

Spotify Playlist



Team Members:

1. Pamu Arun Teja – 20201CAI0152
2. Aakash Kuragayala – 20201CAI0132
3. S. Ganesh – 20201CAI0096
4. A. Karthik sharma – 20201CAI0131

Dataset:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	track_name	artist(s)	n_artist_count	release_date	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso	release_date_iso
2	Seven (feat. Latto, Jung Kook)		2	2023	7	14	553	147	1.41E+08	43	263	45	10	826	125	B	Major	80	89	
3	LALA	Myke Towhe	1	2023	3	23	1474	48	1.34E+08	48	126	58	14	382	92	C#	Major	71	61	
4	vampire	Olivia Rodrigo	1	2023	6	30	1397	113	1.4E+08	94	207	91	14	949	138	F	Major	51	32	
5	Cruel Summer	Taylor Swift	1	2019	8	23	7858	100	8.01E+08	116	207	125	12	548	170	A	Major	55	58	
6	WHERE SHE BAD	Bunni	1	2023	5	18	3133	50	3.03E+08	84	133	87	15	425	144	A	Minor	65	23	
7	Sprinter	Dave, Cen	2	2023	6	1	2186	91	1.84E+08	67	213	88	17	946	141	C#	Major	92	66	
8	Ella Baila Sola	Eslabon Armado	2	2023	3	16	3090	50	7.26E+08	34	222	43	13	418	148	F	Minor	67	83	
9	Columbia	Quevedo	1	2023	7	7	714	43	58149378	25	89	30	13	194	100	F	Major	67	26	
10	fukumean	Gunna	1	2023	5	15	1096	83	95217315	60	210	48	11	953	130	C#	Minor	85	22	
11	La Bebe	- I Peso Pluma	2	2023	3	17	2953	44	5.54E+08	49	110	66	13	339	170	D	Minor	81	56	
12	un x100to	Bad Bunny	2	2023	4	17	2876	40	5.06E+08	41	205	54	12	251	83	F#	Minor	57	56	
13	Super Shy	NewJeans	1	2023	7	7	422	55	58255150	37	202	21	5	168	150	F	Minor	78	52	
14	Flowers	Miley Cyrus	1	2023	1	12	12211	115	1.32E+09	300	215	745	58	1,021	118		Major	71	65	
15	Daylight	David Kushner	1	2023	4	14	3528	98	3.88E+08	80	156	182	24	1,281	130	D	Minor	51	32	
16	As It Was	Harry Styles	1	2022	3	31	23575	130	2.51E+09	403	198	863	46		174	F#	Minor	52	66	
17	Kill Bill	SLAY	1	2022	12	8	8109	77	1.16E+09	183	162	161	12	187	89	G#	Major	64	43	
18	Cupid - Tw	Fifty Fifty	1	2023	2	24	2942	77	4.97E+08	91	212	78	6	0	120	B	Minor	78	76	
19	What Was	Billie Eilish	1	2023	7	13	873	104	30546883	80	227	95	24	1,173	78		Major	44	14	
20	Classy 101	Feid, Young	2	2023	3	31	2610	40	3.35E+08	43	100	54	14	187	100	B	Major	86	67	
21	Like Crazy	Jimin	1	2023	3	24	596	68	3.63E+08	8	104	23	2	29	120	G	Major	63	36	
22	LADY GAG	Gabito Bala	3	2023	6	22	332	26	86444842	11	163	10	4	0	140	F	Minor	65	87	
23	I Can See	'Taylor Swift	1	2023	7	7	516	38	52135248	73	119	42	1	150	123	F#	Major	69	82	
24	I Wanna Be	Arctic Monkeys	1	2013	1	1	12859	110	1.3E+09	24	98	582	2	73	135		Minor	48	44	

Python code to add the URL's:

```
import spotipy

from spotipy.oauth2 import SpotifyClientCredentials

import csv

import os

import json

class CustomCacheHandler(spotipy.cache_handler.CacheHandler):

    def __init__(self):

        pass

    def get_cached_token(self):

        return None
```

```

def save_token_to_cache(self, token_info):
    pass

def get_album_url(track_name, client_id, client_secret, progress_file):
    client_credentials_manager =
    SpotifyClientCredentials(client_id=client_id, client_secret=client_secret,
    cache_handler=CustomCacheHandler())

    sp =
    spotipy.Spotify(client_credentials_manager=client_credentials_manager)

    with open(progress_file, 'r') as file:
        try:
            progress_data = json.load(file)
        except json.decoder.JSONDecodeError:
            progress_data = {'processed_tracks': []}
            processed_tracks = progress_data.get('processed_tracks', [])

        for track_name in processed_tracks:
            print(f'Skipping already processed track: {track_name}')

    results = []
    for track_name in processed_tracks:
        results.append((track_name, get_album_url_for_track(sp,
        track_name)))

    return results, sp

def get_album_url_for_track(sp, track_name):
    results = sp.search(q=track_name, type='track', limit=1)

```

```

if results['tracks']['items']:
    track = results['tracks']['items'][0]
    album_id = track['album']['id']
    album = sp.album(album_id)
    return album['external_urls']['spotify']
else:
    return "Track not found"

