```
from google.colab import files
uploaded = files.upload()
```

```
<IPython.core.display.HTML object>
```

```
Saving spotify.csv to spotify.csv
```

```
# Import packages
import pandas as pd
import datetime as dt
import seaborn as sns
import matplotlib.pyplot as plt
import io
# Import data
spotify = pd.read_csv(io.BytesIO(uploaded['spotify.csv']))
```

```
# 1.Explore data
```

```
spotify.head()
```

```
        Date  Shape of You  Despacito  Something Just Like This
HUMBLE.  \
0   1/6/2017      12287078        NaN                       NaN
NaN
1   1/7/2017      13190270        NaN                       NaN
NaN
2   1/8/2017      13099919        NaN                       NaN
NaN
3   1/9/2017      14506351        NaN                       NaN
NaN
4  1/10/2017      14275628        NaN                       NaN
NaN

   Unforgettable
0            NaN
1            NaN
2            NaN
3            NaN
4            NaN
```

```
spotify.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 6 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Date                      366 non-null    object
 1   Shape of You              366 non-null    int64
 2   Despacito                 359 non-null    float64
 3   Something Just Like This  319 non-null    float64
```

```
 4   HUMBLE.                   282 non-null    float64
 5   Unforgettable             275 non-null    float64
dtypes: float64(4), int64(1), object(1)
memory usage: 17.3+ KB
```

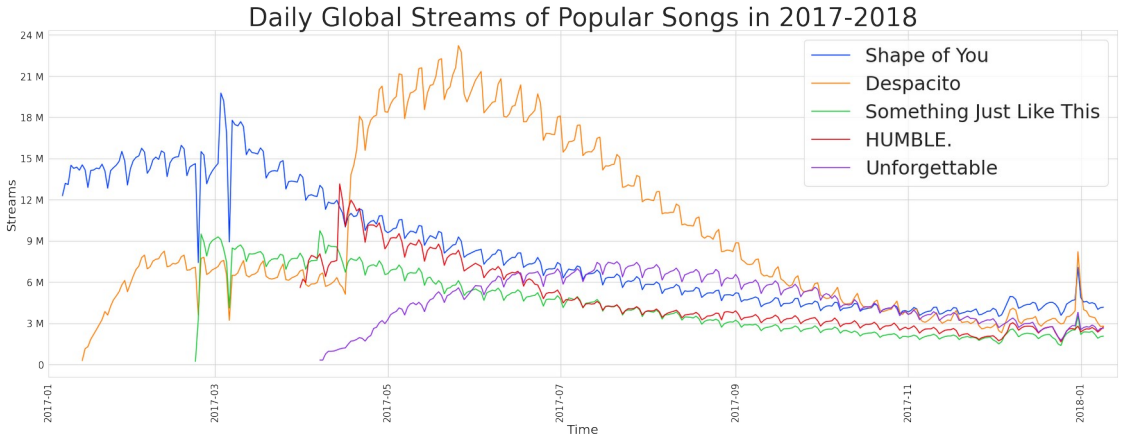
```
# 2.Wrangling data:

# Add a new columns, change data type of the Date column to datetime
spotify['Date_parsed'] = pd.to_datetime(spotify['Date'],
format="%m/%d/%Y")

# There is numbers of NaN values because they doesn't exist due to the
difference of released date.
# As wikipedia: released date as follows:
# Shape of You: 1/6/2017; Despacito: 1/12/2017; Something Just Like
This: 2/22/2017; HUMBLE.: 3/30/2017; Unforgettable: 4/7/2017
spotify.isna().sum()
```

```
Date                          0
Shape of You                  0
Despacito                     7
Something Just Like This     47
HUMBLE.                      84
Unforgettable                91
Date_parsed                   0
dtype: int64
```

```
#Replace NaN by 0
import numpy as np
spotify_0 = spotify.replace(np.nan,int(0))


# Reshape the data structure from wide to long to visualize daily
global streams of each song
spotify_long = pd.melt(spotify, id_vars='Date_parsed',
value_vars=['Shape of You', 'Despacito', 'Something Just Like This',
'HUMBLE.', 'Unforgettable'], var_name='Song', value_name='Streams')

# 3.Visualize data

# 3.1 Daily global streams of each song

# Set up

plt.figure(figsize=(30,10))
sns.set_style('whitegrid')
sns.set_palette('bright')

# Line chart showing daily global streams of each song
p1 = sns.lineplot(x='Date_parsed', y='Streams', hue='Song',
data=spotify_long)
```

```
# Format the plot
p1.set_xlabel("Time", fontsize = 20)
p1.set_ylabel("Streams", fontsize = 20)
p1.set_title("Daily Global Streams of Popular Songs in 2017-2018",
fontsize = 40)
plt.xticks(rotation=90, fontsize = 15)
plt.yticks(fontsize = 15)
plt.legend(fontsize = 30)

import matplotlib.ticker as ticker
from matplotlib.ticker import MultipleLocator
p1.yaxis.set_major_formatter(ticker.EngFormatter())
p1.yaxis.set_major_locator(MultipleLocator(3000000))
p1.set_xlim(pd.to_datetime('2017-01-01'), pd.to_datetime('2018-01-
14'))
```

