Prueba Intertrimestral

Nombre:

Apellidos:

Tiempo de la prueba: 2 Horas

Asignatura: Desarrollo de Aplicaciones para la Visualización de Datos

Fecha: 18 de octubre de 2023

Instrucciones:

- Escribe código limpio y autoexplicativo.
- Se eliminará 0.5 puntos por usar Seaborn o Matplotlib.
- Se pueden utilizar los materiales de clase.
- Se puede utilizar internet para búsqueda de dudas y documentación.
- No se puede utilizar ningún tipo de LLM.
- No se puede utilizar mensajería instantánea.
- Sube tus resultados a tu repositorio de Github.
- Imprime una versión en PDF en A3 y Portrait del notebook.
- Envialo tus resultados a dmartincorral@icai.comillas.edu adjuntando el PDF y la url del notebook subido al repositorio de Github.

Inicialización de librerías

Carga aquí todas las librerías que vayas a utilizar.

In [116]:
```
import pandas as pd
import numpy as np
import plotly.graph_objects as go
import plotly.express as px

import sklearn.datasets
from sklearn.model_selection import train_test_split
from sklearn.svm import LinearSVC
from sklearn.metrics import (classification_report, mean_squared_error, mean_absolute_error, r2_score)
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

Ejercicio 1 (2 puntos):

a) Crea una función que calcule y devuelva el factorial de un número entero. **(0.6 puntos)**

b) Crea una función que verifique si un número es primo o no. **(0.6 puntos)**

c) Muestra en un dataframe los 50 primeros números positivos, si es primo y su factorial utilizando las funciones anteriores. **(0.6 puntos)**

d) ¿Cómo se podría programar en una clase las tres operaciones anteriores? **(0.2 puntos)**

In [117]:
```python
# Apartado A
def factorial(n):
    if n <= 1:
        return 1
    else:
        return n*factorial(n-1)
factorial(5)
```

Out[117]:
```
120
```

In [118]:
```python
# Apartado B
def is_primo(n):
    divisible = False
    for i in range(n):
        if (i > 1) and (i != n):
            if n % i == 0:
                divisible = True
    return not divisible
print(f"El numero 5 {'no' if not is_primo(5) else ''} es primo")
print(f"El numero 10 {'no' if not is_primo(10) else ''} es primo")
```

```
El numero 5  es primo
El numero 10 no es primo
```

In [119]:
```
# Apartado C
df = pd.DataFrame()
numbers = []
primos = []
factorials = []
for i in range(50):
    numbers.append(i+1)
    primos.append(is_primo(i+1))
    factorials.append(factorial(i+1))
df["numbers"] = numbers
df["is_primo"] = primos
df["facotrials"] = factorials
df
```

Out[119]:

	numbers	is_primo	facotrials
0	1	True	1
1	2	True	2
2	3	True	6
3	4	False	24
4	5	True	120
5	6	False	720
6	7	True	5040
7	8	False	40320
8	9	False	362880
9	10	False	3628800
10	11	True	39916800
11	12	False	479001600

12	13	True	6227020800
13	14	False	87178291200
14	15	False	1307674368000
15	16	False	20922789888000
16	17	True	355687428096000
17	18	False	6402373705728000
18	19	True	121645100408832000
19	20	False	2432902008176640000
20	21	False	51090942171709440000
21	22	False	1124000727777607680000
22	23	True	25852016738884976640000
23	24	False	620448401733239439360000
24	25	False	15511210043330985984000000
25	26	False	403291461126605635584000000
26	27	False	10888869450418352160768000000
27	28	False	304888344611713860501504000000
28	29	True	8841761993739701954543616000000
29	30	False	265252859812191058636308480000000
30	31	True	8222838654177922817725562880000000
31	32	False	263130836933693530167218012160000000
32	33	False	8683317618811886495518194401280000000
33	34	False	295232799039604140847618609643520000000

34	35	False	10333147966386144929666651337523200000000
35	36	False	371993326789901217467999448150835200000000
36	37	True	13763753091226345046315979581580902400000000
37	38	False	523022617466601111760007224100074291200000000
38	39	False	203978820811974433586402817399028973568000000000
39	40	False	815915283247897734345611269596115894272000000000
40	41	True	33452526613163807108170062053440751665152000000...
41	42	False	14050061177528798985431426062445115699363840000...
42	43	True	60415263063373835637355132068513997507264512000...
43	44	False	26582715747884487680436258110146158903196385280...
44	45	False	11962222086548019456196316149565771506438373370...
45	46	False	55026221598120889498503054288002548929616517520...
46	47	True	25862324151116818064296435515361197996919763230...
47	48	False	12413915592536072670862289047373375038521486350...
48	49	False	60828186403426756087225216332129537688755283130...
49	50	False	30414093201713378043612608166064768844377641560...

In [120]:
```
class number():
    def __init__(self,value):
        self.value = value

    def factorial(self):
        if self.value <= 1:
            return 1
        else:
            return self.value*number(self.value-1).factorial()

    def is_primo(self):
        divisible = False
        for i in range(self.value):
            if (i > 1) and (i != self.value):
                if self.value % i == 0:
                    divisible = True
        return not divisible
numero = number(5)
print(numero.factorial(), numero.is_primo())
```

```
120 True
```

Ejercicio 2 (4 puntos):

a) Extrae de sklearn el conjunto de datos **California Housing dataset** y transfórmalo a dataframe de pandas **(0.25 puntos)**

b) Construye una función que muestra la estructura del dataset, el número de NAs, tipos de variables y estadísticas básicas de cada una de las variables. **(0.5 puntos)**

c) Construye una **Regresión lineal** y un **Random forest** que predigan el **Median house value** según los datos disponibles. **(0.75 puntos)**

d) Visualiza cuales son las variables (coeficientes) más importantes en cada uno de los modelos. **(1.25 puntos)**

e) Decide a través de las métricas que consideres oportunas, cuál de los dos modelos es mejor, por qué y explica el proceso que has realizado para responder en los puntos anteriores. **(1.25 puntos)**

In [121]:
```
# Apartado A
dataset = sklearn.datasets.fetch_california_housing()
df = pd.DataFrame(data = dataset['data'], columns = dataset['feature_names'])
df["MedianHouseValue"] = dataset["target"]
df
```

Out[121]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedianHouseValue
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422
...	...	...	...	...	...	...	...	...	...
20635	1.5603	25.0	5.045455	1.133333	845.0	2.560606	39.48	-121.09	0.781
20636	2.5568	18.0	6.114035	1.315789	356.0	3.122807	39.49	-121.21	0.771
20637	1.7000	17.0	5.205543	1.120092	1007.0	2.325635	39.43	-121.22	0.923
20638	1.8672	18.0	5.329513	1.171920	741.0	2.123209	39.43	-121.32	0.847
20639	2.3886	16.0	5.254717	1.162264	1387.0	2.616981	39.37	-121.24	0.894

20640 rows × 9 columns

In [122]:
```
# Apartado B
def dataset_struct(df, reg_class):
    print(df.describe()) # Descriptivo del dataset
    print(df.isna().sum()) # Ver si el dataset tiene NAs
    if reg_class == 'class':
        print(df["target"].value_counts(normalize = True)) # Ver si las clases están balanceadas (sólo t
iene sentido si es un problema de clasificación)
    print(df.dtypes) # Tipos de las columnas
dataset_struct(df, 'reg') # Nuestro dataset sirve para hacer una regresión, por lo que no tiene clases
```

