



Spotify Tracks' Popularity

Visualization course project

Search for tracks in Spotify

You can collect 100 tracks at a time by entering a search query

Track, album, artist keywords

Past queries

#	Query	Average popularity	Common genre
1	roadtown	75,234,322	Blues
2	gangsta	100,233,235	rap
3	kazakh	45,234	folk

Search

ПОИСК ПО КЛЮЧЕВОМУ СЛОВУ

- Ввод слова в поисковую строку
- Отправка запроса в Spotify
- Возврат JSON файла с треками
- История запросов
- Расчет сводных метрик по каждому запросу

GET `{{base_content}}/search?type=track&limit=50&q=test&offset=50`

Params ▾

Query Params

	KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/>	type	track	
<input checked="" type="checkbox"/>	limit	50	
<input checked="" type="checkbox"/>	q	test	
<input checked="" type="checkbox"/>	offset	50	
	Key	Value	Description

Body ▾

200 OK 835 ms 235.82 KB

Pretty Raw Preview Visualize JSON ▾

```
1 {
2   "tracks": {
3     "href": "https://api.spotify.com/v1/search?query=test&type=track&offset=50&limit=50",
4     "items": [
5       {
6         "album": {
7           "album_type": "album",
8           "artists": [
9             {
10              "external_urls": {
11                "spotify": "https://open.spotify.com/artist/7COP1HCvVq1cIpZcLQYoNe"
12              },
13              "href": "https://api.spotify.com/v1/artists/7COP1HCvVq1cIpZcLQYoNe",
14              "id": "7COP1HCvVq1cIpZcLQYoNe",
15              "name": "Bedroom",
16              "type": "artist",
17              "uri": "spotify:artist:7COP1HCvVq1cIpZcLQYoNe"
18            }
19          ],
20          "available_markets": [
21            "AD",
22            "AE"
```

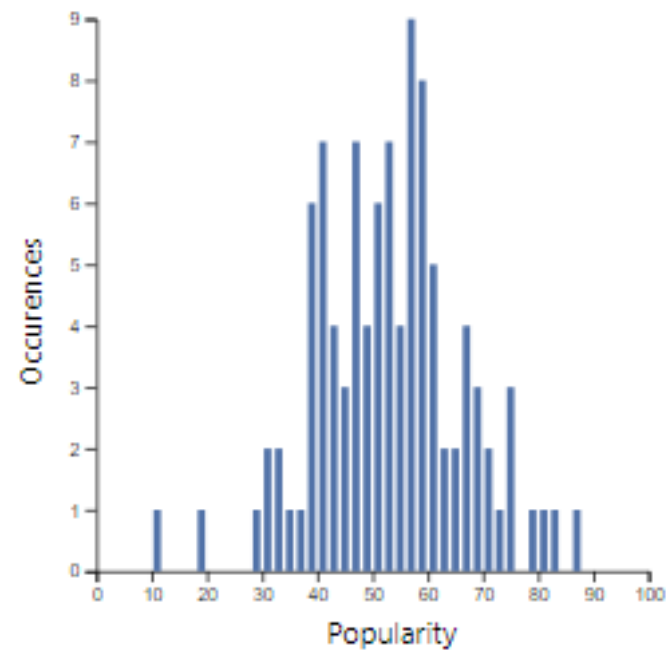


Choose a dataset

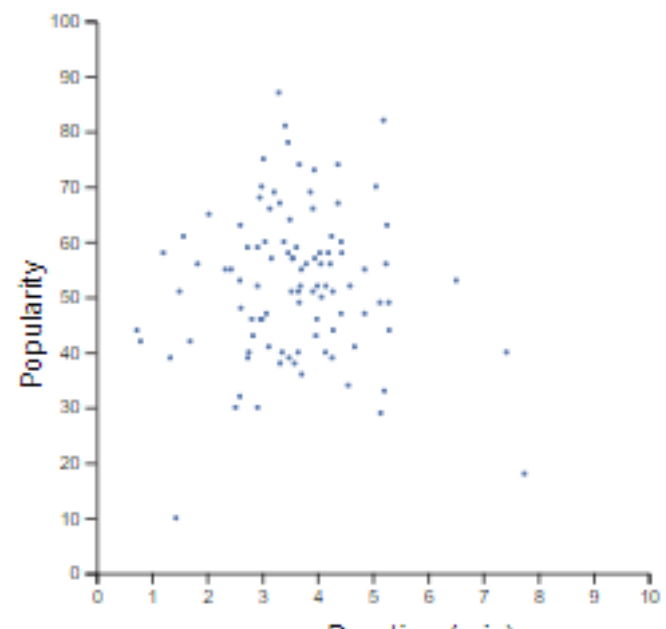
"roadtown"