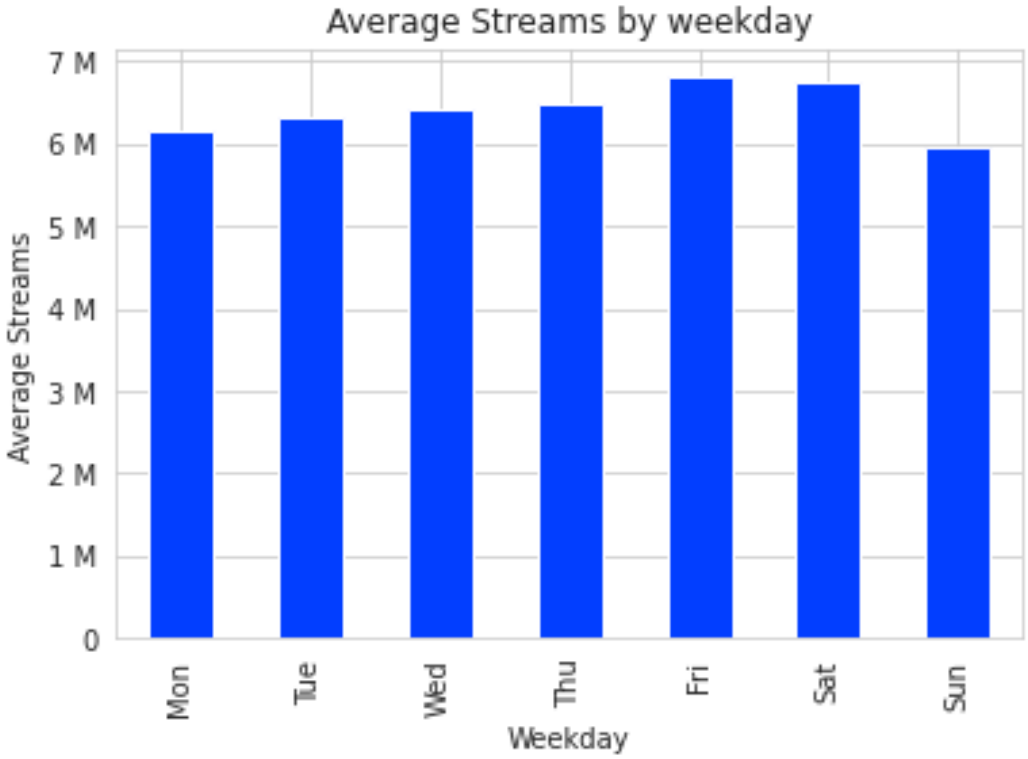
```
(736330.0, 736708.0)
```



These songs from release to cooldown tend to follow a pattern that peaks quickly and declines later, creating a long tail to the right. There is a small fluctuation that repeats regularly in a certain period (weekly), the higher it is, the larger the amplitude of the fluctuation tends to be. There may have been some outside influence that caused the sudden up and down fluctuations of the songs at some stage.