```
             MedInc      HouseAge      AveRooms     AveBedrms    Population  \
count  20640.000000  20640.000000  20640.000000  20640.000000  20640.000000
mean       3.870671     28.639486      5.429000      1.096675   1425.476744
std        1.899822     12.585558      2.474173      0.473911   1132.462122
min        0.499900      1.000000      0.846154      0.333333      3.000000
25%        2.563400     18.000000      4.440716      1.006079    787.000000
50%        3.534800     29.000000      5.229129      1.048780   1166.000000
75%        4.743250     37.000000      6.052381      1.099526   1725.000000
max       15.000100     52.000000    141.909091     34.066667  35682.000000

            AveOccup      Latitude     Longitude  MedianHouseValue
count  20640.000000  20640.000000  20640.000000      20640.000000
mean       3.070655     35.631861   -119.569704          2.068558
std       10.386050      2.135952      2.003532          1.153956
min        0.692308     32.540000   -124.350000          0.149990
25%        2.429741     33.930000   -121.800000          1.196000
50%        2.818116     34.260000   -118.490000          1.797000
75%        3.282261     37.710000   -118.010000          2.647250
max     1243.333333     41.950000   -114.310000          5.000010
MedInc              0
HouseAge            0
AveRooms            0
AveBedrms           0
Population          0
AveOccup            0
```

```
Latitude            0
Longitude           0
MedianHouseValue    0
dtype: int64
MedInc              float64
HouseAge            float64
AveRooms            float64
AveBedrms           float64
Population          float64
AveOccup            float64
Latitude            float64
Longitude           float64
MedianHouseValue    float64
dtype: object
```

In [123]:
```
# Apartado C
# Dividir train-test
X = dataset["data"]
y = dataset["target"]
X_train, X_test, y_train, y_test = train_test_split(X, y ,test_size = 0.3, random_state = 123)

# Hacemos el modelo de regresión lineal
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
lin_reg_predictions = lin_reg.predict(X_test)
lin_reg_predictions_train = lin_reg.predict(X_train)

# Evaluamos el modelo
# Metricas de evaluación
lin_reg_rmse_train = np.sqrt(mean_squared_error(y_train,lin_reg_predictions_train))
lin_reg_mae_train = mean_absolute_error(y_train, lin_reg_predictions_train)
lin_reg_r2_train = r2_score(y_train, lin_reg_predictions_train)

lin_reg_rmse_test = np.sqrt(mean_squared_error(y_test,lin_reg_predictions))
lin_reg_mae_test = mean_absolute_error(y_test, lin_reg_predictions)
```

```
lin_reg_r2_test = r2_score(y_test, lin_reg_predictions)

print("El RMSE de train del modelo de regresión lineal es: {}".format(lin_reg_rmse_train))
print(f"El MAE de train del modelo de regresión lineal es: {lin_reg_mae_train}")
print(f"El R2 de train del modelo de regresión lineal es: {lin_reg_r2_train}")

print("")

print("El RMSE de test del modelo de regresión lineal es: {}".format(lin_reg_rmse_test))
print(f"El MAE de test del modelo de regresión lineal es: {lin_reg_mae_test}")
print(f"El R2 de test del modelo de regresión lineal es: {lin_reg_r2_test}")


print("-------------------------------------------------------------------------------------")
# Hacemos el modelo de random forest
rd_for = RandomForestRegressor()
rd_for.fit(X_train, y_train)
rd_for_predictions = rd_for.predict(X_test)
rd_for_predictions_train = rd_for.predict(X_train)

# Evaluamos el modelo
# Metricas de evaluación
rd_for_rmse_train = np.sqrt(mean_squared_error(y_train,rd_for_predictions_train))
rd_for_mae_train = mean_absolute_error(y_train, rd_for_predictions_train)
rd_for_r2_train = r2_score(y_train, rd_for_predictions_train)

rd_for_rmse_test = np.sqrt(mean_squared_error(y_test,rd_for_predictions))
rd_for_mae_test = mean_absolute_error(y_test, rd_for_predictions)
rd_for_r2_test = r2_score(y_test, rd_for_predictions)

print("El RMSE de train del modelo de random forest es: {}".format(rd_for_rmse_train))
print(f"El MAE de train del modelo de random forest es: {rd_for_mae_train}")
print(f"El R2 de train del modelo de random forest es: {rd_for_r2_train}")

print("")
```

```
print("El RMSE de test del modelo de random forest es: {}".format(rd_for_rmse_test))
print(f"El MAE de test del modelo de random forest es: {rd_for_mae_test}")
print(f"El R2 de test del modelo de random forest es: {rd_for_r2_test}")
```

```
El RMSE de train del modelo de regresión lineal es: 0.7265293808219301
El MAE de train del modelo de regresión lineal es: 0.5330981590557133
El R2 de train del modelo de regresión lineal es: 0.604714940991568

El RMSE de test del modelo de regresión lineal es: 0.7187326821064413
El MAE de test del modelo de regresión lineal es: 0.5288160032841038
El R2 de test del modelo de regresión lineal es: 0.6093458386889428
--------------------------------------------------------------------------------
El RMSE de train del modelo de random forest es: 0.19306759543846547
El MAE de train del modelo de random forest es: 0.12461571508859397
El R2 de train del modelo de random forest es: 0.9720859526475544

El RMSE de test del modelo de random forest es: 0.4983861350327727
El MAE de test del modelo de random forest es: 0.3268060369993542
El R2 de test del modelo de random forest es: 0.8121593546629549
```

In [124]:
```
# Apartado C
print(lin_reg.coef_)
print(rd_for)
df.corr() # Podemos ver que las variables que mejor explican el MedianHouseValue son el MedInc, HouseAg
e, AveRooms y, finalmente, la Latitude
```