```

```

def add_urls_to_csv(input_csv, client_id, client_secret,
output_csv='albums_with_urls.csv', progress_file='progress.json'):

```

```

    with open(input_csv, 'r') as csvfile:
        reader = csv.DictReader(csvfile)
        rows = list(reader)

```

```

    if os.path.exists(progress_file):
        with open(progress_file, 'r') as file:
            try:
                progress_data = json.load(file)
            except json.decoder.JSONDecodeError:
                progress_data = {'processed_tracks': []}

```

```

    else:
        progress_data = {'processed_tracks': []}

```

```

    with open(output_csv, 'a', newline="", encoding='utf-8') as csvfile:
        fieldnames = ['track_name', 'Album URL']
        writer = csv.DictWriter(csvfile, fieldnames=fieldnames)

```

```

        if csvfile.tell() == 0: # Only write header if file is empty

```

```

writer.writeheader()

for row in rows:
    track_name = row['track_name']
    if track_name in progress_data['processed_tracks']:
        print(f"Skipping already processed track: {track_name}")
        continue

    results, sp = get_album_url(track_name, client_id, client_secret,
progress_file)
    for track_name, album_url in results:
        row_data = {'track_name': track_name, 'Album URL': album_url}
        writer.writerow(row_data)
        progress_data['processed_tracks'].append(track_name)

    with open(progress_file, 'w') as file:
        json.dump(progress_data, file)

# Example usage
input_csv = 'C:/Disc_D/downloads/archive/spotify-2023.csv'
client_id = "0054a24f2fc643c69d56d020dd5f70be"
client_secret = "98b4a4b772ad4eca934a92ca60c246a0"
output_csv = 'C:/Disc_D/downloads/archive/albums_with_urls.csv'
progress_file = 'progress.json'

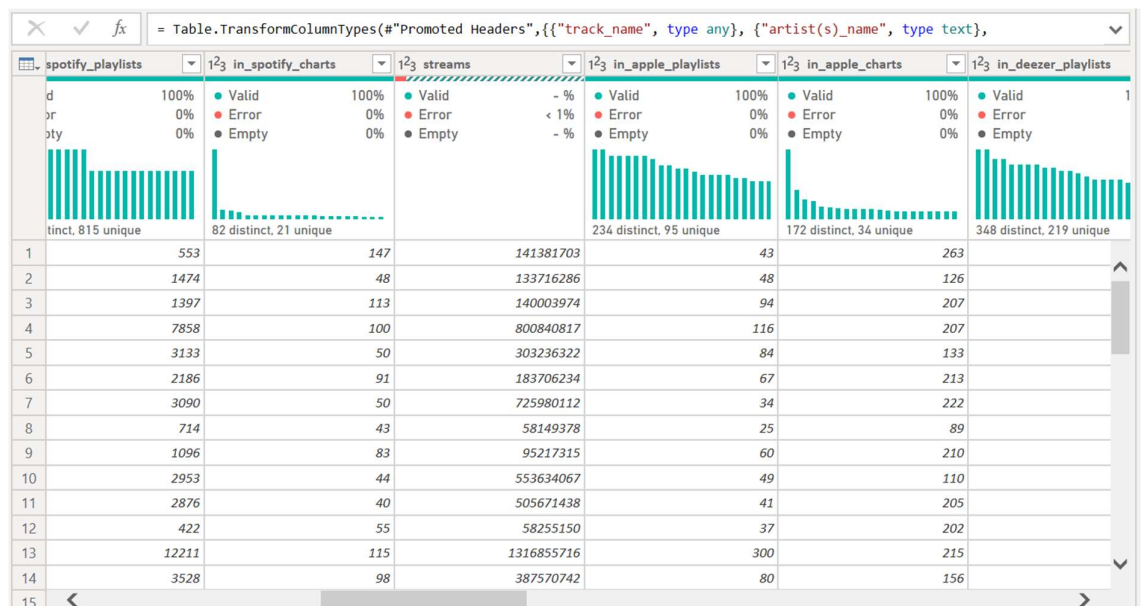
add_urls_to_csv(input_csv, client_id, client_secret, output_csv,
progress_file)