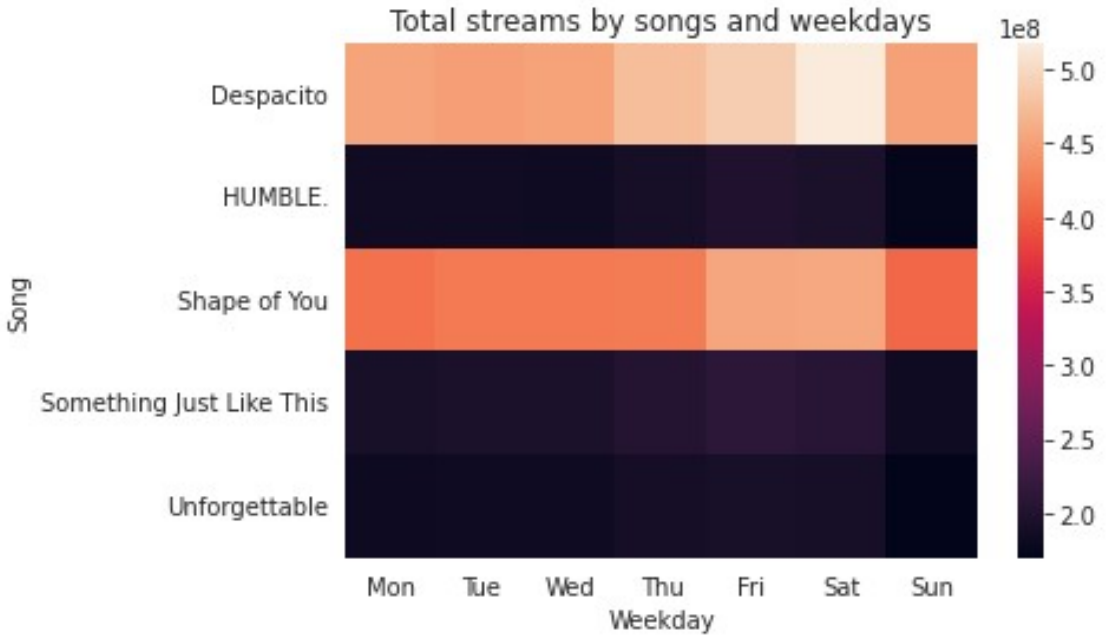
```
# 3.2 Streams by weekday

#Add weekday columns and sort values
spotify_long['weekday_no'] = spotify_long['Date_parsed'].dt.weekday
spotify_long_sorted = spotify_long.sort_values('weekday_no')
#Visualize total streams by weekdays
y = spotify_long_sorted.groupby('weekday_no')
['Streams'].mean().plot(kind='bar')
y.set_xticklabels(['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'])
y.set_xlabel('Weekday')
y.set_ylabel('Average Streams')
```

```
y.set_title('Average Streams by weekday')
y.yaxis.set_major_formatter(ticker.EngFormatter())
```




```
#Visualize total streams by weekdays by songs
z = sns.heatmap(pd.crosstab(spotify_long_sorted['Song'],
spotify_long_sorted['weekday_no'],
values=spotify_long_sorted['Streams'], aggfunc='sum'))
z.set(title='Total streams by songs and weekdays', xlabel='Weekday',
ylabel='Song')
z.set_xticklabels(['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'])
```

```
[Text(0.5, 0, 'Mon'),
 Text(1.5, 0, 'Tue'),
 Text(2.5, 0, 'Wed'),
 Text(3.5, 0, 'Thu'),
 Text(4.5, 0, 'Fri'),
 Text(5.5, 0, 'Sat'),
 Text(6.5, 0, 'Sun')]
```