```
[ 4.36326444e-01  9.17705383e-03 -1.04601995e-01  6.10289220e-01
 -3.32797843e-06 -3.71107412e-03 -4.21565906e-01 -4.34173462e-01]
RandomForestRegressor()
```

Out[124]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedianHouseValue
MedInc	1.000000	-0.119034	0.326895	-0.062040	0.004834	0.018766	-0.079809	-0.015176	0.688075
HouseAge	-0.119034	1.000000	-0.153277	-0.077747	-0.296244	0.013191	0.011173	-0.108197	0.105623
AveRooms	0.326895	-0.153277	1.000000	0.847621	-0.072213	-0.004852	0.106389	-0.027540	0.151948
AveBedrms	-0.062040	-0.077747	0.847621	1.000000	-0.066197	-0.006181	0.069721	0.013344	-0.046701
Population	0.004834	-0.296244	-0.072213	-0.066197	1.000000	0.069863	-0.108785	0.099773	-0.024650
AveOccup	0.018766	0.013191	-0.004852	-0.006181	0.069863	1.000000	0.002366	0.002476	-0.023737
Latitude	-0.079809	0.011173	0.106389	0.069721	-0.108785	0.002366	1.000000	-0.924664	-0.144160
Longitude	-0.015176	-0.108197	-0.027540	0.013344	0.099773	0.002476	-0.924664	1.000000	-0.045967
MedianHouseValue	0.688075	0.105623	0.151948	-0.046701	-0.024650	-0.023737	-0.144160	-0.045967	1.000000

In [125]:
```
# Apartado D -->
```

La métrica que más importa para decidir la bondad de un modelo de regresión es el R-cuadrado (R2). Esta métrica indica cuánto de la variable objetivo son capaces de explicar las variables con las que cuenta el modelo.

Además, lo que importa no es tanto la R2 del conjunto de train sino la de test. Esto se debe a que la R2 de train siempre (o en muchos casos) se puede mejorar iterando o añadiendo más complejidad al modelo, pero éste puede acabar perdiendo importancia porque se ajuste muy bien a los datos de train pero pierda la generalidad, siendo inaplicable a un conjunto que no haya visto antes (Este fenómeno se conoce como "overfitting). La R2 de test, por el contrario, ofrece información sobre el desempeño del modelo sobre datos que no ha conocido antes para ser entrenado, que son los datos que podríamos asumir que seguirán ocurriendo. Por lo tanto, para medir la bondad de un modelo sobre datos desconocidos es más relevante la R2 de test.

Arriba podemos ver que la R2 del modelo de random forest es significativamente superior (alrededor de un 20%) al de la regresión lineal, por lo que podemos asumir que este modelo es mejor.

En otro orden de cosas, para responder a los puntos anteriores he cargado el dataset y me lo he descrito con la función 'dataset_struct' (para lo que he incluido unos descriptores de las estadísticas básicas de cada variable a través de la función 'describe', si cada variable contiene valores nulos con la función 'isna().sum()', un conteo de los valores de las diferentes clases para los casos en los que estemos haciendo una clasificación, y los tipos de datos de cada variable mediante el elemento 'dtypes').

Una vez vista la estructura del dataset, me lo he dividido en un conjunto de entrenamiento y de test, que sirven para lo que hemos visto previamente de probar el modelo con datos que no ha visto antes como proxy a la vida real.

Después, he creado los modelos, los he entrenado con los datos de train, he hecho las predicciones y, finalmente, me he sacado algunas métricas interesantes para evaluarlos. Este método es aplicable a ambos modelos, tanto al de regresión lineal como al de random forest.

Finalmente, para comprobar qué variables son más relevantes para el modelo he realizado una matriz de correlaciones con la variable target. Aquellas variables que tengan una correlación más alta (sea positiva o negativa, i.e. en valor absoluto) con la variable objetivo serán las que más expliquen la misma y, por lo tanto, las más relevantes.

Ejercicio 3 (4 puntos):

Consideremos el dataset que contiene **The Most Streamed Spotify Songs 2023** que se encuentra en el respositorio.

Información de las variables:

- track_name: Name of the song
- artist(s)_name: Name of the artist(s) of the song
- vartist_count: Number of artists contributing to the song
- released_year: Year when the song was released
- released_month: Month when the song was released
- release_day: Day of the month when the song was released
- in_spotify_playlists: Number of Spotify playlists the song is included in
- in_spotify_charts: Presence and rank of the song on Spotify charts
- streams: Total number of streams on Spotify
- in_apple_playlists: Number of Apple Music playlists the song is included in
- in_apple_charts: Presence and rank of the song on Apple Music charts
- in_deezer_playlists: Number of Deezer playlists the song is included in
- in_deezer_charts: Presence and rank of the song on Deezer charts
- in_shazam_charts: Presence and rank of the song on Shazam charts
- bpm: Beats per minute, a measure of song tempo
- key: Key of the song
- mode: Mode of the song (major or minor)
- danceability_%: Percentage indicating how suitable the song is for dancing
- valence_%: Positivity of the song's musical content
- energy_%: Perceived energy level of the song
- acousticness_%: Amount of acoustic sound in the song
- instrumentalness_%: Amount of instrumental content in the song
- liveness_%: Presence of live performance elements
- speechiness_%: Amount of spoken words in the song

Para las respuestas b, c, d, e, f y g es imperativo acompañarlas respuestas con una visualización.

a) Lee el fichero en formato dataframe, aplica la función del ejercicio 2.b, elimina NAs y convierte a integer si fuera necesario. **(0.25 puntos)**

b) ¿Cuántos artistas únicos hay? **(0.25 puntos)**

c) ¿Cuál es la distribución de reproducciones? **(0.5 puntos)**

d) ¿Existe una diferencia signitificativa en las reproducciones entre las canciones de un solo artista y las de más de uno? **(0.5 puntos)**

e) ¿Cuáles son las propiedades de una canción que mejor correlan con el número de reproducciones de una canción? **(0.5 puntos)**

f) ¿Cuáles son las variables que mejor predicen las canciones que están por encima el percentil 50? **(1 puntos)**

Nota: Crea una variable binaria (Hit/No Hit) en base a 3.c, crea una regresión logística y visualiza sus coeficientes.

g) Agrupa los 4 gráficos realizados en uno solo y haz una recomendación a un sello discográfico para producir un nuevo hit. **(1 puntos)**

In [126]:
```
# Apartado A
spotify = pd.read_csv('./spotify-2023.csv', encoding='iso-8859-1')
dataset_struct(spotify, 'reg')
print('-----------------------------------------------------------------')
spotify = spotify.dropna()
dataset_struct(spotify, 'reg')
spotify["streams"] = pd.to_numeric(spotify["streams"], errors='coerce')
spotify
```

