(datos$Acousticness, modelo$residuals)) +
   geom_point() +
   labs(title = "", x = "acústica", y = "residuos") +
   geom_smooth(formula = "y ~ x", method = "loess", color = "firebrick") +
   geom_hline(yintercept = 0) +
   theme_bw()
grid.arrange(plot1, plot2, plot3)
```

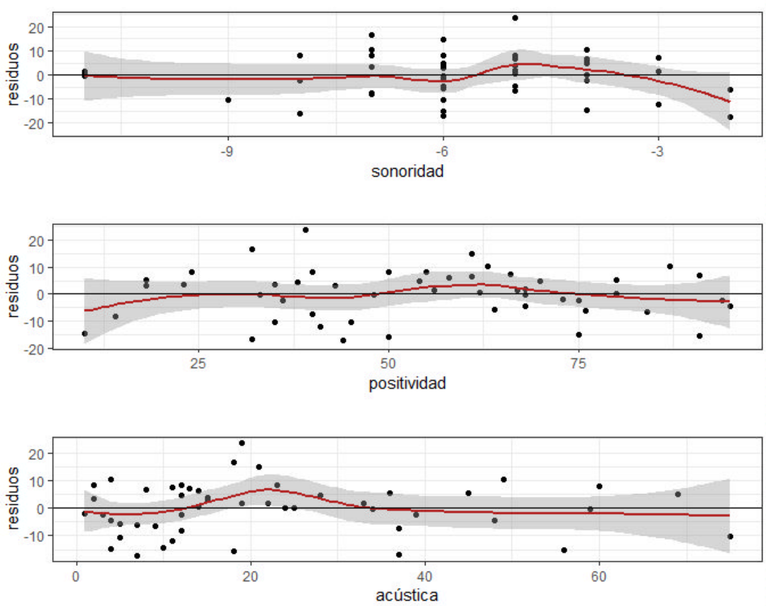


Figura 15: Diagramas de dispersión

Se cumple la linealidad para todos los predictores.

4.4.2 Distribución normal de los residuos

Los residuos se deben distribuir de forma normal con media 0. Para comprobarlo se recurre a los cuantiles normales y a un test de contraste de normalidad.

```
qqnorm(modelo$residuals)
qqline(modelo$residuals)
shapiro.test(modelo$residuals)
```

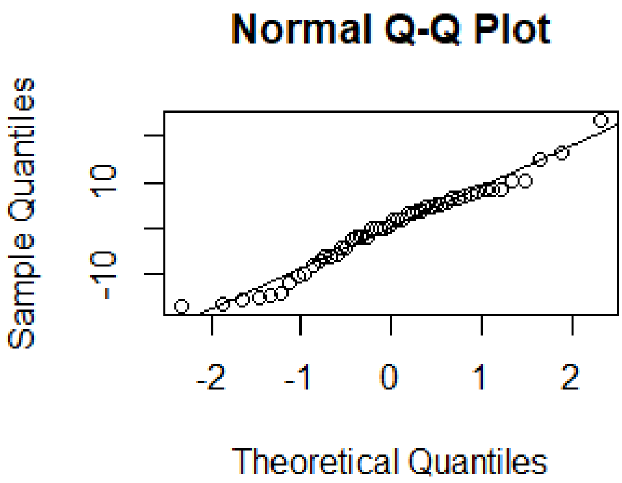
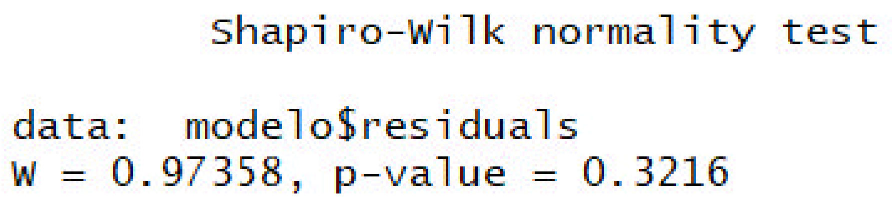


Figura 16: Gráfico de normalidad



```
Shapiro-Wilk normality test

data:  modelo$residuals
W = 0.97358, p-value = 0.3216
```

Figura 17: Resultado de la prueba Shakiro-Wilk

Tanto el análisis gráfico como el test de hipótesis confirman la normalidad.

4.4.3 Variabilidad constante de los residuos (homocedasticidad)

Al representar los residuos frente a los valores ajustados por el modelo, los primeros se tienen que distribuir de forma aleatoria en torno a 0, manteniendo aproximadamente la misma variabilidad a lo largo del eje de las abcisas. Si se observa algún patrón específico, por ejemplo forma cónica o mayor dispersión en los extremos, significa que la variabilidad es dependiente del valor ajustado y por lo tanto no hay homocedasticidad.