As above charts, average streams gradually increase from Monday to Saturday, reach the top on Friday or Saturday and suddenly decrease on Sunday for all 5 songs

Use original spotify data to visualize the pattern in total streams

```
#Add weekday and month columns
spotify_0['day_of_week'] = spotify_0['Date_parsed'].dt.weekday
spotify_0['month'] = spotify_0['Date_parsed'].dt.to_period('M')

#Add total_streams column

spotify_0['total_streams'] = spotify_0['Shape of You'] +
spotify_0['Despacito'] + spotify_0['Something Just Like This']
+ spotify_0['HUMBLE.'] + spotify_0['Unforgettable']
spotify_0.head()
```

```
        Date  Shape of You  Despacito  Something Just Like This
HUMBLE.  \
0   1/6/2017      12287078        0.0                       0.0
0.0
1   1/7/2017      13190270        0.0                       0.0
0.0
2   1/8/2017      13099919        0.0                       0.0
0.0
3   1/9/2017      14506351        0.0                       0.0
0.0
4  1/10/2017      14275628        0.0                       0.0
0.0

   Unforgettable Date_parsed  day_of_week    month  total_streams
0            0.0  2017-01-06            4  2017-01     12287078.0
```

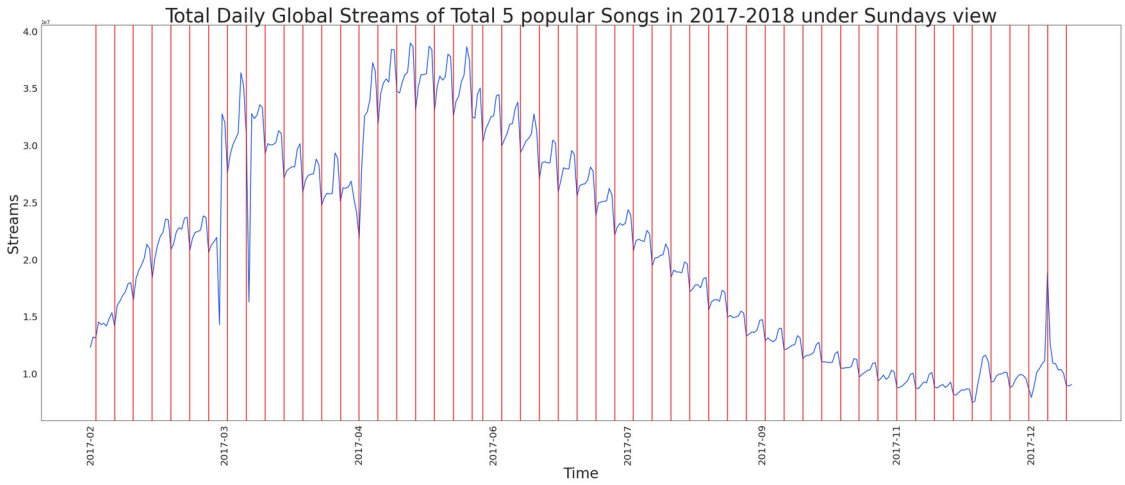
```
1            0.0  2017-01-07            5  2017-01     13190270.0
2            0.0  2017-01-08            6  2017-01     13099919.0
3            0.0  2017-01-09            0  2017-01     14506351.0
4            0.0  2017-01-10            1  2017-01     14275628.0
```

3.2.1 Sunday view

```
# Set xticks by month by index
spotify_0['month'].iloc[[0, 50, 100, 150, 200, 250, 300, 350]]
```

```
0      2017-01
50     2017-02
100    2017-04
150    2017-06
200    2017-07
250    2017-09
300    2017-11
350    2017-12
Name: month, dtype: period[M]
```