```
       artist_count  released_year  released_month  released_day  \
count    953.000000     953.000000      953.000000    953.000000
mean       1.556139    2018.238195        6.033578     13.930745
std        0.893044      11.116218        3.566435      9.201949
min        1.000000    1930.000000        1.000000      1.000000
25%        1.000000    2020.000000        3.000000      6.000000
50%        1.000000    2022.000000        6.000000     13.000000
75%        2.000000    2022.000000        9.000000     22.000000
max        8.000000    2023.000000       12.000000     31.000000
```

```
       in_spotify_playlists  in_spotify_charts  in_apple_playlists  \
count            953.000000         953.000000          953.000000
mean            5200.124869          12.009444           67.812172
std             7897.608990          19.575992           86.441493
min               31.000000           0.000000            0.000000
25%              875.000000           0.000000           13.000000
50%             2224.000000           3.000000           34.000000
75%             5542.000000          16.000000           88.000000
max            52898.000000         147.000000          672.000000

       in_apple_charts  in_deezer_charts         bpm  danceability_%  \
count       953.000000        953.000000  953.000000       953.00000
mean         51.908709          2.666317  122.540399        66.96957
std          50.630241          6.035599   28.057802        14.63061
min           0.000000          0.000000   65.000000        23.00000
25%           7.000000          0.000000  100.000000        57.00000
50%          38.000000          0.000000  121.000000        69.00000
75%          87.000000          2.000000  140.000000        78.00000
max         275.000000         58.000000  206.000000        96.00000

        valence_%    energy_%  acousticness_%  instrumentalness_%  liveness_%  \
count  953.000000  953.000000      953.000000          953.000000  953.000000
mean    51.431270   64.279119       27.057712            1.581322   18.213012
std     23.480632   16.550526       25.996077            8.409800   13.711223
min      4.000000    9.000000        0.000000            0.000000    3.000000
25%     32.000000   53.000000        6.000000            0.000000   10.000000
50%     51.000000   66.000000       18.000000            0.000000   12.000000
75%     70.000000   77.000000       43.000000            0.000000   24.000000
max     97.000000   97.000000       97.000000           91.000000   97.000000

       speechiness_%
count     953.000000
mean       10.131165
std         9.912888
min         2.000000
```

```
25%          4.000000
50%          6.000000
75%         11.000000
max         64.000000
track_name                0
artist(s)_name            0
artist_count              0
released_year             0
released_month            0
released_day              0
in_spotify_playlists      0
in_spotify_charts         0
streams                   0
in_apple_playlists        0
in_apple_charts           0
in_deezer_playlists       0
in_deezer_charts          0
in_shazam_charts         50
bpm                       0
key                      95
mode                      0
danceability_%            0
valence_%                 0
energy_%                  0
acousticness_%            0
instrumentalness_%        0
liveness_%                0
speechiness_%             0
dtype: int64
track_name              object
artist(s)_name          object
artist_count             int64
released_year            int64
released_month           int64
released_day             int64
```

```
in_spotify_playlists     int64
in_spotify_charts        int64
streams                 object
in_apple_playlists       int64
in_apple_charts          int64
in_deezer_playlists     object
in_deezer_charts         int64
in_shazam_charts        object
bpm                      int64
key                     object
mode                    object
danceability_%           int64
valence_%                int64
energy_%                 int64
acousticness_%           int64
instrumentalness_%       int64
liveness_%               int64
speechiness_%            int64
dtype: object
```

```
       artist_count  released_year  released_month  released_day  \
count    817.000000     817.000000      817.000000    817.000000
mean       1.567931    2018.457772        6.018360     13.696450
std        0.876211      10.829267        3.572554      9.299663
min        1.000000    1930.000000        1.000000      1.000000
25%        1.000000    2021.000000        3.000000      5.000000
50%        1.000000    2022.000000        5.000000     13.000000
75%        2.000000    2022.000000        9.000000     22.000000
max        8.000000    2023.000000       12.000000     31.000000

       in_spotify_playlists  in_spotify_charts  in_apple_playlists  \
count            817.000000         817.000000          817.000000
mean            4849.898409          11.722154           60.161567
std             7741.126455          18.617668           74.923594
min               31.000000           0.000000            0.000000
```

```
25%              829.000000           0.000000           12.000000
50%             2040.000000           3.000000           32.000000
75%             4890.000000          16.000000           78.000000
max            52898.000000         147.000000          532.000000

       in_apple_charts  in_deezer_charts         bpm  danceability_%  \
count       817.000000        817.000000  817.000000      817.000000
mean         49.473684          2.451652  122.565483       67.391677
std          49.570455          5.397024   28.174803       14.688458
min           0.000000          0.000000   65.000000       23.000000
25%           6.000000          0.000000   99.000000       57.000000
50%          34.000000          0.000000  120.000000       70.000000
75%          84.000000          2.000000  141.000000       79.000000
max         275.000000         45.000000  206.000000       96.000000

        valence_%    energy_%  acousticness_%  instrumentalness_%  liveness_%  \
count  817.000000  817.000000      817.000000          817.000000  817.000000
mean    51.201958   64.362301       26.309670            1.676867   18.168911
std     23.620978   16.107587       25.470972            8.767328   13.541996
min      4.000000   14.000000        0.000000            0.000000    3.000000
25%     32.000000   53.000000        5.000000            0.000000   10.000000
50%     51.000000   66.000000       17.000000            0.000000   12.000000
75%     70.000000   76.000000       41.000000            0.000000   24.000000
max     97.000000   97.000000       97.000000           91.000000   97.000000

       speechiness_%
count     817.000000
mean       10.526316
std        10.219987
min         2.000000
25%         4.000000
50%         6.000000
75%        12.000000
max        64.000000
track_name             0
```

```
artist(s)_name          0
artist_count            0
released_year           0
released_month          0
released_day            0
in_spotify_playlists    0
in_spotify_charts       0
streams                 0
in_apple_playlists      0
in_apple_charts         0
in_deezer_playlists     0
in_deezer_charts        0
in_shazam_charts        0
bpm                     0
key                     0
mode                    0
danceability_%          0
valence_%               0
energy_%                0
acousticness_%          0
instrumentalness_%      0
liveness_%              0
speechiness_%           0
dtype: int64
track_name             object
artist(s)_name         object
artist_count            int64
released_year           int64
released_month          int64
released_day            int64
in_spotify_playlists    int64
in_spotify_charts       int64
streams                object
in_apple_playlists      int64
in_apple_charts         int64
```