```

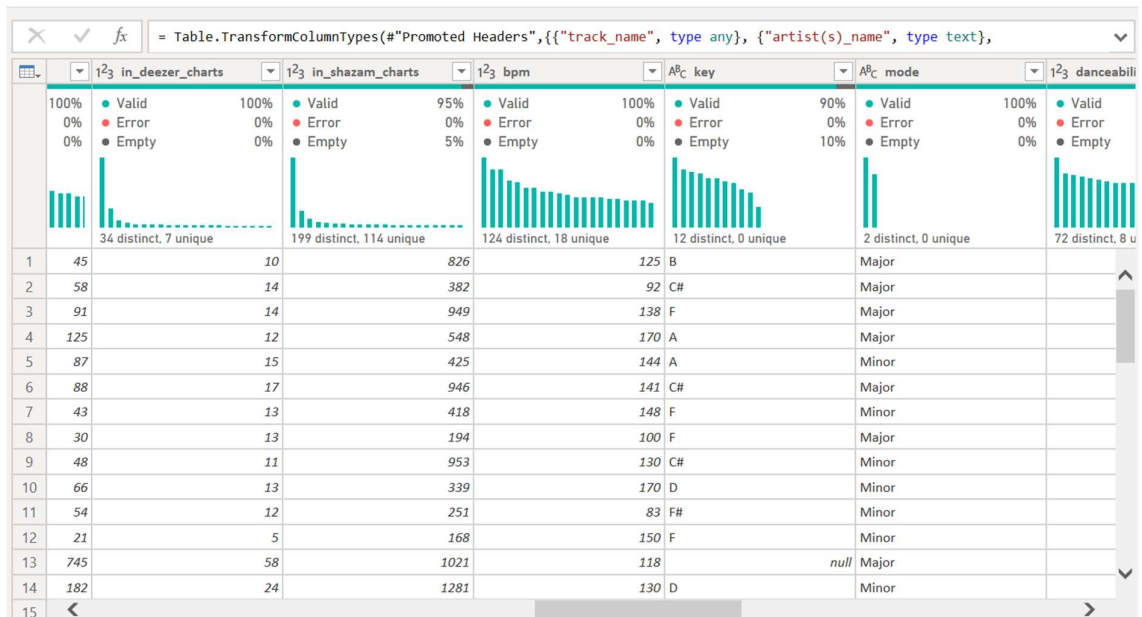
Y1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE														
																																cover_url													
1	track_name	artist	count	released	released	released	in_spotify	in_spotify	streams	in_apple	in_deezer	in_spotify	in_shazam	key																			mode	danceability	energy	s_acoustic	instrumentalness	speechiness	cover_url						
2	LALA	Mylva Tove	1	2023	9	23	1474	48	136	08	126	58	14	382	92	Major	71	61	74	7	0	10	4	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																					
3	vampire	Olivia Troy	1	2023	6	80	1397	113	146	08	94	207	91	14	349	138	F	Major	51	32	53	17	0	10	6	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
4	Heartbreaker	Olivia Troy	1	2023	6	80	1397	113	146	08	94	207	91	14	349	138	F	Major	51	32	53	17	0	10	6	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
5	WINDIE SH!t Budd	runo	1	2023	5	18	5133	50	186	08	84	133	87	15	425	144	M	Minor	65	23	80	14	63	11	6	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
6	Scrittori	Dave Kus	1	2023	6	1	2186	91	180	08	67	213	88	17	946	141	M	Minor	65	23	80	14	63	11	6	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
7	Beach	Beach	1	2023	3	16	1890	50	78	08	44	55	80	13	148	100	F	Major	67	26	71	76	48	0	10	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
8	Columba	Guendro	1	2023	7	7	714	43	58	07	25	89	30	13	184	100	F	Major	67	26	71	76	48	0	10	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
9	Kuvenen	Guendro	1	2023	5	15	1096	83	95	07	60	210	48	11	953	130	F	Major	65	22	62	12	12	0	28	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
10	Beach	Beach	1	2023	3	16	1890	50	78	08	44	55	80	13	148	100	F	Major	67	26	71	76	48	0	10	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
11	in x000s	Beach Budd	1	2023	4	17	2876	40	51	08	41	205	54	12	251	83	F	Major	57	56	72	13	0	27	5	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
12	Super Sky	Headlines	1	2023	7	7	122	58	08	07	22	58	08	07	22	58	F	Major	71	65	68	6	0	19	7	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
13	Flowers	Millie Cyrus	1	2023	1	12	1211	115	136	09	300	215	745	58	1021	118	F	Major	71	65	68	6	0	19	7	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
14	Driftin'	Dave Kus	1	2023	4	14	3528	98	98	08	80	156	182	14	1381	130	D	Minor	51	32	43	83	0	9	9	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
15	Beach	Beach	1	2023	3	16	1890	50	78	08	44	55	80	13	148	100	F	Major	67	26	71	76	48	0	10	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			
16	Kill Bill	SEZ	1	2022	12	8	8109	77	126	09	185	162	161	12	187	89	G	Major	64	43	73	5	17	16	4	https://scontent.cdninstagram.com/v/t51.2885-15/331468221_7306566288333648_8777405																			

Data cleaning in PowerBI:

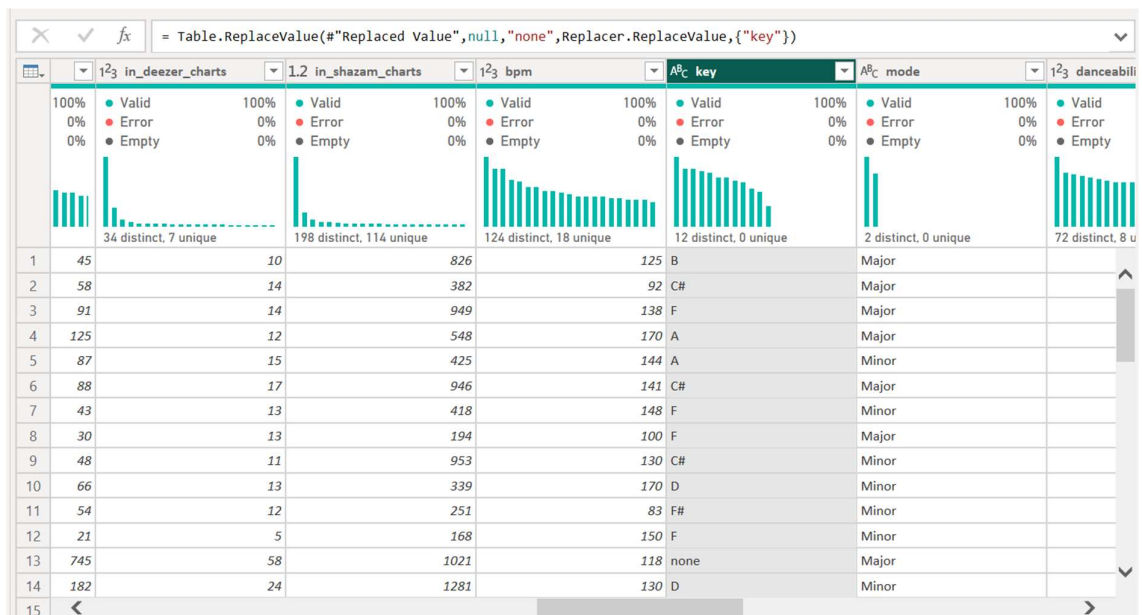
1. Streams:

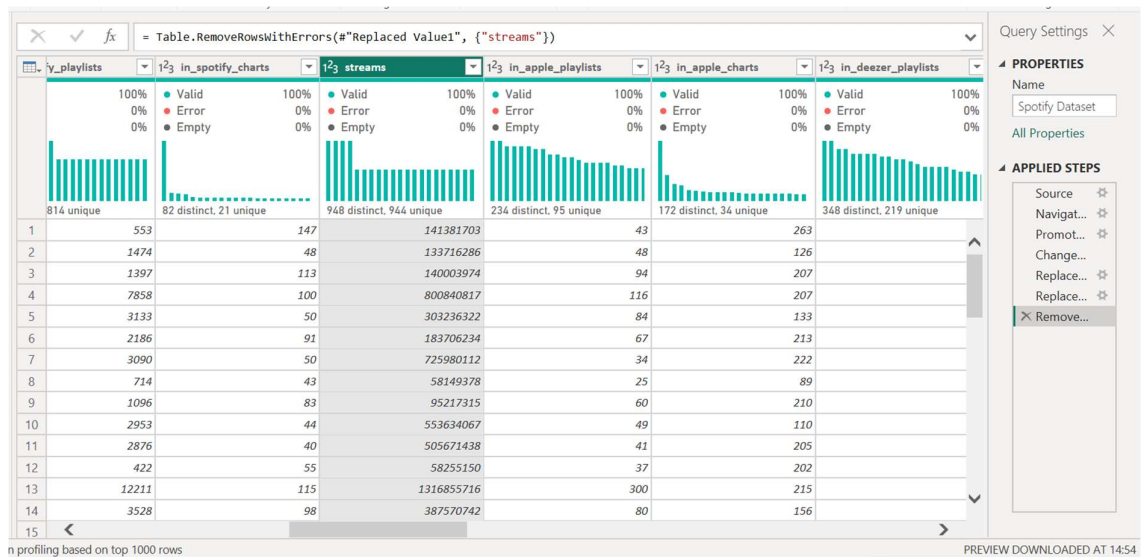


2. Empty rows in the shazam_charts, key:



After cleaning the errors:





➤ Creating new Measures for Data Visualization:

a. `_track`:

`_Track = COUNT('Spotify Dataset'[track_name])`

b. `_top song vs avg val`:

`_Top song vs avg val =
DIVIDE(
 [_Top Song streams] - [_Average Stream per year],
 [_Average Stream per year]
)`

c. `_top song vs avg`:

`_Top song vs AVG =
VAR x = [_Top song vs avg val] RETURN`

`IF(x > 0,
 FORMAT(x, "#.0%") & " " & UNICHAR (9650),
 FORMAT(x, "#.0%") & " " & UNICHAR (9660))`

d. `_Top song streams`:


```

_Top Song streams =
CALCULATE(
    SUM('Spotify Dataset'[streams]),
    'Spotify Dataset'[streams] = MAX('Spotify
Dataset'[streams])
)

```

e. _Percent val:

```

_Percent_val =
AVERAGE([energy_%])

```

f. _Max streams:

```

_Max streams = MAX('Spotify Dataset'[streams])

```

g. _Average stream per year:

```

_Average Stream per year =
CALCULATE(
    AVERAGE('Spotify Dataset'[streams]),
    ALLEXCEPT('Spotify Dataset', 'Calendar'[Year])
)

```

➤ Creating a new column called Date:

```

Date =
DATE([released_year],[released_month],[released_day])

```

➤ Creating a new table called Calendar:

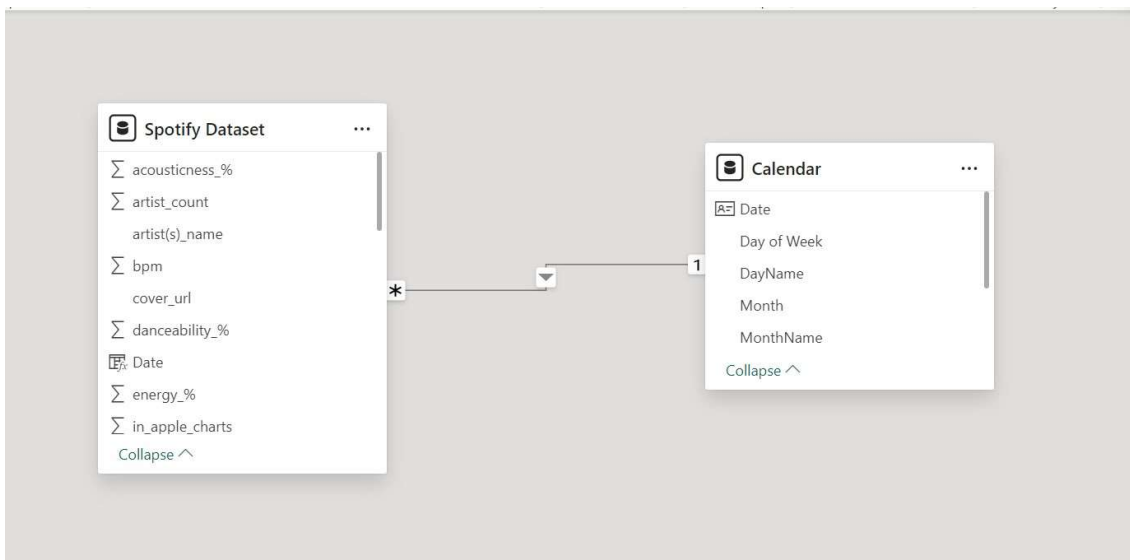
```

Calendar = ADDCOLUMNS(CALENDAR(MIN('Spotify
Dataset'[Date]), MAX('Spotify Dataset'[Date])),
    "Year", YEAR([Date]),
    "Quarter", QUARTER([Date]),
    "Quater (Q)", FORMAT([Date], "\QQ"),
    "Month", MONTH([Date]),
    "MonthName", FORMAT([Date], "mmm"),
    "Day of Week", WEEKDAY([Date]),
    "DayName", FORMAT([Date], "dddd"))

```

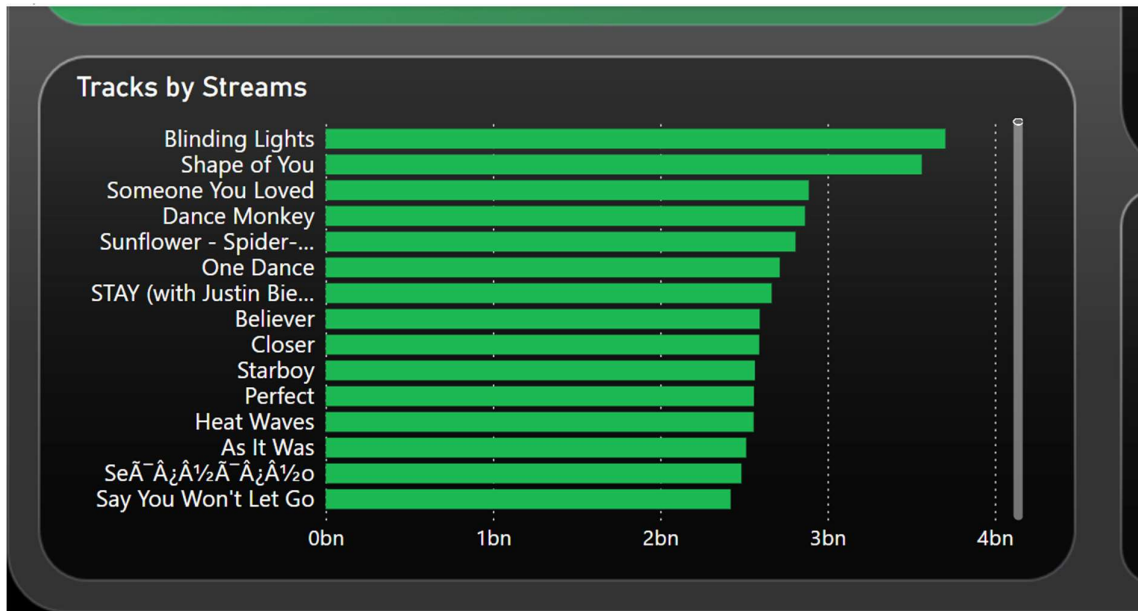
Date	Year	Quarter	Quarter (Q)	Month	MonthName	Day of Week	DayName
04-07-1930	1930	3	Q3	7	Jul	6	Friday
11-07-1930	1930	3	Q3	7	Jul	6	Friday
18-07-1930	1930	3	Q3	7	Jul	6	Friday
25-07-1930	1930	3	Q3	7	Jul	6	Friday
03-07-1931	1931	3	Q3	7	Jul	6	Friday
10-07-1931	1931	3	Q3	7	Jul	6	Friday
17-07-1931	1931	3	Q3	7	Jul	6	Friday
24-07-1931	1931	3	Q3	7	Jul	6	Friday
31-07-1931	1931	3	Q3	7	Jul	6	Friday
01-07-1932	1932	3	Q3	7	Jul	6	Friday
08-07-1932	1932	3	Q3	7	Jul	6	Friday
15-07-1932	1932	3	Q3	7	Jul	6	Friday
22-07-1932	1932	3	Q3	7	Jul	6	Friday
29-07-1932	1932	3	Q3	7	Jul	6	Friday
07-07-1933	1933	3	Q3	7	Jul	6	Friday
14-07-1933	1933	3	Q3	7	Jul	6	Friday
21-07-1933	1933	3	Q3	7	Jul	6	Friday
28-07-1933	1933	3	Q3	7	Jul	6	Friday
06-07-1934	1934	3	Q3	7	Jul	6	Friday
13-07-1934	1934	3	Q3	7	Jul	6	Friday
20-07-1934	1934	3	Q3	7	Jul	6	Friday