```
ggplot(data = datos, aes(modelo$fitted.values, modelo$residuals)) +
   geom_point() +
   geom_smooth(formula = "y ~ x", method = "loess", color = "firebrick", se = FALSE) +
   geom_hline(yintercept = 0) +
   theme_bw()
bptest(modelo)
```

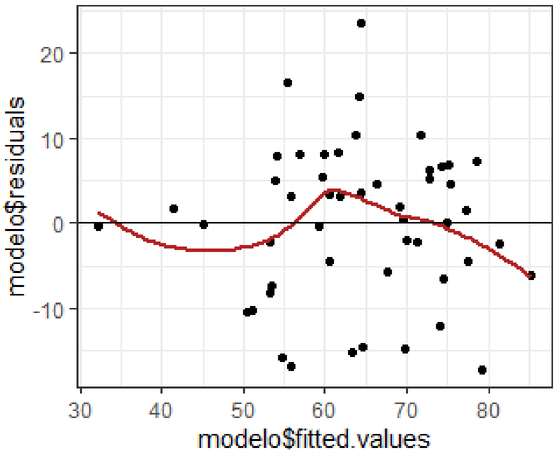


Figura 18: Residuos frente a los valores ajustados por el modelo

```
        studentized Breusch-Pagan test

data:  modelo
BP = 2.5486, df = 3, p-value = 0.4666
```

Figura 19: Resultado de la prueba Breusch-Pagan

No hay evidencias de falta de homocedasticidad.

4.5 Código

Solución en R

5 Análisis de Varianza (ANOVA)

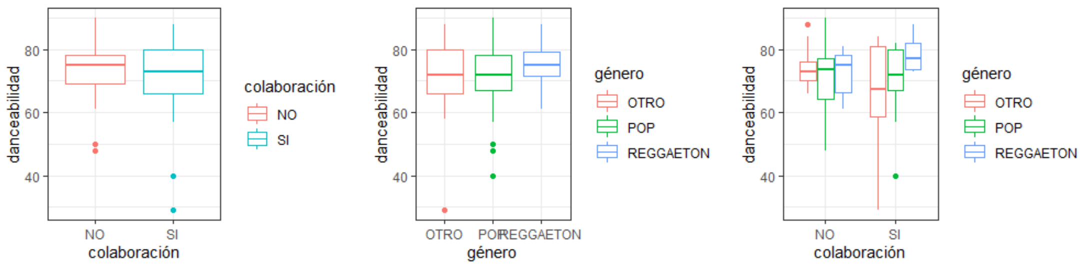
En este capítulo haremos un estudio acerca de la influencia que tienen el género musical (pop, reggaeton, otros) y la realización de colaboraciones (featuring) sobre el nivel de bailabilidad en canciones. Al concluir es capítulo llegaremos a conclusiones extremadamente interesantes. En el epígrafe 5.3 pondremos explícitamente las pruebas de hipótesis para el análisis de los residuos omitidas en los capítulos anteriores por su sencillez.

5.1 Representación gráfica de las observaciones

En primer lugar se generan los diagramas box-plot para identificar posibles diferencias significativas, asimetrías, valores atípicos y homogeneidad de varianza entre los distintos niveles. Se puede acompañar a los gráficos con las medias y varianza de cada grupo.

```
ggplot(data = datos, aes(x = colaboracion, y = danceabilidad, color = colaboracion)) +
  geom_boxplot() +
  labs(title = "", x = "colaboración", y = "danceabilidad", color = "colaboración") +
  theme_bw()
```

```
ggplot(data = datos, aes(x = genero, y = danceabilidad, color = genero)) +
  geom_boxplot() +
  labs(title = "", x = "género", y = "danceabilidad", color = "género") +
  theme_bw()
ggplot(data = datos, aes(x = colaboracion, y = danceabilidad,
color = genero)) +
  geom_boxplot() +
  labs(title = "", x = "colaboración", y = "danceabilidad", color = "género") +
  theme_bw()
```



(a) Danceabilidad vs Colaboración (b) Danceabilidad vs Género (c) Danceabilidad vs Colaboración vs Género

Figura 20: Box-plot de las observaciones estudiadas

A partir de la representación gráfica podemos intuir que existe una diferencia no muy marcada en los índices de bailabilidad dependiendo de la existencia o no de una colaboración y del género de la canción. El nivel de bailabilidad parece ser mayor en canciones cuyo género es el reggaeton que en otros géneros y en temas que no poseen colaboraciones, aunque la significancia se tendrá que confirmar con el ANOVA. A priori parece que se satisfacen las condiciones necesarias para un ANOVA, aunque habrá que confirmarlas estudiando los residuos. También es posible identificar algunas interacciones de los dos factores de forma descriptiva mediante gráficos de interacción. Si las líneas que describen los datos para cada uno de los niveles son paralelas significa que el comportamiento es similar independientemente del nivel del factor, es decir, no hay interacción.

```
ggplot(data = datos, aes(x = colaboracion, y = danceabilidad, colour = genero,
                         group = genero)) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") +
  labs(title = "", x = "colaboración", y = "media␣de␣danceabilidad",
       colour = "género") +
  theme_bw()
ggplot(data = datos, aes(x = genero, y = danceabilidad, colour = colaboracion,
                         group = colaboracion)) +
  stat_summary(fun.y = mean, geom = "point") +
  stat_summary(fun.y = mean, geom = "line") +
  labs(title = "", x = "género", y = "media␣de␣danceabilidad",
       colour = "colaboración") +
  theme_bw()
```

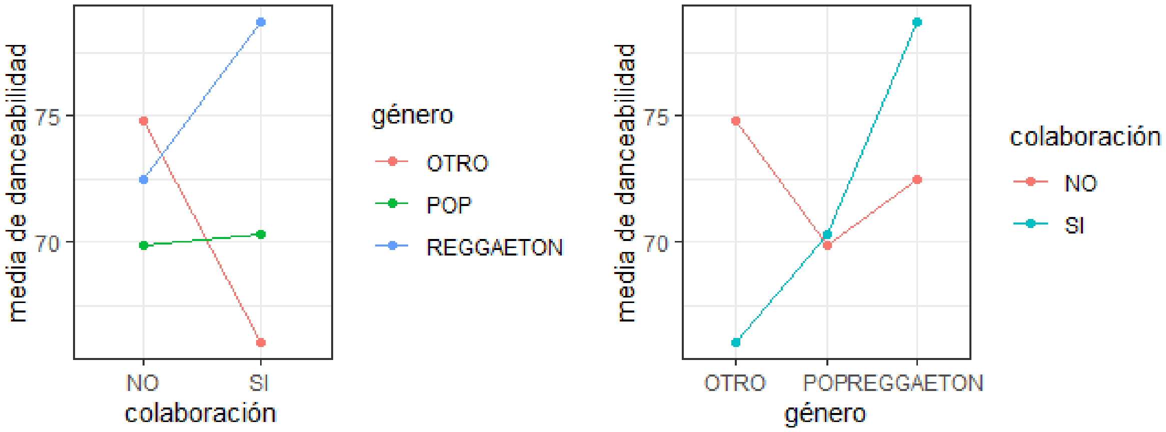


Figura 21: Interacciones entre las observaciones

Se observa una clara interacción entre ambos factores. La respuesta a la bailabilidad es distinta según la colaboración y el género. En canciones con más de un artista, la respuesta es mayor cuando las canciones pertencen al reggaeton que cuando pertenecen a otros géneros musicales, sin embargo, cuando no hay colaboración el reggaeton, como género, no es el más bailable. El ANOVA permite saber si las diferencias observadas son significativas.

5.2 Generar el modelo ANOVA

A continuación, llevaremos a cabo el análisis de varianzas, el cual nos permite saber si las diferencias observadas son significativas.

```
anova <- aov(danceabilidad ~ colaboracion * genero, data = datos)
summary(anova)
etaSquared(anova)
```

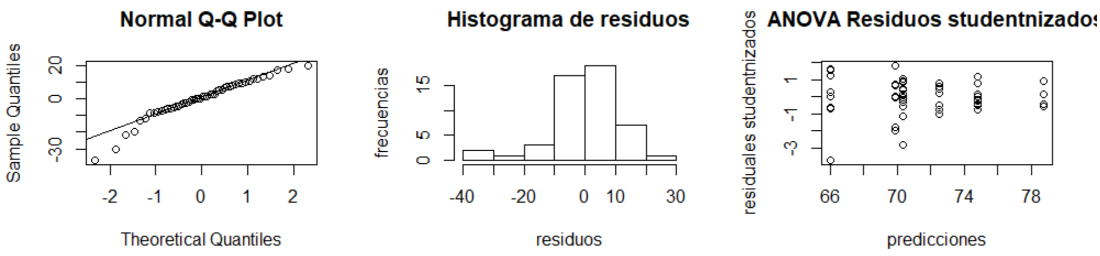
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
colaboracion	1	61	60.50	0.417	0.522
genero	2	132	66.18	0.457	0.636
colaboracion:genero	2	403	201.48	1.390	0.260
Residuals	44	6378	144.95		

Figura 22: Resumen del ANOVA

	eta.sq	eta.sq.part
colaboracion	0.00408064	0.004442035
genero	0.01898067	0.020331888
colaboracion:genero	0.05778261	0.059426095

Figura 23: Resumen del ANOVA

El análisis de varianza no encuentra diferencias significativas en los valores de la bailabilidad por parte del factor colaboración, pero sí encuentra diferencias significativas sobre el género y entre al menos dos grupos de las combinaciones de colaboración y género, es decir, hay significancia para la interacción. Además, el orden en el que se multiplican los factores no afecta, únicamente si el tamaño de los grupos es igual, de lo contrario sí afecta.



(a) Normalidad (b) Independencia (c) Homocedasticidad

Figura 24: Gráficos del análisis de los residuos para el ANOVA

5.3 Análisis de los residuos

Para poder dar por válidos los resultados del ANOVA es necesario verificar que se satisfacen los supuestos.

Supuestos del modelo:

1. Los residuos siguen una distribución normal con media 0.
2. Los residuos son independientes entre sí.
3. Los residuos tienen la misma varianza σ^2.

5.3.1 Distribución normal de los residuos

Como se muestra en la figura 24 inciso a), los residuos siguen una distribución normal ya que están alineados en una línea recta, con ligeras diferencias en los primeros y los últimos. Para una mejor comprensión realizaremos la prueba de *Shakiro-Wilk*, que se plantea de la siguiente manera:

$$H_0 : \text{los datos no proceden de una distribución normal}$$
$$H_1 : \text{los datos proceden de una distribución normal}$$