```
in_deezer_playlists     object
in_deezer_charts         int64
in_shazam_charts        object
bpm                      int64
key                     object
mode                    object
danceability_%           int64
valence_%                int64
energy_%                 int64
acousticness_%           int64
instrumentalness_%       int64
liveness_%               int64
speechiness_%            int64
dtype: object
```

Out[126]:

	track_name	artist(s)_name	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts	streams
0	Seven (feat. Latto) (Explicit Ver.)	Latto, Jung Kook	2	2023	7	14	553	147	141381703.0
1	LALA	Myke Towers	1	2023	3	23	1474	48	133716286.0
2	vampire	Olivia Rodrigo	1	2023	6	30	1397	113	140003974.0
3	Cruel Summer	Taylor Swift	1	2019	8	23	7858	100	800840817.0
4	WHERE SHE GOES	Bad Bunny	1	2023	5	18	3133	50	303236322.0
...	...	...	...	...	...	...	...	...	...
948	My Mind & Me	Selena Gomez	1	2022	11	3	953	0	91473363.0
949	Bigger Than The Whole Sky	Taylor Swift	1	2022	10	21	1180	0	121871870.0
950	A Veces (feat. Feid)	Feid, Paulo Londra	2	2022	11	3	573	0	73513683.0
951	En La De Ella	Feid, Sech, Jhayco	3	2022	10	20	1320	0	133895612.0
952	Alone	Burna Boy	1	2022	11	4	782	2	96007391.0

817 rows × 24 columns

In [127]:
```
# Apartado B
artistas_unicos = set(spotify["artist(s)_name"])
print(artistas_unicos)
print(f'Hay {len(artistas_unicos)} artistas únicos')
```

{'Skrillex, Flowdan, Fred again..', 'Arijit Singh, Vishal Dadlani, Sukriti Kakar, Vishal-Shekhar, Shekh
ar Ravjiani, Kumaar', 'J Balvin, Maria Becerra', 'PnB Rock', 'Karol G, Ovy On The Drums', 'Vishal-Shekh
ar, Shilpa Rao, Caralisa Monteiro, Kumaar, Vishal Dadlani, Shekhar Ravjiani', 'Sebastian Yatra, Manuel
Turizo, Beï¿½ï', 'Matuï¿½ï¿½, Wiu, ', 'Doja Cat, The Weeknd', 'Chase Atlantic', 'Justin Quiles, Lenny T
avï¿½ï¿½rez, BL', 'Meghan Trainor', 'Kendrick Lamar, Taylour Paige', 'Lord Huron', 'Gwen Stefani, Blake
Shelton', 'L7nnon, DJ Biel do Furduncinho, Bianca', 'Carin Leon, Grupo Frontera', 'Kali Uchis', 'Calvin
Harris, Halsey, Pharrell Williams, Justin Timberlake', 'Tiï¿½ï¿½sto, Kar', 'Mahalini', 'Nengo Flow, Anu
el Aa, Chris Jedi, Chencho Corleone', 'Rauw Alejandro, Bizarrap', 'TV Girl', 'Kendrick Lamar, Beth Gibb
ons', 'Michael Bublï¿', 'Lizzy McAlpine', 'Miguel', 'Mr.Kitty', 'Fran C, Polima WestCoast, Nickoog Clk,
Pablito Pesadilla', 'PinkPantheress, Ice Spice', 'Quevedo, Jhayco', 'Jessi', 'ThxSoMch', 'Tyler, The Cr
eator, Kali Uchis', 'Coldplay, BTS', 'Chencho Corleone, Bad Bunny', 'Justin Bieber', 'Kendrick Lamar, J
ay Rock', 'Childish Gambino', 'Swae Lee, Lil Wayne, Offset, Metro Boomin', 'Prezioso, Gabry Ponte, LUM!
X', 'Yng Lvcas', 'Kendrick Lamar, Tanna Leone', 'Mae Stephens', 'Nicky Youre, Dazy', 'Stephanie Beatri
z, Diane Guerrero', 'Imagine Dragons, League of Legends, Arcane', 'The Weeknd, Lil Wayne', 'Arijit Sing
h, Sachin-Jigar, Amitabha Bhattacharya', 'Daddy Yankee', 'Leah Kate', 'Melody, Ana Castela, Dj Chris No
Beat', 'Justin Bieber, Don Toliver', 'Labrinth', 'Shubh', 'J Balvin, Bad Bunny', 'Beach Weather', 'Kord
hell', '(G)I-DLE', 'Lizzo', 'Polo G', 'Sabrina Carpenter', 'Dr. Dre, Snoop Dogg', 'YOASOBI', 'Kendrick
Lamar, Blxst, Amanda Reifer', 'Danny Ocean', 'Sech, Mora', 'AnnenMayKantereit, Giant Rooks', 'Cigarette
s After Sex', 'Muni Long', 'JISOO', 'Tainy, Bad Bunny', 'Suki Waterhouse', 'Jack Harlow', 'Yandel, Fei
d', 'Grupo Marca Registrada, Grupo Frontera', 'P!nk', 'Karol G', 'Lil Yachty', 'Central Cee', 'a-ha', '
Alec Benjamin', 'Bellakath', 'Wisin & Yandel, ROSALï¿½', 'Dua Lipa, DaBaby', 'sped up 8282', 'Jaymes Yo
ung', 'Tini', 'Rich The Kid, Matuï¿', 'Camila Cabello, Ed Sheeran', 'Halsey, BTS', 'TAEYANG, Lisa', 'Po
st Malone, Doja Cat', 'RM', 'BYOR, Imanbek', 'Hotel Ugly', 'Panic! At The Disco', 'Luï¿½ï¿½sa Sonza, MC
Frog, Dj Gabriel do Borel, Davi K', 'Chanel', 'Bruno Mars', 'Sam Ryder', 'SZA, Travis Scott', 'Lil Dur
k, Morgan Wallen', 'Christian Nodal', 'Marshmello, Manuel Turizo', 'Hozier', 'teto', 'Kendrick Lamar, S
ampha', 'Ugly Dray, Tesla Jnr', 'Bad Bunny, Jhay Cortez', 'Calvin Harris, Ellie Goulding', 'Drake, Futu
re, Young Thug', 'Tears For Fears', 'Abhijay Sharma, Riar Saab', 'THE ANXIETY, Willow, Tyler Cole', 'De
an Martin', 'Alvaro Diaz, Rauw Alejandro', 'The Rare Occasions', 'Mï¿½ï¿½ne', 'Residente, Bizarrap', 'T
yga, Doja Cat', 'Morgan Wallen', 'King', 'Twisted, Oliver Tree', 'Nile Rodgers, LE SSERAFIM', 'The Rone
ttes', 'Eminem, Dina Rae', 'Anitta', 'Sean Paul, Dua Lipa', 'Lauren Spencer Smith, Lauren Spencer Smit
h, Lauren Spencer Smith', 'INTERWORLD', 'Nayeon', 'Rauw Alejandro, ROSALï¿½', 'El Chachito, Junior H',
'Chino Pacas', 'Daddy Yankee, Bad Bunny', 'Camila Cabello, Willow', 'Loreen', 'The Weeknd, Future', 'Ka
nii, PR1ISVX', 'Bebe Rexha, David Guetta', 'Sidhu Moose Wala', 'Kate Bush', 'Sean Paul, Feid', 'Mc Viti
n Da Igrejinha, MC Tairon, DJ Win', 'Junior H, Peso Pluma', 'Charlie Puth, BTS, Jung Kook', 'Natanael C
ano, Peso Pluma', 'Drake', 'Dj LK da Escï¿½ï¿½cia, Tchakabum, mc jhenny, M', 'Plan B', 'Masked Wolf', '

Anuel Aa, Jhay Cortez', 'The Weeknd, Daft Punk', 'Vundabar', 'Mc Pedrinho, Pedro Sampaio', 'NF', 'Becky G, Peso Pluma', 'HA SUNG WOON, Jimin', 'Southstar', 'Tyler, The Creator', 'Kaifi Khalil', 'Tiï¿½ï¿', 'Arcangel, De La Ghetto, Justin Quiles, Lenny Tavï¿½ï¿½rez, Sech, Dalex, Dimelo Flow, Rich Music', 'The Weeknd, Lana Del Rey', 'Feid, Young Miko', 'J Balvin, Nio Garcia, Bad Bunny', 'SZA, Doja Cat', 'Shawn Mendes, Camila Cabello', 'Paloma Faith', 'Tate McRae', 'Jimin', 'SiM', 'Doechii', 'Emmy Meli', 'Jain', 'Mc Pedrinho, DJ 900', 'Aitana, zzoilo', 'The Neighbourhood', 'TWICE', 'Luciano', 'Yung Lean', 'Armani White', 'Karol G, Quevedo', 'Zach Bryan', 'Nicki Minaj', 'Doja Cat', 'Fujii Kaze', 'Billie Eilish', '50 Cent', 'Future, Lil Uzi Vert, Metro Boomin', 'IVE', 'Ray Dalton, Ryan Lewis, Macklemore', 'Taylor Swift', 'Em Beihold', 'Eminem', 'Adele', 'Fuerza Regida, Grupo Frontera', 'Tony Dize, Bad Bunny', 'Chris Brown, Rvssian, Rauw Alejandro', 'Mc Livinho, DJ Matt D', 'Luis R Conriquez, La Adictiva', 'Maria Becerra', 'Marshmello, Juice WRLD', 'NIKI', 'Drake, Future, Tems', 'sped up