Useful diagrams

Popularity distribution



Corellation b/w dur and pop

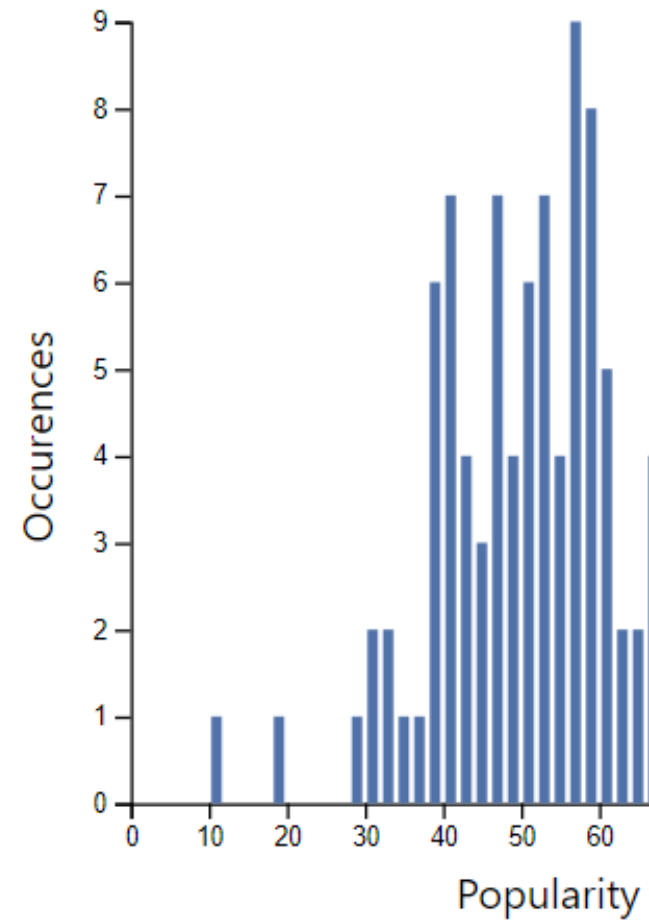


Keyword visualization

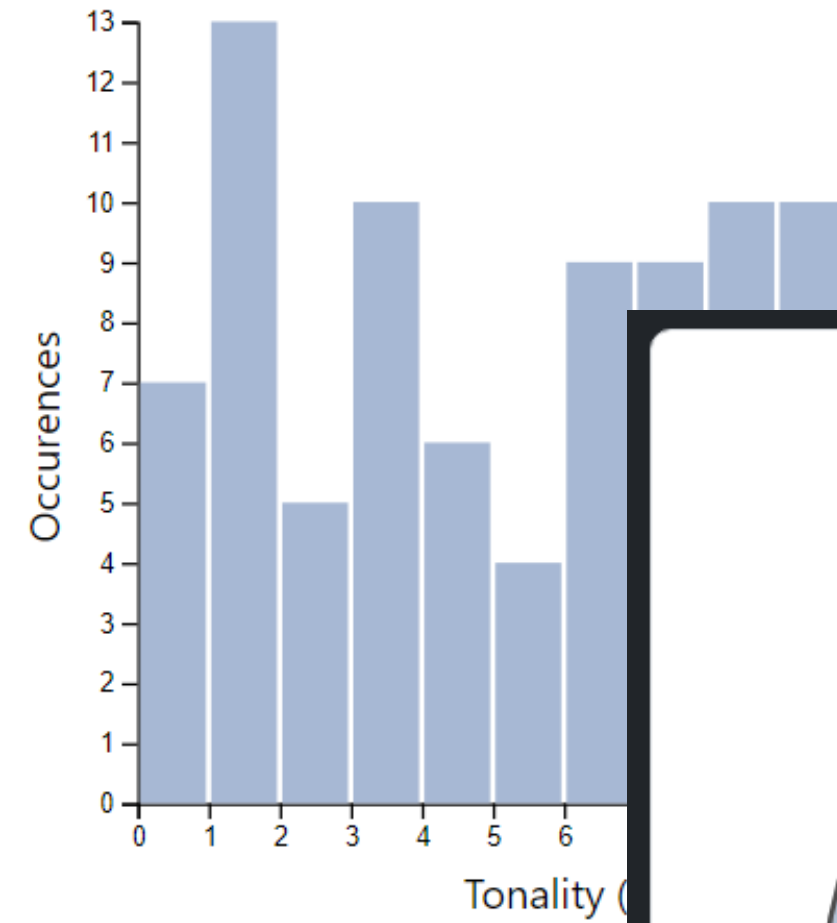
ВИЗУАЛИЗАЦИЯ ДАННЫХ ПО ОДНОМУ
ЗАПРОСУ

- Распределение популярности
- Кореляция популярности и длины трека
- Распределение тональностей
- Отношение количества треков с Explicit-контентом

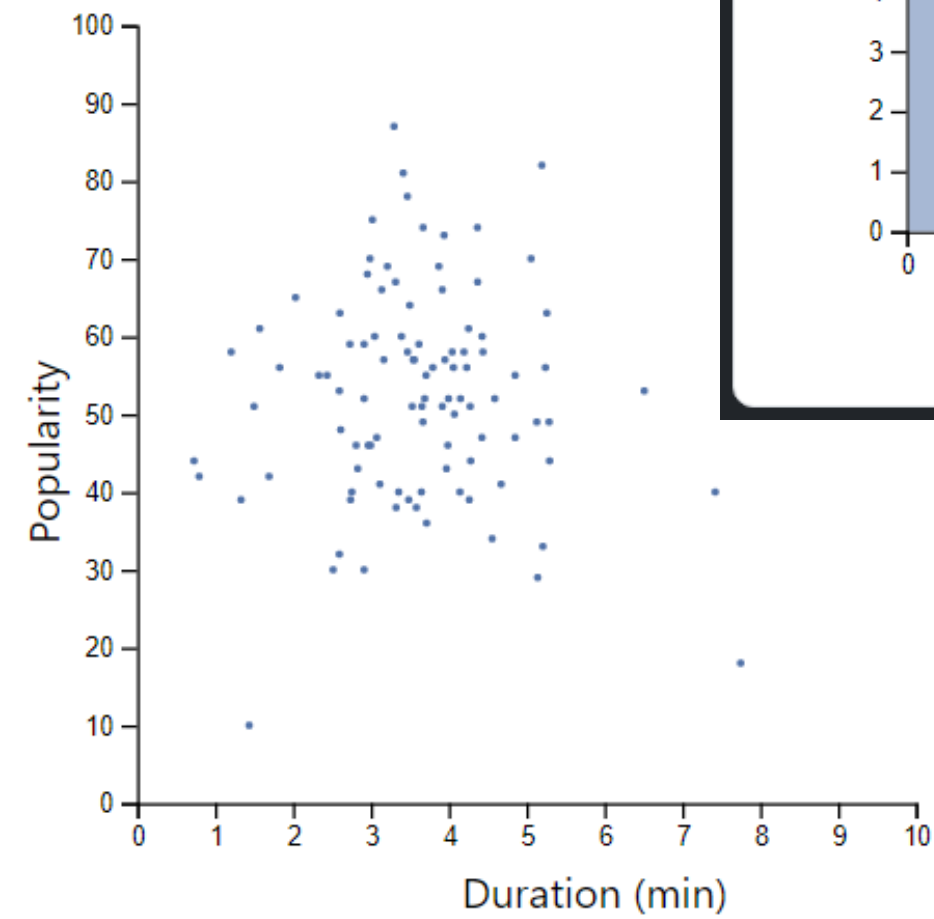
Popularity distribution



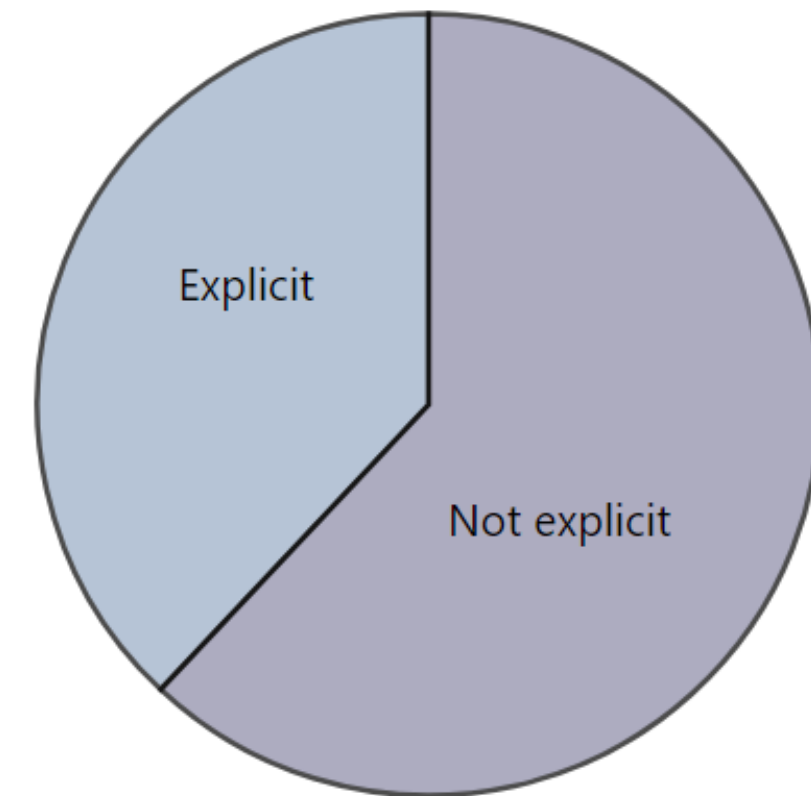
Tonality distribution



Corellation b/w dur a



Explicit content



Statistics

СВОДНАЯ ИНФОРМАЦИЯ

- Средняя популярность
- Тональность
- Средняя продолжительность
- Кол-во нот в лейтмотиве



Stats

Metrics	Value
Average popularity	75.5%
Common tonality	C#
Average duration	3.55 min
Notes in main riff	4.76



```
import json

with open(os.path.join('C:\\', 'Users', 'kozga', 'Desktop', 'tracks.json'), 'r') as tracks_json:
    data = json.load(tracks_json)
    res = []
    track_list = data["tracks"]["items"]
    for obj in track_list:
        dict = {}
        dict["artist"] = obj["artists"][0]["name"]
        dict["duration_min"] = obj["duration_ms"]/60/1000
        dict["explicit"] = obj["explicit"]
        dict["track_name"] = obj["name"]
        dict["popularity"] = obj["popularity"]
        dict["id"] = obj["id"]
        res.append(dict)
    json_file = json.dumps(res, indent=4)

with open(os.path.join('C:\\', 'Users', 'kozga', 'Desktop', 'result2.json'), 'w') as file:
    file.write(json_file)
```

```
In [6]: import os, json, random
tonalities = ['A', 'A#', 'B', 'C', 'C#', 'D', 'D#', 'E', 'F', 'F#', 'G', 'G#']

with open(os.path.join('C:\\', 'Users', 'kozga', 'Desktop', 'result.json'), 'r') as tracks_json:
    data = json.load(tracks_json)
    for track in data:
        # track["tonality"] = tonalities[random.randint(0, 11)]
        track["tonality"] = random.randint(0, 11)
    json_file = json.dumps(data, indent=4)

with open(os.path.join('C:\\', 'Users', 'kozga', 'Desktop', 'result3.json'), 'w') as file:
    file.write(json_file)
```

```
var svg_dist = d3.selectAll(".distribution"),
    margin = 100,
    width = svg_dist.attr("width") - margin,
    height = svg_dist.attr("height") - margin;

// get the data
d3.json("{% static 'json/result.json' %}", function (data) {
  // X axis: scale and draw:
  let x = d3.scaleLinear().domain([0, 100]).range([0, width]);
  svg_dist
    .append("g")
    .attr(
      "transform",
      "translate(" +
        margin / 2 +
        "," +
        (height + margin / 2) +
        ")"
    )
    .call(d3.axisBottom(x));
  svg_dist
    .append("text")
    .attr("x", (width + margin / 2) / 2)
    .attr("y", height + margin - 10)
    .text("Popularity");

  // set the parameters for the histogram
  let histogram = d3
    .histogram()
    .value(function (d) {
      return d.popularity;
    })
```

JS

d3.js для
построения
гистограмм,
scatterplots и pie
charts


```
from django.shortcuts import render

def parser(request):
    return render(request, 'parser/parser.html')

def visualization(request):
    return render(request, 'parser/visualization.html')

def overall(request):
    return render(request, 'parser/overall.html')
```

