

Probabilidad y estadística. B7

DISEÑO: Diversidad lingüística en la música de Europa

OBJETIVO

El objetivo de nuestro trabajo es descubrir mediante las listas TOP 50 de Spotify si en la Europa Este y en la Oeste se escucha más música en la lengua propia de la región, o en consecuencia son más populares las canciones extranjeras. Además de esto, averiguar donde hay más diversidad musical comparando el número de lenguas distintas de las canciones.

VARIABLES

Las variables escogidas son las siguientes:

- **Variable binaria:**

Zona de Europa (x), indica a qué zona de Europa pueden pertenecer las canciones. Sus valores son Este y Oeste.

- **Variables numéricas (VADs):**

Porcentaje de canciones propias (y), indica el tanto por ciento de canciones cantadas en la lengua propia del país al cual pertenece la canción. pertenecientes a una misma zona que

Número de idiomas únicos (z), es la cantidad de canciones habladas en una lengua diferente a la oficial de ese país (o zona).

PLAN DE RECOGIDA

En nuestra recogida de datos, usaremos una web que permite acceder a las listas TOP 50 de Spotify por países y fecha. De todos los países que tienen lista, seleccionaremos solo los que pertenezcan a Europa y los clasificaremos en dos bloques:

Europa Este incluye: Bulgaria, Rusia, Ucrania, Eslovaquia, Rumanía, Hungría, Polonia, la República Checa, Letonia, Lituania y Estonia.

Europa Oeste incluye: Portugal, Alemania, Austria, Bélgica, Dinamarca, España, Finlandia, Francia, Grecia, Irlanda, Islandia, Italia, Noruega, Países Bajos, Reino Unido, Suiza, Suecia y Luxemburgo.

Por lo tanto, al haber 29 países (11 en Europa Este y 18 en Europa Oeste), habrá como máximo 1450 canciones diferentes. Pero ese no es el valor real debido a que hay canciones que se escuchan en varios países. Por lo que después de analizar las listas y unir las en dos bloques (Este, Oeste), el número de canciones únicas pasa a ser 608.

Todos los datos pertenecen a las listas TOP 50 del día 20/11/2020, y se han obtenido mediante el siguiente enlace: <https://spotifycharts.com/regional>.

	A	B	C	D	E	F	G	H	I	J	K
1	SÃo Paulo	portugues	Portugal								
2	Lemonade (feat. Gunr	ingles	Portugal	Alemania	Austria						
3	Mood (feat. iann dior)	ingles	Portugal	Alemania	Austria						
4	inesquecível	portugues	Portugal								
5	Lonely (with Benny bl	ingles	Portugal	Alemania	Austria	Bélgica	Dinamarca				
6	Fica	portugues	Portugal								
7	positions	ingles	Portugal	Alemania	Austria	Bélgica	Dinamarca		Finlandia	Francia	Grecia
8	Therefore I Am	ingles	Portugal	Alemania	Austria	Bélgica	Dinamarca		Finlandia		Grecia
9	What You Know Bout	ingles	Portugal	Alemania	Austria						
10	For The Night (feat. Lil	ingles	Portugal	Alemania	Austria						
11	Monster (Shawn Meni	ingles	Portugal	Alemania		Bélgica	Dinamarca				
12	Ronaldo	portugues	Portugal								
13	Dakiti	español	Portugal					España			
14	Hawái	español	Portugal					España			
15	Mood Suings (feat. Lil	ingles	Portugal								
16	Dynamite	koreano	Portugal	Alemania	Austria	Bélgica			Finlandia	Francia	Grecia
17	Prisoner (feat. Dua Lip	ingles	Portugal	Alemania							
18	Watermelon Sugar	ingles	Portugal		Austria						

Muestra de las primeras canciones y de sus repeticiones en varios países.

Entre todos los valores, se escogen la cantidad de canciones con lenguas pertenecientes a cada X y este será el valor de la variable Y.

Por último, para conseguir los datos de la variable Z, para todas aquellas canciones que no pertenezcan a la variable Y, se contará el número de distintas lenguas para cada X.

El siguiente código escrito en C++, facilitará la obtención de los datos al hacer el recuento de idiomas:

```
#include <iostream>
#include <map>
```

```
#include <string>
using namespace std;

int main(){
    string idioma;
    map<string, int> total;
    map<string, int>:: iterator it;
    while(cin >> idioma and idioma != final){
        it = total.find(idioma);
        if(it == total.end()){
            total[idioma] = 1;
        }
        else ++(it->second);
    }
    it = total.begin();
    while(it != total.end()){
        cout << it->first << ' ' << it->second;
        ++it;
    }
}
```