nightcore, ARIZONATEARS, Lil Uzi Vert', 'Tulus', 'Natanael Cano, Gabito Ballesteros, Peso Pluma', 'Joji', 'Burna Boy', 'Halsey, Suga', 'Kendrick Lamar, Sam Dew, Baby Keem', 'Kendrick Lamar, Beyoncï¿', 'Quevedo, La Pantera, Juseph, Cruz Cafunï¿½ï¿½, Bï¿½ï¿½jo, Abhir Hathi', 'Peso Pluma, Yng Lvcas', 'Quevedo', 'Surf Curse', 'Migrantes, LiL Cake, Nico Valdi', 'Mabel Matiz, Mert Demir', 'Melanie Martinez', 'Drake, Travis Scott', 'Metro Boomin, Coi Leray', 'Zï¿½ï¿½ Fe', 'J. Cole', 'Myke Towers', 'SEVENTEEN', 'Libianca', 'Eminem, Nate Dogg', 'Kendrick Lamar, Kodak Black', 'MC Xenon, Os Gemeos da Putaria', 'Nessa Barrett', 'NewJeans', 'Feid, Mora', 'Mac DeMarco', 'Taylor Swift, Ice Spice', 'Yahritza Y Su Esencia, Grupo Frontera', 'Rauw Alejandro', 'Jnr Choi', 'Ozuna, Tiago pzk', 'Tory Lanez', 'Cherish, ACRAZE', 'Israel & Rodolffo, Mari Fernandez', 'Anggi Marito', 'Taylor Swift, Lana Del Rey', 'XXXTENTACION', 'Maluma', 'Perry Como, The Fontane Sisters, Mitchell Ayres & His Orchestra', 'Jason Derulo', 'BLESSD, Peso Pluma', 'Karol G, Romeo Santos', 'Rï¿½ï¿½ma, Selena G', 'Em Beihold, Stephen Sanchez', 'Elley Duhï¿', 'Manuel Turizo', 'Lil Nas X', 'Zion & Lennox', 'Harry Styles', 'Dave, Central Cee', 'Ed Sheeran, J Balvin', 'Shawn Mendes', 'Luude, Colin Hay', 'DJ Escobar, MC MENOR SG, MC MENOR HR', 'Zï¿½ï¿½ Neto & Crist', 'Musical Youth', 'Dave', 'Xamï¿½ï¿½, Gustah, Neo B', 'Peso Pluma, Grupo Frontera', 'David Guetta, Ella Henderson, Becky Hill', 'Myke Towers, Quevedo', 'Maroon 5', 'Natanael Cano', 'Kelly Clarkson', 'Feid', 'Nicki Minaj, Ice Spice', 'Kanye West', 'Eminem, Dr. Dre', 'Anitta, Tini, Becky G', 'Selena Gomez', 'Arcangel, Bad Bunny', 'Stephen Sanchez', 'Chris Molitor', 'Ludwig Goransson, Foudeqush', 'Paulo Londra', 'James Hype, Miggy Dela Rosa', 'Peggy Gou', 'Jessica Darrow', 'Drake, DJ Khaled, Lil Baby', 'Rihanna', "Shakin' Stevens", 'Yung Gravy', 'Kendrick Lamar, Baby Keem', 'Dua Lipa', 'TOMORROW X TOGETHER', 'Pharrell Williams, Nile Rodgers, Daft Punk', 'LF System', 'The Kid Laroi', 'Fifty Fifty', 'MC Caverinha, KayBlack', 'Luciano, Aitch, Bï¿½', 'Jung Kook', 'Darlene Love', 'Sam Smith', 'Nicky Jam, Feid', 'YEAT', 'James Arthur', 'John Lennon, The Harlem Community Choir, The Plastic Ono Band, Yoko Ono', 'Bizarrap, Peso Pluma', 'BLACKPINK', 'John Legend, Metro Boomin', 'Kodak Black, NLE Choppa, Muni Long, JVKE, Jimin', 'Sleepy hallow, 347aidan', 'Kenshi Yonezu', 'Blackbear, BoyWithUke', 'The Chainsmokers, Halsey', 'Lost Frequencies, Calum Scott', 'Omar Apoll

o', 'GODZZ__-, Zakaria', 'Andy Williams', 'Pharrell Williams, Tyler, The Creator, 21 Savage', 'Simone M
endes', 'David Kushner', 'Robin Schulz, Oliver Tree', 'Karol G, Shakira', 'Seafret', 'Ed Sheeran', 'Bom
ba Estï¿½ï¿½reo, Bad B', 'Bad Bunny', 'Mainstreet, Chefin', 'The Killers', 'Bizarrap, Quevedo', 'Intens
e, AP Dhillon, Gurinder Gill', 'Kanye West, XXXTENTACION', 'Bizarrap, Tiago pzk', 'Kendrick Lamar, Ghos
tface Killah, Summer Walker', 'Post Malone, Swae Lee', 'J. Cole, Lil Durk', 'Keane', 'Marï¿½ï¿½lia Mend
onï¿½ï¿½a, George Henrique &', 'Ayparia, unxbected', 'Lit Killah, Maria Becerra, Tiago pzk, NICKI NICOL
E', 'Fuerza Regida, Chino Pacas', 'Jasiel Nuï¿½ï¿½ez, Peso P', 'The Weeknd, 21 Savage, Metro Boomin', '
Feid, Mora, Saiko, Quevedo', 'Luke Combs', 'Feid, Myke Towers, Sky Rompiendo', 'Nat King Cole', 'Rihann
a, Calvin Harris', 'Lil Uzi Vert', 'A$AP Rocky, Metro Boomin, Roisee', 'Bad Bunny, Eladio Carrion', 'Es
labon Armado, Peso Pluma', 'Conan Gray', 'Drake, WizKid, Kyla', 'Dua Lipa, Megan Thee Stallion', 'Kendr
ick Lamar', 'Rex Orange County', 'Lauren Spencer Smith', 'Sam Smith, Kim Petras', "Jordan Fisher, Josh
Levi, Finneas O'Connell, 4*TOWN (From Disney and Pixarï¿½ï¿½ï¿½s Turning Red), Topher Ngo, Grayson Vil
l", 'Young Thug, Future, Gunna', 'Shakira, Rauw Alejandro', 'Travis Scott, Young Thug, Metro Boomin', '
Sachin-Jigar, Shadab Faridi, Altamash Faridi, Amitabh Bhattacharya, Varun Jain', 'Ruth B.', 'Duki, NICK
I NICOLE, Cris Mj, Standly, Stars Music Chile', 'Dr. Dre, 2Pac, Roger', 'Trueno, Tiago pzk', 'PinkPanth
eress', 'Latto, Jung Kook', 'Brray, Rauw Alejandro, Lyanno', 'V', 'Ryan Castro', 'SZA, Phoebe Bridger
s', 'SZA, Don Toliver', 'Marshmello, Jonas Brothers', 'Dove Cameron', 'Chuck Berry', 'Gabito Ballestero
s, Junior H, Peso Pluma', 'Julieta Venegas, Bad Bunny, Tainy', 'Morgan Wallen, Eric Church', 'Stromae',
'Travis Scott, 21 Savage, Metro Boomin', 'Dean Lewis', 'Mahmood, Blanco', 'MNEK, Jax Jones', 'Nirvana',
'Don Toliver, Future, Justin Bieber', 'Beyoncï¿', 'RM, Colde', 'Eden Muï¿½ï', 'j-hope', 'LE SSERAFIM',
'Paul McCartney', 'Arctic Monkeys', 'j-hope, J. Cole', 'd4vd', 'Yahritza Y Su Esencia', 'Taiu, Milo j',
'Treyce', 'Steve Lacy', 'James Blake, Metro Boomin', 'Kanye West, Lil Durk, Cardi B', 'Frank Sinatra',
'Tiï¿½ï¿½sto, Tate M', 'Sebastian Yatra', 'Drake, Travis Scott, 21 Savage', 'Israel & Rodolffo, Ana Cas
tela', 'Cartel De Santa, La Kelly', 'ENHYPEN', 'Mambo Kingz, DJ Luian, Anuel Aa', 'Nengo Flow, Bad Bunn
y', 'Avicii', 'Ana Castela, AgroPlay', 'Tini, Maria Becerra', 'Future, Chris Brown, Metro Boomin', 'NLE
Choppa', 'Marï¿½ï¿½lia Mendo', 'Josï¿½ï¿½ Felic', 'Bruno Mars, Anderson .Paak, Silk Sonic', 'Troye Siva
n', 'Future', 'Gunna', 'Shakira', 'Yuridia, Angela Aguilar', 'Benson Boone', 'Kanye West, Alicia Keys,
Fivio Foreign', 'Semicenk, Doï¿½ï¿½u ', 'Drake, Project Pat, 21 Savage', 'Ariana Grande, The Weeknd', '
Lady Gaga, Bradley Cooper', 'The Police', 'Grupo Marca Registrada, Junior H', 'Charli XCX, Jax Jones, J
oel Corry, Saweetie', 'Schï¿½ï¿½rze, DJ R', 'Ed Sheeran, Fireboy DML', 'C. Tangana', 'Bad Bunny, Tain
y', 'The Weeknd, Post Malone', 'The Weeknd', 'Raim Laode', 'De La Ghetto, Duki, Quevedo', 'Buscabulla,
Bad Bunny', 'Baby Tate', 'Arcangel, Bizarrap', 'The Weeknd, Gesaffelstein', 'Victor Cibrian', 'Halsey',
'SZA', 'Ozuna, Feid', 'Polima WestCoast, Pailita', 'Travis Scott, Metro Boomin', 'Chris Rea', 'Drake, 2
1 Savage', 'MC Ryan SP, Love Funk, Mc Paiva ZS', 'Carin Leon', 'Tiï¿½ï¿½sto, Ava', 'Bing Crosby, John S
cott Trotter & His Orchestra, Ken Darby Singers', 'SALES', 'Tini, L-Gante', 'NMIXX', 'Fuerza Regida, Na