```
residuales <- anova$residuals
shapiro.test(residuales)
```

```
        Shapiro-Wilk normality test

data:  residuales
W = 0.94204, p-value = 0.01624
```

Figura 25: Resultado de la prueba Shakiro-Wilk

Como $p - valor = 0.01624 \approx 0.016 < 0.05$, entonces podemos rechazar la hipótesis nula y aceptar la hipótesis alternativa, por lo que aceptaríamos la hipótesis de normalidad en los residuos.

5.3.2 Independencia entre los residuos

Para verificar la independencia entre residuos consecutivos llevaremos a cabo la prueba de Durbin-Watson.

$$H_0 : \text{los residuos no son independientes}$$
$$H_1 : \text{los residuos son independientes}$$

```
dwtest(anova)
```

```
        Durbin-Watson test

data:  anova
DW = 2.1707, p-value = 0.7564
alternative hypothesis: true autocorrelation is greater than 0
```

Figura 26: Resultado de la prueba Durbin-Watson

Como $p - valor = 0.7564 > 0.05$, entonces la prueba de independencia no es significativa, por lo que no podemos rechazar H_0 ni la independencia entre los residuos.

5.3.3 Varianza constante de los residuos (homocedasticidad)

Una prueba para verificar si las varianzas son constantes es la de prueba de Bartlett para comprobar la homocedasticidad de las varianzas.

$$H_0 : \text{los residuos no tienen la misma varianza}$$
$$H_1 : \text{los residuos tienen la misma varianza}$$

```
bartlett.test(residuales, colaboracion)
bartlett.test(residuales, datos$genero)
```

```
        Bartlett test of homogeneity of variances

data:  residuales and colaboracion
Bartlett's K-squared = 2.1587, df = 1, p-value = 0.1418
```

(a) Homocedasticidad para la observación colaboración

```
        Bartlett test of homogeneity of variances

data:  residuales and datos$genero
Bartlett's K-squared = 4.7274, df = 2, p-value = 0.09407
```

(b) Homocedasticidad para la observacioón género

Figura 27: Resultados de las pruebas Bartlett

En las dos pruebas efectuadas se cumple que sus respectivos $p - valores$ son mayores que el nivel de significación usado (0.05), entonces no podemos rechazar la hipótesis nula (H_0) ni aceptar la hipótesis alternativa (H_1), por lo que no se cumple el supuesto planteado.

5.4 Código

Solución en R

6 Reducción de Dimensiones

Las razones por las nos interesarían reducir la dimensionalidad son varias:

1. Nos interesa identificar y eliminar las variables irrelevantes.
2. No siempre el mejor modelo es el que más variables tiene en cuenta.
3. Se mejora el rendimiento computacional, lo que se traduce en un ahorro en costo y tiempo.
4. Se reduce la complejidad, lo que lleva a facilitar la comprensión del modelo y sus resultados.

6.1 Análisis de los datos

Primeramente estudiaremos la correlación de nuestra muestra. Para ello, calcularemos la matriz de correlación con el objetivo de conocer si existe algún tipo de correlación entre los datos.

```
m <- cor(datos)
print(m)
```

```
                 Beats_Per_Minute      Energy Danceability Loudness.dB.    Liveness
Beats_Per_Minute       1.00000000  0.04375559 -0.0941828916   0.01701619 -0.16728576
Energy                 0.04375559  1.00000000  0.0182535758   0.67079357  0.16276771
Danceability          -0.09418289  0.01825358  1.0000000000   0.01625454 -0.14963620
Loudness.dB.           0.01701619  0.67079357  0.0162545370   1.00000000  0.25865203
Liveness              -0.16728576  0.16276771 -0.1496362023   0.25865203  1.00000000
Valence               -0.01158582  0.43881959  0.1728289768   0.23761380  0.01612347
Length                -0.13928840  0.22467681 -0.0001852976   0.21921874  0.13178234
Acousticness          -0.03144960 -0.33989165 -0.0981653774  -0.13829961  0.02132824
Speechiness            0.55705188 -0.08985967 -0.1034719217  -0.27221263 -0.12528606
Popularity             0.19609692 -0.08029497 -0.0714132526  -0.04308543  0.09256423
                     Valence        Length Acousticness  Speechiness  Popularity
Beats_Per_Minute -0.01158582 -0.1392883997 -0.031449597  0.557051878  0.19609692
Energy            0.43881959  0.2246768064 -0.339891653 -0.089859668 -0.08029497
Danceability      0.17282898 -0.0001852976 -0.098165377 -0.103471922 -0.07141325
Loudness.dB.      0.23761380  0.2192187407 -0.138299607 -0.272212633 -0.04308543
Liveness          0.01612347  0.1317823354  0.021328241 -0.125286062  0.09256423
Valence           1.00000000 -0.0177817772 -0.052323306 -0.053241746 -0.31775236
Length           -0.01778178  1.0000000000 -0.076292690  0.046755261 -0.08763886
Acousticness     -0.05232331 -0.0762926901  1.000000000  0.008293376 -0.03468404
Speechiness      -0.05324175  0.0467552609  0.008293376  1.000000000  0.23855303
Popularity       -0.31775236 -0.0876388589 -0.034684041  0.238553032  1.00000000
```

Figura 28: Matriz de correlación

Si bien la matriz de correlación es efectiva, suele ser incómodo tratar de ver si es una matriz altamente correlacionada o no a simple vista. Sin embargo, tenemos la función *symnum* que si le pasamos una matriz de correlación no dará de forma gráfica si está o no altamente correlacionada esta matriz.

```
symnum(m)
```

```
                 B E D L. Lv V Ln A S P
Beats_Per_Minute 1
Energy             1
Danceability         1
Loudness.dB.       ,   1
Liveness                  1
Valence            .         1
Length                         1
Acousticness       .              1
Speechiness      .                  1
Popularity                   .        1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```

Figura 29: Matriz de correlación en forma gráfica

6.2 Análisis de las Componentes Principales (ACP)

Como se puede observar en la leyenda de la figura anterior, tendremos una matriz altamente correlacionada mientras más valores tengamos marcados con signos +, ? y *B*. En este caso predominan los espacios en blanco, aunque aparecen comas y puntos también, por lo que podemos decir que no está altamente correlacionada. En este caso ya las variables son independientes por lo que este análisis solo serviría para reducir dimensión. Así que podemos proseguir a realizar el análisis de componentes principales.

```
acp <- prcomp(datos, scale = TRUE)
summary(acp)
```

```
Importance of components:
                          PC1    PC2    PC3    PC4     PC5     PC6     PC7     PC8
Standard deviation     1.5203 1.2818 1.1709 1.0261 0.98859 0.91483 0.84647 0.73996
Proportion of Variance 0.2311 0.1643 0.1371 0.1053 0.09773 0.08369 0.07165 0.05475
Cumulative Proportion  0.2311 0.3955 0.5325 0.6378 0.73557 0.81926 0.89091 0.94566
                           PC9    PC10
Standard deviation     0.56540 0.47296
Proportion of Variance 0.03197 0.02237
Cumulative Proportion  0.97763 1.00000
```

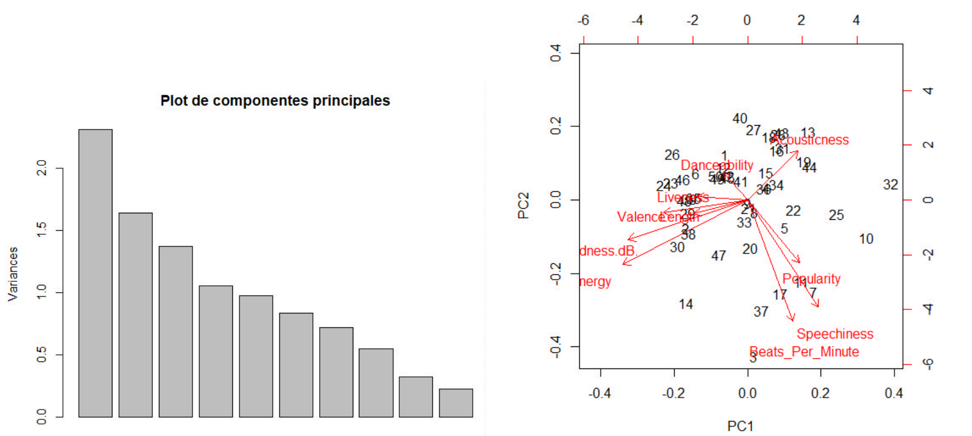
Figura 30: Importancia de los componentes

Un detalle a tener en cuenta sería la magnitud de las variables, en este caso necesitamos estandarizar, por lo que el parámetro *scale* de la función *prcomp* debería ser *TRUE*. Además, en el sumario de la función los valores propios aparecen marcados como *Standard Desviation*, cuando en inglés su significado es *Eigenvalue*.

6.3 Gráficos

Una vez realizado este análisis necesitamos saber a quién escoger para obtener nuestras componentes principales. Para eso necesitamos ver la proporción acumulativa (*accumulative proportion*), en el caso de la primera componente $PC1$ es aproximadamente igual a 0.23, que solo explicaría un 23 %, por lo que necesitamos al menos otra componente para tener un valor mayor al 70 %. Con las tres primeras componentes obtenemos lo explicado con anterioridad, además de acuerdo con al criterio de Kaiser tenemos que las tres primeras componentes presentan valores propios superiores a la unidad. Por lo que, $PC1$, $PC2$ y $PC3$ son nuestras componentes principales. Otra forma de corroborar que necesitamos solo las tres primeras componentes sería graficarlas, como se muestra a continuación.

```
plot(acp, main = "Plot␣de␣componentes␣principales")
biplot(acp)
```



(a) Plot de componentes principales (b) Biplot del ACP

Figura 31: Gráficos

A la derecha podemos ver un biplot del análisis de las componentes principales, el cual es una representación, en un mismo gráfico, de las filas y las columnas de una matriz de datos $X(n \ x \ p)$. Suponiendo X matriz centrada, el biplot clásico se lleva a cabo mediante la descomposición singular de la matriz. La interpretación de estos gráficos nos ofrece más información sobre la muestra.

6.4 Interpretación de los datos

El siguiente paso es la interpretación de los datos para el beneficio de la investigación, para eso debemos buscar la matriz de valores propios y así sabremos qué variable es importante para cada componente y en qué medida.

```r
print(acp$rotation)
```

	PC1	PC2	PC3	PC4	PC5
Beats_Per_Minute	0.1934460	-0.61085212	0.17158426	-0.18639314	-0.075763204
Energy	-0.5306202	-0.32647323	0.02788703	-0.01588672	-0.070737871
Danceability	-0.1017655	0.13802918	0.45698210	0.46549518	-0.001331891
Loudness.dB.	-0.5083115	-0.19896377	-0.16474817	-0.05428867	-0.160248127
Liveness	-0.2163808	0.01571758	-0.56411697	-0.15868259	-0.158116342
Valence	-0.3582471	-0.06491833	0.44246650	-0.37625549	-0.093329153
Length	-0.2297225	-0.07006150	-0.24413901	0.12058166	0.853661783
Acousticness	0.2143396	0.24842138	-0.10714696	-0.63964477	0.037633078
Speechiness	0.3003760	-0.53866050	0.09836552	-0.12848430	0.297010694
Popularity	0.2195057	-0.31685632	-0.37288322	0.37206216	-0.333906731

	PC6	PC7	PC8	PC9	PC10
Beats_Per_Minute	-0.04348913	0.16251350	0.44730645	0.48615827	0.234050980
Energy	0.06303420	0.10078169	-0.16972107	-0.40268418	0.633035111
Danceability	-0.68225803	-0.03777300	0.25777056	-0.08620955	0.065505687
Loudness.dB.	-0.10948986	0.46578680	0.19800179	-0.09022042	-0.608255170
Liveness	-0.28151774	-0.61206920	0.35140301	0.03956224	0.074787820
Valence	-0.13549853	-0.35889380	-0.45282748	0.36332180	-0.192793687
Length	-0.14499743	0.07975829	-0.11437656	0.31072048	0.049212612
Acousticness	-0.51829096	0.37427157	-0.10974648	-0.13163271	0.179518846
Speechiness	-0.14052913	-0.29416133	-0.05555005	-0.55049880	-0.304640353
Popularity	-0.33115108	0.09296954	-0.55561897	0.18823538	-0.002018759

Figura 32: Matriz de valores propios λ_i

Para la interpretación de los datos nos apoyaremos fundamentalmente en la matriz de los valores propios (figura 32).

Comencemos por la primera componente, tomamos el mayor valor propio sin importar el signo 0.530 y lo dividimos por 2, esto da 0.265, en consecuencia todo valor propio cuyo módulo esté por encima de 0.265 en la columna de la $PC1$ nos dará las variables que conforman esta componente. Por tanto, la interpretación sería que la $PC1$ está caracterizada por canciones con una marcada energía, sonoridad, positividad y con gran cantidad de texto hablado. Siguiendo el mismo análisis y algoritmo para la segunda componente, $PC2$, obtenemos canciones con gran popularidad, factor de voz, energía y un fuerte golpe de beats por minutos. Procedemos de la misma manera para la tercera componente, $PC3$, por lo que llegamos a la conclusión que dicha componente está formada con temas alegres, bailables, populares y con gran probabilidad de ser grabados en directos.

Para una mejor comprensión y entendimiento de lo anterior veremos el cuadro 1, en el cual presentaremos las observaciones de cada componente. Los resultados de este análisis muestran canciones enérgicas, populares, no muy largas, felices y con ritmo, aunque con una marcada falta de duración y de acústica.

6.5 Código

Solución en R

Observación	PC1	PC2	PC3
Cantidad de golpes por minutos	NO	SÍ	NO
Energía	SÍ	SÍ	NO
Bailabilidad	NO	NO	SÍ
Sonoridad	SÍ	NO	NO
Factor de directo	NO	NO	SÍ
Positividad	SÍ	NO	SÍ
Duración	NO	NO	NO
Acústica	NO	NO	NO
Factor de voz recitada	SÍ	SÍ	NO
Popularidad	NO	SÍ	SÍ

Cuadro 1: Observaciones vs Componentes

7 Clusters

La agrupación es una técnica para agrupar puntos de datos similares en un grupo y separar las diferentes observaciones en diferentes grupos. Los *clusters* se crean de manera que tengan un orden predeterminado, es decir, una jerarquía. Estas jerarquías de *clústeres* puede crearse de arriba a abajo o viceversa. Por lo tanto, son dos tipos: *divisivo* y *aglomerativo*.

7.1 Preparando los datos

Para realizar la agrupación, los datos deben prepararse de acuerdo con las siguientes pautas: las filas deben contener observaciones, las columnas deben ser variables, los datos no pueden tener valores faltantes, los datos en las columnas deben estar estandarizados o escalados, para que las variables sean comparables y este análisis se realiza a variables cuantitativas solamente. Todo lo expresado se tiene en cuenta es el siguiente listado de código.

```
quitar_variables_cualitativas <- function(datos)
{
   datos <- datos[, -18]
   datos <- datos[, -17]
   datos <- datos[, -16]
   datos <- datos[, -14]
   datos <- datos[, -3]
   datos <- datos[, -2]
   datos <- datos[, -1]
   return(datos)
}

datos <- read.table(file.choose(), header = TRUE)
datos <- quitar_variables_cualitativas(datos)
datos <- scale(datos)
```

La última línea del listado de código mostrado anteriormente escala todas las variables numéricas. Esto significa que cada variable tendrá una media cero y una desviación estándar de uno. Esto se hace para evitar que el algoritmo de agrupamiento depende de una unidad variable aleatoria.

7.2 Agrupamiento jerárquico aglomerativo

El agrupamiento jerárquico aglomerativo también es conocido como aglomeración aglomerativa jerárquica (HAC) o AGNES (acrónimo de aglomeración de anidación). En este método, cada observación se asigna a su propio clúster. Luego, se calcula la similitud (o distancia) entre cada uno de los clusters y los dos clusters más similares se fusionan en uno. Finalmente, se repite este paso hasta que solo quede un grupo. Para la función *hclust*, se requieren los valores de distancia que se pueden calcular en *R* utilizando la función *dist*. La medida predeterminada para la función *dist* es *euclidiana*, sin embargo, puede cambiarse con el argumento del método. Con esto, también necesitamos especificar el método de vinculación que queremos usar (es decir, *completo*, *promedio* o *único*).

```
d <- dist(datos, method = "euclidean")
hc1 <- hclust(d, method = "complete" )
plot(hc1, cex = 0.6, hang = -1, main = "", xlab = "", ylab = "")
```

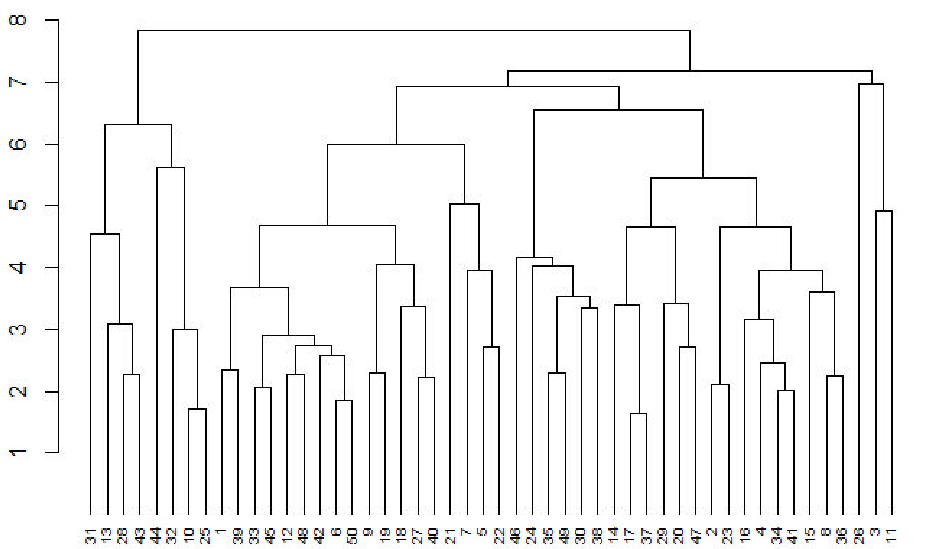
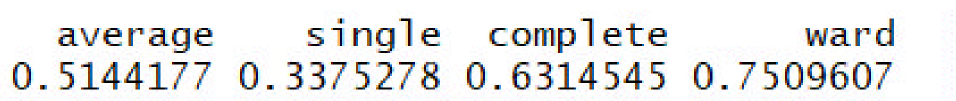


Figura 33: Denograma

Otra alternativa es la función *agnes*. Ambas funciones son bastante similares; sin embargo, con la función *agnes* también puede obtenerse el coeficiente de aglomeración, que mide la cantidad de estructura de agrupamiento encontrada (los valores más cercanos a 1 sugieren una estructura de agrupación fuerte). A continuación realizaremos un experimento con el objetivo de identificar el mejor método de vinculación.

```
ac <- function(metodo)
{
    agnes(datos, method = metodo)$ac
}

metodos <- c("average", "single", "complete", "ward")
names(metodos) <- c("average", "single", "complete", "ward")
map_dbl(metodos, ac)
```

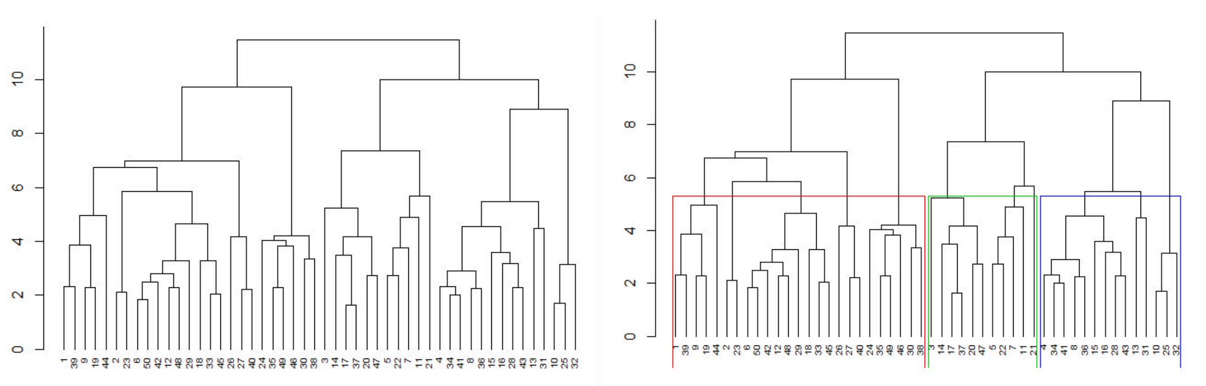


```
  average    single  complete      ward
0.5144177 0.3375278 0.6314545 0.7509607
```

Figura 34: Resultados del experimento

Podemos llegar a la conclusión que el parámetro para el método de vinculación es *ward* ya que es el valor más cercano a la unidad. A continuación haremos nuevamente un agrupamiento jerárquico aglomerativo con el nuevo parámetro y resaltaremos los grupos confeccionados.

```
hc3 <- agnes(datos, method = "ward")
pltree(hc3, cex = 0.6, hang = -1, main = "", xlab = "", ylab = "")
pltree(hc3, hang = -1, cex = 0.6)
rect.hclust(hc3, k = 3, border = 2:10)
```



(a) Denograma con el parámetro *ward*

(b) Denograma resaltando los grupos formados

Figura 35: Denogramas

El método de *ward* apunta a minimizar la varianza total dentro del grupo. En cada paso, se fusionan el par de clústeres con una distancia mínima entre los clústeres. En otras palabras, forma grupos de una manera que minimiza la pérdida asociada con cada grupo. En cada paso, se considera la unión de cada par de clústeres posible y se combinan los dos clusters cuya fusión da como resultado un aumento mínimo en la pérdida de información. Por otro lado, nos percatamos que existen características muy distintivas y específicas por parte de las canciones formadas en los tres grupos (figura 35 inciso b).

7.3 Agrupamiento divisional jerárquico

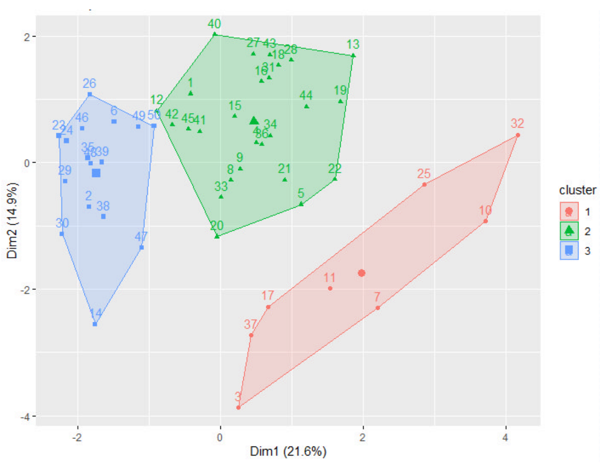
En el método divisivo suponemos que todas las observaciones pertenecen a un único grupo y luego dividimos el clúster en dos grupos menos similares. Esto se repite recursivamente en cada grupo hasta que haya un grupo para cada observación. Esta técnica también se llama *DIANA*, llamada así por ser el acrónimo de *análisis divisivo*.

Los algoritmos de partición son enfoques de agrupamiento que dividen los conjuntos de datos, que contienen n observaciones, en un conjunto de k grupos. Los algoritmos requieren que el analista especifique el número de clústeres que se generarán.

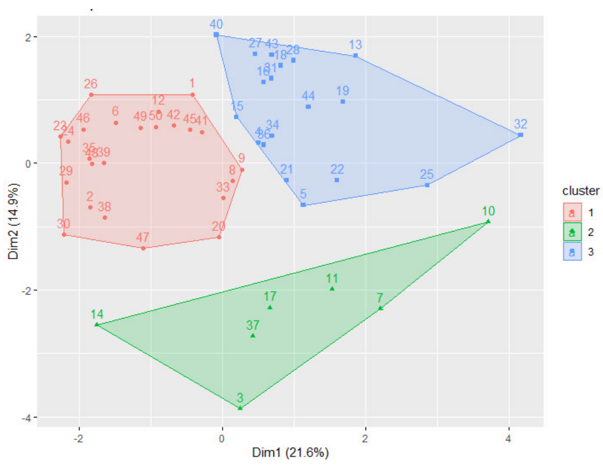
1. Algoritmo k-means utilizado por primera vez por James MacQueen en 1967, y tiene como objetivo la partición de un conjunto de n observaciones en k grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano.

2. Algoritmo k-medoids o PAM (partición alrededor de medoids) desarrollado por Kaufman & Rousseeuw en 1990, en el que, cada grupo está representado por uno de los objetos en el grupo. Tanto el k-medoids como el k-means son algoritmos que trabajan con particiones (dividiendo el conjunto de datos en grupos) y ambos intentan minimizar la distancia entre puntos que se añadirían a un grupo y otro punto designado como el centro de ese grupo. En contraste con el algoritmo k-means, k-medoids escoge datapoints como centros y trabaja con una métrica arbitraria de distancias entre datapoints.

```
cluster_kmeans <- kmeans(datos, 3, nstart = 50)
fviz_cluster(cluster_kmeans, data = datos, frame.type = "convex")
cluster_pam <- pam(datos, 3)
fviz_cluster(cluster_pam)
```



(a) Usando el algoritmo k-means



(b) Usando el algoritmo k-medoids

Figura 36: Gráficos

Una limitación de k-means es que se espera que los grupos sean separables, con forma esférica y de tamaño similar. No se garantiza que k-means llegue siempre a la solución óptima debido a que el resultado final va a depender de los centroides iniciales. Por otra parte, ambos métodos esperan que se les sea especificado el número de grupos a formar.

7.4 Código

Solución en R