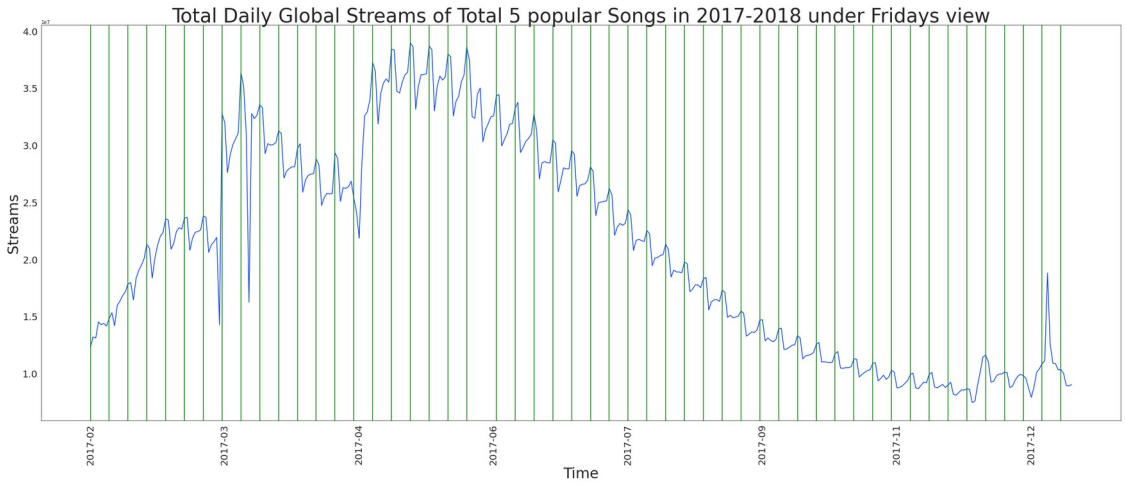
```
plt.figure(figsize=(40,15))
sns.set_style('white')
sns.set_palette('bright')

# Line chart showing daily global streams
sunday = sns.lineplot(x=spotify_0.index, y='total_streams',
data=spotify_0)
sunday.set_xticklabels(['2017-01', '2017-02', '2017-03', '2017-04',
'2017-06', '2017-07', '2017-09', '2017-11', '2017-12'])


# Format the plot
sunday.set_xlabel("Time", fontsize = 30)
sunday.set_ylabel("Streams", fontsize = 30)
sunday.set_title("Total Daily Global Streams of Total 5 popular Songs
in 2017-2018 under Sundays view", fontsize = 40)
plt.xticks(rotation=90, fontsize = 20)
plt.yticks(fontsize = 20)



#The red lines corresponds to the time point which is Sunday
sun = spotify_0[spotify_0['day_of_week'] == 6]

for xc in sun.index:
    plt.axvline(x=xc, color='red')
```



The red lines (Sundays) meet the lowest points in the week cycle.

3.2.1 Friday view

```
plt.figure(figsize=(40,15))
sns.set_style('white')
sns.set_palette('bright')

# Line chart showing daily global streams
friday = sns.lineplot(x=spotify_0.index, y='total_streams',
data=spotify_0)
friday.set_xticklabels(['2017-01', '2017-02', '2017-03', '2017-04',
'2017-06', '2017-07', '2017-09', '2017-11', '2017-12'])


# Format the plot
friday.set_xlabel("Time", fontsize = 30)
friday.set_ylabel("Streams", fontsize = 30)
friday.set_title("Total Daily Global Streams of Total 5 popular Songs
in 2017-2018 under Fridays view", fontsize = 40)
plt.xticks(rotation=90, fontsize = 20)
plt.yticks(fontsize = 20)



#The red lines corresponds to the time poins which is Fridays
fri = spotify_0[spotify_0['day_of_week'] == 4]

for xc in fri.index:
    plt.axvline(x=xc, color='green')
```



The green lines (fridays) meet the highest points in the week cycle.

The pattern tends to be repeated everyweek.