```
tanael Cano', 'Lana Del Rey', 'Fuerza Regida', 'Bad Bunny, Grupo Frontera', 'Lady Gaga', 'Coi Leray', '
Shae Gill, Ali Sethi', 'PSY, Suga', 'Bizarrap, Villano Antillano', 'Bad Bunny, The Marï¿½ï', 'Justin Bi
eber, The Kid Laroi', 'Post Malone', 'Sech, Bad Bunny, Mora', 'Duki', 'Rels B', 'Lil Tjay', 'The Chains
mokers, Coldplay', 'Riton, Nightcrawlers, Mufasa & Hypeman, Dopamine', 'BIGBANG', 'The Walters', 'Aeros
mith', 'Chencho Corleone, Rauw Alejandro', 'Big One, FMK, Ke personajes', 'Jack Black', 'Kenia OS', 'Co
ldplay', 'Fuerza Regida, Peso Pluma', 'ROSALï¿½', 'Playboi Carti', 'Marï¿½ï¿½lia Mendonï¿½ï¿½a, Maiara
&', 'OneRepublic', 'Vance Joy', 'Gunna, Lil Baby', 'Arijit Singh, Sachin-Jigar', 'IU, Agust D', 'Olivia
Rodrigo', 'Bad Bunny, Rauw Alejandro', 'The Weeknd, Madonna, Playboi Carti', 'Cris Mj', 'Leo Santana',
'Imagine Dragons, League of Legends, JID, Arcane', 'New West', 'BTS', 'Calvin Harris, Dua Lipa, Young T
hug', 'Feid, Sech, Jhayco', 'Linkin Park', 'Steve Aoki, Tini, La Joaqui', 'JVKE', 'Dua Lipa, Elton Joh
n, Pnau', 'Kodak Black', 'Jin', 'Labrinth, Zendaya', 'Kaliii, Kaliii', 'WizKid, Toian, Metro Boomin, Do
n Toliver, Beam', 'Lana Del Rey, Taylor Swift', 'Billie Eilish, Khalid', 'David Guetta, Shakira, Black
Eyed Peas', 'Stray Kids', 'Ghost', 'Niall Horan', 'Sleepy hallow', 'Frank Ocean', 'De La Ghetto, Feid,
Polima WestCoast, Paloma Mami, Pailita', 'KALUSH', 'dennis, MC Kevin o Chris', 'Eminem, Dido', 'Junior
H, Eden Muï¿½ï', 'Lasso', 'The Weeknd, Tyler, The Creator', 'Willow', 'Veigh, Bvga Beatz, Supernova En
t, Prod Malax', 'Freddie Dredd', 'John Legend', 'Kevin Kaarl', 'Karol G, Becky G', 'Maldy, Karol G', 'N
icki Minaj, Lil Baby', 'Styrx, utku INC, Thezth', 'Offset, JID', 'Feid, Paulo Londra', '21 Savage, Gunn
a', 'Olga Merediz, Stephanie Beatriz, Encanto - Cast', 'Ovy On The Drums, Quevedo', 'Imagine Dragons',
"Guns N' Roses", 'Aventura, Bad Bunny', 'The Goo Goo Dolls', 'Charlie Puth', 'Shakira, Bizarrap', 'Mile
y Cyrus', 'Radiohead', 'Demi Lovato', 'Metallica', 'Edison Lighthouse', 'Juice WRLD', 'Feid, Alejo, Rob
i', 'Sofia Carson', 'David Guetta, Anne-Marie, Coi Leray', 'Oxlade', 'Anuel Aa, Myke Towers, Jhay Corte
z', 'Frank Sinatra, B. Swanson Quartet', 'girl in red', 'Lil Baby', 'Future, Metro Boomin, Don Tolive
r', 'Kali Uchis, Amaarae, Moliy', 'Lewis Capaldi', 'RAYE, 070 Shake', 'Agust D'}
Hay 571 artistas únicos
```

In [128]:
```
# Apartado C
fig = px.histogram(spotify, x="streams")
fig.show()
```

In [129]:
```
print(spotify.groupby("artist_count"))
```