Model View of the Tables:

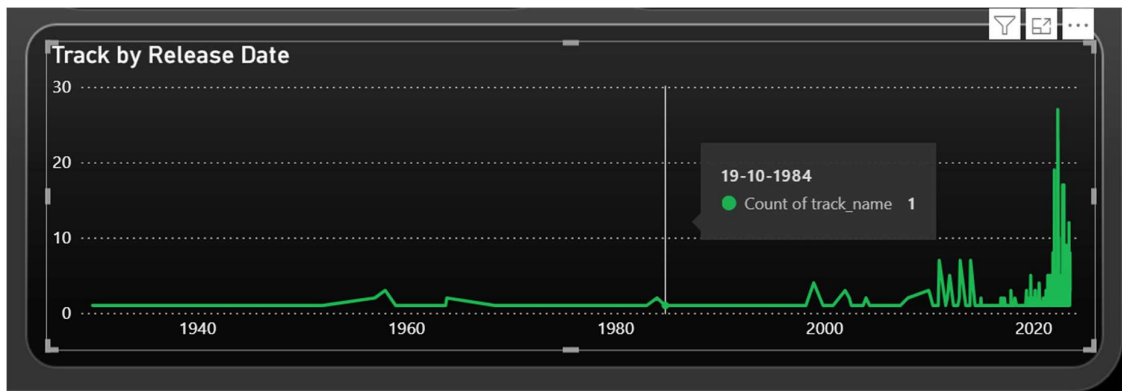


Visualization:

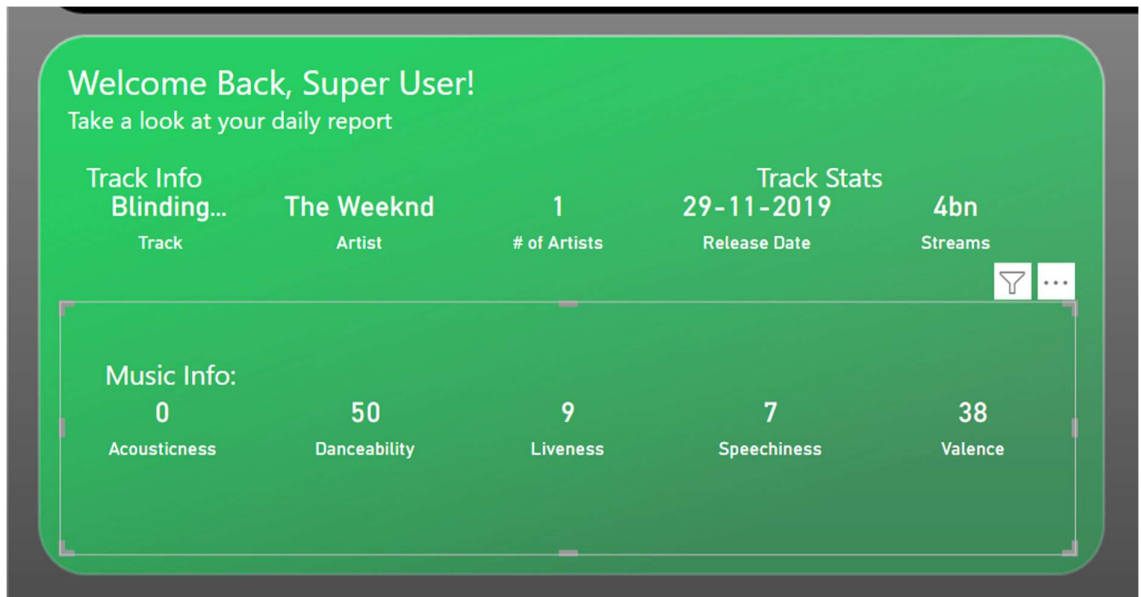
1. Visualization of Sum of streams and Track Names using bar chart:



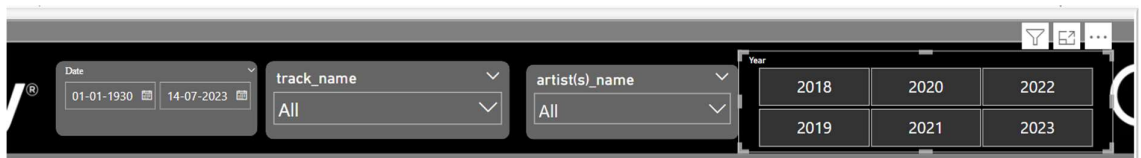
2. Visualization of track name and Date using line chart:



3. Visualization of Few columns that describe the music info and track info using cards:



4. Visualization of Date, track name, artist name and Year using Slicers:



5. Visualization of URL's using HTML Content:

Code:

```
_Image.html =
VAR x =
CALCULATE(
    MAX('Spotify Dataset'[cover_url]),
    'Spotify Dataset'[streams] = MAX('Spotify Dataset'[streams])
)
RETURN

"
<!DOCTYPE html>
<html lang='en'>
<head>
<meta charset='UTF-8'>
<title>Image Cropping</title>
<style>
.image-container {
    width: 458px; /* Width of the container */
    height: 140px; /* Height of the container */
    overflow: hidden; /* Hide parts of the image that don't fit */
    border-radius: 15px; /* Rounded corners */
    position: relative; /* Relative positioning for the child element */
}
```

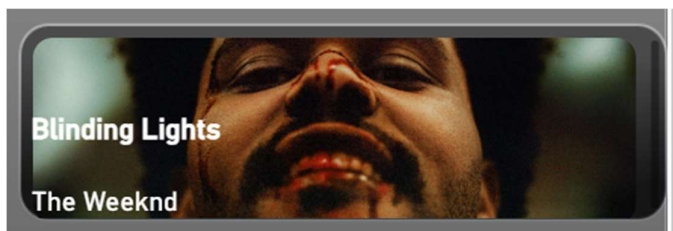
```

.image {
  object-fit: cover; /* Cover the entire container */
  object-position: center; /* Center the image */
  width: 100%; /* Full width */
  height: 100%; /* Full height */
}
</style>
</head>
<body>

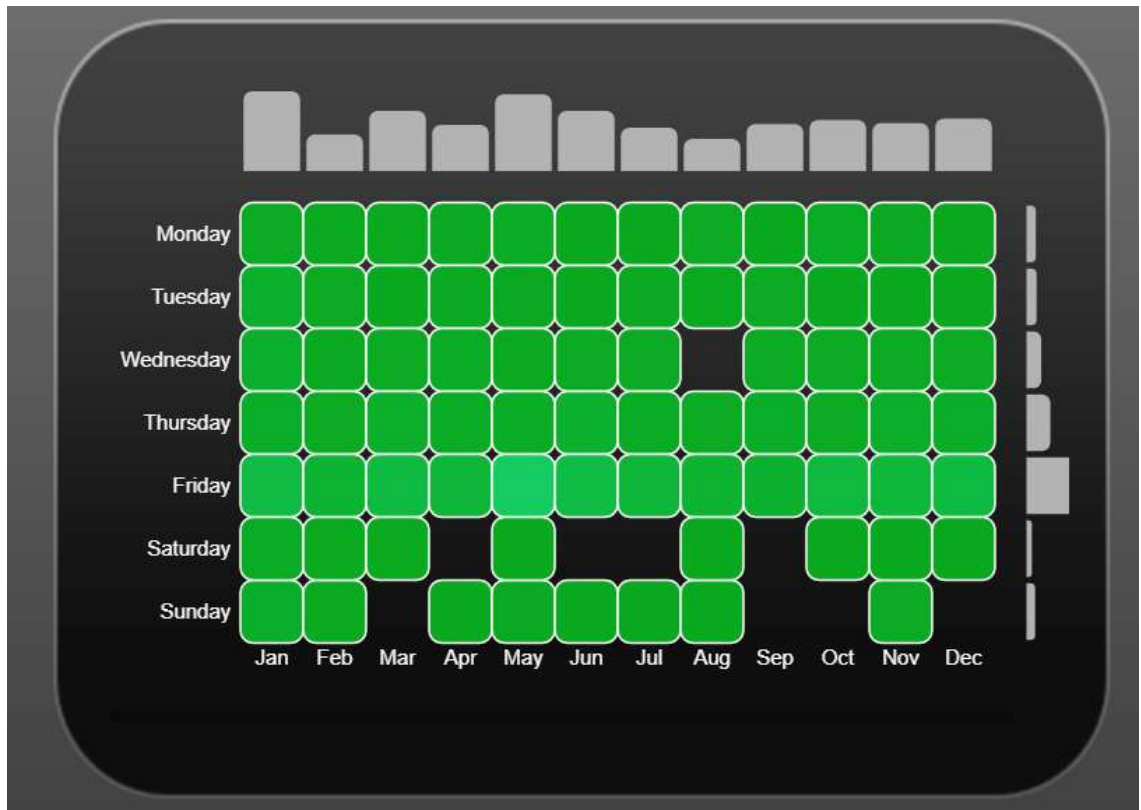
<div class='image-container'>
  <img src='&x&' alt='Album Cover' class='image'>
</div>

</body>
</html>
"

```



6. Visualization of release day of the week, Month of the name using Deneb Visualization:



7. Visualization of Energy of the song using Deneb Visualization tool:

Code:

```
{
  "$schema": "https://vega.github.io/schema/vega/v5.json",
  "width": 350,
  "height": 350,
  "padding": 15,
  "signals": [
    {
      "name": "textGradient",
      "update": "{gradient: 'linear', stops: [{offset: 0, color: '#036d18'},
      {offset: 1, color: '#1db954'}]}"
    },
    {
```

```
"name": "percent",
"update": "0",
"on": [
  {
    "events": {
      "type": "timer",
      "throttle": 0
    },
    "update": "round(data('dataset')[0]['_Percent_val'])"
  }
]
},
],
"data": [
  {"name": "dataset"},
  {
    "name": "back",
    "values": [],
    "transform": [
      {
        "type": "sequence",
        "start": 0,
        "stop": 100,
        "step": 1,
        "as": "val"
      },
      {
```

```
    "type": "formula",
    "expr": "1",
    "as": "t"
  },
  {
    "type": "pie",
    "field": "t",
    "startAngle": {"signal": "0"},
    "endAngle": {"signal": "2*PI"}
  }
],
{
  "name": "front",
  "values": [],
  "transform": [
    {
      "type": "sequence",
      "start": 0,
      "stop": {"signal": "percent"},
      "step": 1,
      "as": "val"
    },
    {
      "type": "formula",
      "expr": "1",
      "as": "t"
    }
  ]
}
```



```
    },
    {
      "type": "pie",
      "field": "t",
      "startAngle": {"signal": "0"},
      "endAngle": {
        "signal": "((2*PI)/100)*percent"
      }
    }
  ]
}
],
"scales": [
  {
    "name": "color",
    "type": "linear",
    "domain": {
      "data": "back",
      "field": "val"
    },
    "range": [
      "#036d18",
      "#1db954"
    ]
  }
],
"marks": [
```

```
{
  "type": "arc",
  "from": {"data": "back"},
  "encode": {
    "enter": {
      "fill": {"value": "#b3b3b3"},
      "x": {"signal": "width / 2"},
      "y": {"signal": "height / 2"}
    },
    "update": {
      "startAngle": {
        "field": "startAngle"
      },
      "endAngle": {
        "field": "endAngle"
      },
      "padAngle": {
        "signal": "0.015"
      },
      "innerRadius": {
        "signal": "(width / 2)-15"
      },
      "outerRadius": {
        "signal": "width / 2"
      }
    }
  }
}
```

```
},
{
  "type": "arc",
  "from": {"data": "front"},
  "encode": {
    "enter": {
      "fill": {
        "scale": "color",
        "field": "val"
      },
      "x": {"signal": "width / 2"},
      "y": {"signal": "height / 2"}
    },
    "update": {
      "startAngle": {
        "field": "startAngle"
      },
      "endAngle": {
        "field": "endAngle"
      },
      "padAngle": {
        "signal": "0.015"
      },
      "innerRadius": {
        "signal": "(width / 2)-15"
      },
      "outerRadius": {
```

```
        "signal": "width / 2"
    }
}
}
},
{
    "type": "arc",
    "data": [{"a": 1}],
    "encode": {
        "enter": {
            "fill": {"value": "#b3b3b3"},
            "x": {"signal": "width / 2"},
            "y": {"signal": "height / 2"}
        },
        "update": {
            "startAngle": {"signal": "0"},
            "endAngle": {
                "signal": "2*PI"
            },
            "innerRadius": {
                "signal": "(width / 2)-25"
            },
            "outerRadius": {
                "signal": "(width / 2)-20"
            }
        }
    }
}
```

```
},  
{  
  "type": "text",  
  "data": [{}],  
  "encode": {  
    "update": {  
      "text": {  
        "signal": "percent + '%" "  
      },  
      "align": {"value": "center"},  
      "fontWeight": {  
        "value": "bold"  
      },  
      "fill": {  
        "signal": "textGradient"  
      },  
      "x": {"signal": "width /2"},  
      "y": {"signal": "width /2"},  
      "dy": {"value": 10},  
      "fontSize": {"value": 70}  
    }  
  }  
},  
{  
  "type": "text",  
  "data": [{}],  
  "encode": {
```

```
"update": {  
  "text": {  
    "value": "Energy%"  
  },  
  "align": {"value": "center"},  
  "fontWeight": {  
    "value": "bold"  
  },  
  "fill": {"value": "#9092a1"},  
  "x": {"signal": "width /2"},  
  "y": {"signal": "width /2"},  
  "dy": {"value": 40},  
  "fontSize": {"value": 30}  
}  
}  
}  
]  
}
```