```
<pandas.core.groupby.generic.DataFrameGroupBy object at 0x7f91df92fe20>
```

In [130]:
```
# Apartado E
spotify.corr()
# Las variables que mejor correlan son in_spotify_playlists y in_apple_playslists
```

Out[130]:

	artist_count	released_year	released_month	released_day	in_spotify_playlists	in_spotify_charts	streams	in_apple_playl
artist_count	1.000000	0.073564	0.033857	-0.014762	-0.085226	-0.008570	-0.109760	-0.017
released_year	0.073564	1.000000	0.076439	0.166377	-0.390729	0.068679	-0.242726	-0.201
released_month	0.033857	0.076439	1.000000	0.057784	-0.122407	-0.050186	-0.046041	-0.034
released_day	-0.014762	0.166377	0.057784	1.000000	-0.107616	0.016068	-0.025116	-0.011
in_spotify_playlists	-0.085226	-0.390729	-0.122407	-0.107616	1.000000	0.141343	0.780404	0.688
in_spotify_charts	-0.008570	0.068679	-0.050186	0.016068	0.141343	1.000000	0.214034	0.208
streams	-0.109760	-0.242726	-0.046041	-0.025116	0.780404	0.214034	1.000000	0.735
in_apple_playlists	-0.017024	-0.201474	-0.034029	-0.011916	0.688316	0.208202	0.735321	1.000
in_apple_charts	-0.075271	0.003479	-0.017813	0.007727	0.215676	0.556804	0.269137	0.364
in_deezer_charts	0.020585	0.095741	0.006942	0.055426	0.101283	0.566161	0.184329	0.326
bpm	-0.058844	-0.011570	-0.049400	-0.033394	-0.034483	0.028830	-0.025694	0.005
danceability_%	0.214078	0.215032	-0.054808	0.084244	-0.096981	0.051338	-0.093268	-0.012
valence_%	0.123650	-0.047643	-0.110355	0.062751	-0.029823	0.050040	-0.051014	0.041
energy_%	0.137530	0.078886	-0.086897	0.047318	0.035875	0.104963	-0.036499	0.039
acousticness_%	-0.094704	-0.133224	0.055046	0.000785	-0.064633	-0.072853	-0.005751	-0.070
instrumentalness_%	-0.061269	-0.023958	0.035481	0.023040	-0.024570	-0.005814	-0.033039	-0.054
liveness_%	0.034354	0.008489	0.001329	-0.011094	-0.051973	-0.026582	-0.056664	-0.064
speechiness_%	0.131486	0.129887	0.042127	-0.014602	-0.077610	-0.094102	-0.099968	-0.097

In [ ]: