

# Y.Music

## Konten

- [Pendahuluan](#)
- [Tahap 1. Ikhtisar Data](#)
  - [Kesimpulan](#)
- [Tahap 2. Pra-pemrosesan data](#)
  - [2.1 Gaya Penulisan Judul](#)
  - [2.2 Nilai-Nilai yang Hilang](#)
  - [2.3 Duplikat](#)
  - [2.4 Kesimpulan](#)
- [Tahap 3. Menguji Hipotesis](#)
  - [3.1 Hipotesis 1: aktivitas pengguna di dua kota](#)
  - [3.2 Hipotesis 2: preferensi musik pada hari Senin dan Jumat](#)
  - [3.3 Hipotesis 3: preferensi genre di kota Springfield dan Shelbyville](#)
- [Temuan](#)

## Pendahuluan

Setiap kali kita melakukan penelitian, kita perlu merumuskan hipotesis yang kemudian dapat kita uji. Terkadang kita menerima hipotesis ini; tetapi terkadang kita juga menolaknya. Untuk membuat keputusan yang tepat, sebuah bisnis harus dapat memahami apakah asumsi yang dibuatnya benar atau tidak.

### Tujuan:

Menguji tiga hipotesis:

1. Aktivitas pengguna berbeda-beda tergantung pada hari dan kotanya.
2. Pada senin pagi, penduduk Springfield dan Shelbyville mendengarkan genre yang berbeda. Hal ini juga berlaku untuk Jumat malam.
3. Pendengar di Springfield dan Shelbyville memiliki preferensi yang berbeda. Di Springfield, mereka lebih suka musik pop, sementara Shelbyville, musik rap memiliki lebih banyak penggemar.

## Tahapan

Data tentang perilaku pengguna disimpan dalam berkas

`/datasets/music_project_en.csv`. Tidak ada informasi tentang kualitas data, jadi perlu memeriksanya lebih dahulu sebelum menguji hipotesis.

Pertama, mengevaluasi kualitas data dan melihat apakah masalahnya signifikan. Kemudian, selama pra-pemrosesan data, akan mencoba memperhitungkan masalah yang paling serius.

Proyek ini akan terdiri dari tiga tahap:

- 1. Ikhtisar Data
- 2. Pra-pemrosesan Data
- 3. Menguji Hipotesis

[Kembali ke Daftar Isi](#)

## Tahap 1. Ikhtisar Data

In [58]:

import pandas as pd

In [59]:

df = pd.read\_csv('/datasets/music\_project\_en.csv')  
df.describe()

Out[59]:

	userID	Track	artist	genre	City	time	Day
count	65079	63736	57512	63881	65079	65079	65079
unique	41748	39666	37806	268	2	20392	3
top	A8AE9169	Brand	Kartvelli	pop	Springfield	08:14:07	Friday
freq	76	136	136	8850	45360	14	23149

In [60]:

df.head(10)

Out[60]:

	userID	Track	artist	genre	City	time	Day
0	FFB692EC	Kamigata To Boots	The Mass Missile	rock	Shelbyville	20:28:33	Wednesday
1	55204538	Delayed Because of Accident	Andreas Rönnerberg	rock	Springfield	14:07:09	Friday
2	20EC38	Funiculi funiculà	Mario Lanza	pop	Shelbyville	20:58:07	Wednesday
3	A3DD03C9	Dragons in the Sunset	Fire + Ice	folk	Shelbyville	08:37:09	Monday
4	E2DC1FAE	Soul People	Space Echo	dance	Springfield	08:34:34	Monday
5	842029A1	Chains	Obladaet	rusrap	Shelbyville	13:09:41	Friday
6	4CB90AA5	True	Roman Messer	dance	Springfield	13:00:07	Wednesday
7	F03E1C1F	Feeling This Way	Polina Griffith	dance	Springfield	20:47:49	Wednesday
8	8FA1D3BE	L'estate	Julia Dalia	ruspop	Springfield	09:17:40	Friday
9	E772D5C0	Pessimist	NaN	dance	Shelbyville	21:20:49	Wednesday

In [61]:

df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 65079 entries, 0 to 65078
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   userID      65079 non-null  object
1   Track        63736 non-null  object
2   artist       57512 non-null  object
3   genre        63881 non-null  object
4   City         65079 non-null  object
5   time         65079 non-null  object
6   Day          65079 non-null  object
dtypes: object(7)
memory usage: 3.5+ MB
```

Tabel ini berisi tujuh kolom. Semuanya menyimpan tipe data yang sama, yaitu: `object`.

Berdasarkan dokumentasi:

- `'userID'` — pengenalan pengguna
- `'Track'` — judul trek
- `'artist'` — nama artis
- `'genre'`
- `'City'` — kota tempat pengguna berada
- `'time'` — lama waktu lagu tersebut dimainkan
- `'Day'` — nama hari

Kita dapat melihat tiga masalah dengan gaya penulisan nama kolom:

1. Beberapa nama huruf besar, beberapa huruf kecil.
2. Ada penggunaan spasi pada beberapa nama.
3. Penggunaan spasi setelah nama kolom.

Jumlah nilai kolom berbeda. Ini berarti data mengandung nilai yang hilang.

## Kesimpulan

Setiap baris dalam tabel menyimpan data pada judul trek yang diputarkan. Beberapa kolom menggambarkan lagu itu sendiri: judul trek, artis, dan genre. Sisanya menyampaikan informasi tentang pengguna: kota asal mereka, waktu mereka memutar lagu.

Jelas bahwa data tersebut cukup untuk menguji hipotesis. Namun, ada nilai-nilai yang hilang.

Selanjutnya, kita perlu melakukan pra-pemrosesan data terlebih dahulu.

[Kembali ke Daftar Isi](#)

## Tahap 2. Pra-pemrosesan Data

Perbaiki format pada judul kolom dan atasi nilai yang hilang. Kemudian, periksa apakah ada duplikat dalam data.

## Gaya Penulisan Judul

Tampilkan judul kolom:

```
In [62]: df.columns

Out[62]: Index(['userID', 'Track', 'artist', 'genre', 'City', 'time', 'Day'], dtype='object')

In [63]: df = df.rename(columns={
    'userID': 'user_id',
    'Track': 'track',
    'artist': 'artist',
    'genre': 'genre',
    'City': 'city',
    'time': 'time',
    'Day': 'day'
})
df.columns

Out[63]: Index(['user_id', 'track', 'artist', 'genre', 'city', 'time', 'day'], dtype='object')
```

```
In [64]: df
```

Out[64]:

	user_id	track	artist	genre	city	time	day
0	FFB692EC	Kamigata To Boots	The Mass Missile	rock	Shelbyville	20:28:33	Wednesday
1	55204538	Delayed Because of Accident	Andreas Rönnerberg	rock	Springfield	14:07:09	Friday
2	20EC38	Funiculi funiculà	Mario Lanza	pop	Shelbyville	20:58:07	Wednesday
3	A3DD03C9	Dragons in the Sunset	Fire + Ice	folk	Shelbyville	08:37:09	Monday
4	E2DC1FAE	Soul People	Space Echo	dance	Springfield	08:34:34	Monday
...	...	...	...	...	...	...	...
65074	729CBB09	My Name	McLean	rnb	Springfield	13:32:28	Wednesday
65075	D08D4A55	Maybe One Day (feat. Black Spade)	Blu & Exile	hip	Shelbyville	10:00:00	Monday
65076	C5E3A0D5	Jalopiina	NaN	industrial	Springfield	20:09:26	Friday
65077	321D0506	Freight Train	Chas McDevitt	rock	Springfield	21:43:59	Friday
65078	3A64EF84	Tell Me Sweet Little Lies	Monica Lopez	country	Springfield	21:59:46	Friday

65079 rows × 7 columns

[Kembali ke Daftar Isi](#)

## Nilai-Nilai yang Hilang

In [65]: *# menghitung nilai yang hilang*  
`print(df.isna().sum())`

```
user_id      0
track       1343
artist      7567
genre       1198
city         0
time         0
day          0
dtype: int64
```

In [66]: *# Loop nama kolom dan ganti nilai yang hilang dengan 'unknown'*  
`columns_to_replace = ['track', 'artist', 'genre']`  
  
`for column in columns_to_replace:`  
 `df[column] = df[column].fillna('unknown')`  
  
`df.head()`

Out[66]:

	user_id	track	artist	genre	city	time	day
0	FFB692EC	Kamigata To Boots	The Mass Missile	rock	Shelbyville	20:28:33	Wednesday
1	55204538	Delayed Because of Accident	Andreas Rönberg	rock	Springfield	14:07:09	Friday
2	20EC38	Funiculi funiculà	Mario Lanza	pop	Shelbyville	20:58:07	Wednesday
3	A3DD03C9	Dragons in the Sunset	Fire + Ice	folk	Shelbyville	08:37:09	Monday
4	E2DC1FAE	Soul People	Space Echo	dance	Springfield	08:34:34	Monday

In [67]: *# menghitung nilai yang hilang*  
`print(df.isna().sum())`

```
user_id      0
track         0
artist        0
genre         0
city          0
time          0
day           0
dtype: int64
```

[Kembali ke Daftar Isi](#)

## Duplikat

In [68]: *# menghitung duplikat eksplisit*  
`print(df.duplicated().sum())`

```
3826
```

```
In [69]: # menghapus duplikat eksplisit  
df = df.drop_duplicates()
```

```
In [70]: # memeriksa duplikat  
print(df.duplicated().sum())
```

0

```
In [71]: # melihat nama genre yang unik  
sorted(df['genre'].unique())
```

```
Out[71]: ['acid',
          'acoustic',
          'action',
          'adult',
          'africa',
          'afrikaans',
          'alternative',
          'ambient',
          'americana',
          'animated',
          'anime',
          'arabesk',
          'arabic',
          'arena',
          'argentinetango',
          'art',
          'audiobook',
          'avantgarde',
          'axé',
          'baile',
          'balkan',
          'beats',
          'bigroom',
          'black',
          'bluegrass',
          'blues',
          'bollywood',
          'bossa',
          'brazilian',
          'breakbeat',
          'breaks',
          'broadway',
          'cantautori',
          'cantopop',
          'canzone',
          'caribbean',
          'caucasian',
          'celtic',
          'chamber',
          'children',
          'chill',
          'chinese',
          'choral',
          'christian',
          'christmas',
          'classical',
          'classicmetal',
          'club',
          'colombian',
          'comedy',
          'conjazz',
          'contemporary',
          'country',
          'cuban',
          'dance',
          'dancehall',
          'dancepop',
          'dark',
          'death',
          'deep',
```

'deutschrock',  
'deutschspr',  
'dirty',  
'disco',  
'dnb',  
'documentary',  
'downbeat',  
'downtempo',  
'drum',  
'dub',  
'dubstep',  
'eastern',  
'easy',  
'electronic',  
'electropop',  
'emo',  
'entehno',  
'epicmetal',  
'estrada',  
'ethnic',  
'eurofolk',  
'european',  
'experimental',  
'extrememetal',  
'fado',  
'film',  
'fitness',  
'flamenco',  
'folk',  
'folklore',  
'folkmetal',  
'folkrock',  
'folktronica',  
'forró',  
'frankreich',  
'französisch',  
'french',  
'funk',  
'future',  
'gangsta',  
'garage',  
'german',  
'ghazal',  
'gitarre',  
'glitch',  
'gospel',  
'gothic',  
'grime',  
'grunge',  
'gypsy',  
'handsup',  
'hard'n'heavy',  
'hardcore',  
'hardstyle',  
'hardtechno',  
'hip',  
'hip-hop',  
'hiphop',  
'historisch',  
'holiday',



'hop',  
'horror',  
'house',  
'idm',  
'independent',  
'indian',  
'indie',  
'indipop',  
'industrial',  
'inspirational',  
'instrumental',  
'international',  
'irish',  
'jam',  
'japanese',  
'jazz',  
'jewish',  
'jpop',  
'jungle',  
'k-pop',  
'karadeniz',  
'karaoke',  
'kayokyoku',  
'korean',  
'laiko',  
'latin',  
'latino',  
'leftfield',  
'local',  
'lounge',  
'loungeelectronic',  
'lovers',  
'malaysian',  
'mandopop',  
'marschmusik',  
'meditative',  
'mediterranean',  
'melodic',  
'metal',  
'metalcore',  
'mexican',  
'middle',  
'minimal',  
'miscellaneous',  
'modern',  
'mood',  
'mpb',  
'muslim',  
'native',  
'neoklassik',  
'neue',  
'new',  
'newage',  
'newwave',  
'nu',  
'nujazz',  
'numetal',  
'oceania',  
'old',  
'opera',

'orchestral',  
'other',  
'piano',  
'pop',  
'popelectronic',  
'popeurodance',  
'post',  
'posthardcore',  
'postrock',  
'power',  
'progmetal',  
'progressive',  
'psychedelic',  
'punjabi',  
'punk',  
'quebecois',  
'ragga',  
'ram',  
'rancheras',  
'rap',  
'rave',  
'reggae',  
'reggaeton',  
'regional',  
'relax',  
'religious',  
'retro',  
'rhythm',  
'rnb',  
'rnr',  
'rock',  
'rockabilly',  
'romance',  
'roots',  
'ruspop',  
'rusrap',  
'rusrock',  
'salsa',  
'samba',  
'schlager',  
'self',  
'sertanejo',  
'shoegazing',  
'showtunes',  
'singer',  
'ska',  
'slow',  
'smooth',  
'soul',  
'soulful',  
'sound',  
'soundtrack',  
'southern',  
'specialty',  
'speech',  
'spiritual',  
'sport',  
'stonerrock',  
'surf',  
'swing',

```
'synthpop',
'sängerportrait',
'tango',
'tanzorchester',
'taraftar',
'tech',
'techno',
'thrash',
'top',
'traditional',
'tradjazz',
'trance',
'tribal',
'trip',
'triphop',
'tropical',
'türk',
'türkçe',
'unknown',
'urban',
'uzbek',
'variété',
'vi',
'videogame',
'vocal',
'western',
'world',
'worldbeat',
'iii']
```

```
In [72]: # fungsi untuk mengganti duplikat implisit
def replace_wrong_genres(wrong_genres, correct_genre):
    for wrong_genre in wrong_genres:
        df['genre'] = df['genre'].replace(wrong_genre, correct_genre)
```

```
In [73]: # menghapus duplikat implisit
wrong_genre_1 = ['hip', 'hop', 'hip-hop']
correct_genre_1 = 'hiphop'

replace_wrong_genres(wrong_genre_1, correct_genre_1)
```

/tmp/ipykernel\_103/1174385683.py:4: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
df['genre'] = df['genre'].replace(wrong\_genre, correct\_genre)

Pastikan nama duplikat telah dihapus. Tampilkan daftar nilai unik dari kolom 'genre' :

```
In [74]: # memeriksa duplikat implisit
print(df.duplicated().sum())
```

0

[Kembali ke Daftar Isi](#)

## Kesimpulan

Kita mendeteksi tiga masalah dengan data:

- Gaya penulisan judul yang salah
- Nilai-nilai yang hilang
- Duplikat eksplisit dan implisit

Judulnya pun sekarang telah dibersihkan untuk mempermudah pemrosesan tabel. Semua nilai yang hilang telah diganti dengan `'unknown'`. Tapi kita masih harus melihat apakah nilai yang hilang dalam `'genre'` akan memengaruhi perhitungan kita.

Tidak adanya duplikat akan membuat hasil lebih tepat dan lebih mudah dipahami.

Sekarang kita dapat melanjutkan ke pengujian hipotesis.

[Kembali ke Daftar Isi](#)

## Tahap 3. Menguji Hipotesis

### Hipotesis 1: Membandingkan Perilaku Pengguna di Dua Kota

Menurut hipotesis pertama, pengguna dari Springfield dan Shelbyville memiliki perbedaan perilaku dalam mendengarkan musik. Pengujian ini menggunakan data pada hari: Senin, Rabu, dan Jumat.

- Pisahkan pengguna ke dalam kelompok berdasarkan kota.
- Bandingkan berapa banyak lagu yang dimainkan setiap kelompok pada hari Senin, Rabu, dan Jumat.

```
In [77]: # Menghitung Lagu yang diputar di setiap kota
print(df.groupby('city')['city'].count())
```

```
city
Shelbyville    18512
Springfield    42741
Name: city, dtype: int64
```

Springfield memiliki lebih banyak lagu yang dimainkan daripada Shelbyville. Namun bukan berarti warga Springfield lebih sering mendengarkan musik. Kota ini lebih besar, dan memiliki lebih banyak pengguna.

Sekarang kelompokkan data menurut hari dan temukan jumlah lagu yang diputar pada hari Senin, Rabu, dan Jumat.

```
In [62]: # Menghitung trek yang diputar pada masing-masing hari
```

```
print(df.groupby('city')['city'].count())
```

```
city
Shelbyville    18512
Springfield    42741
Name: city, dtype: int64
```

Rabu adalah hari paling tenang secara keseluruhan. Tetapi jika kita mempertimbangkan kedua kota secara terpisah, kita mungkin akan memiliki kesimpulan yang berbeda.

Fungsi `number_tracks()` untuk menghitung jumlah lagu yang diputarkan untuk hari dan kota tertentu. Ini akan membutuhkan dua parameter:

- nama hari
- nama kota

```
In [36]: def number_track(day, city):
        users = df[df['day'] == day]
        track = users[users['city'] == city]
        track_list_count = track.groupby('user_id')['track'].count()
        return(track_list_count.sum())
```

```
In [37]: # jumlah lagu yang diputarkan di Springfield pada hari Senin
        print(number_track('Monday', 'Springfield'))

15740
```

```
In [38]: # jumlah lagu yang diputarkan di Shelbyville pada hari Senin
        print(number_track('Monday', 'Shelbyville'))

5614
```

```
In [39]: # jumlah lagu yang diputarkan di Springfield pada hari Rabu
        print(number_track('Wednesday', 'Springfield'))

11056
```

```
In [40]: # jumlah lagu yang diputarkan di Shelbyville pada hari Rabu
        print(number_track('Wednesday', 'Shelbyville'))

7003
```

```
In [41]: # jumlah lagu yang diputarkan di Springfield pada hari Jumat
        print(number_track('Friday', 'Springfield'))

15945
```

```
In [42]: # jumlah lagu yang diputarkan di Shelbyville pada hari Jumat
        print(number_track('Friday', 'Shelbyville'))

5895
```

```
In [117... # tabel dengan hasil
        data = [['Springfield', 15740, 11056, 15945],
                ['Shelbyville', 5614, 7003, 5895]]
        new_columns = ['city', 'monday', 'wednesday', 'friday']
        new_table = pd.DataFrame(data = data, columns = new_columns)

        new_table
```

Out[117]:

	city	monday	wednesday	friday
0	Springfield	15740	11056	15945
1	Shelbyville	5614	7003	5895

## Kesimpulan

Data mengungkapkan perbedaan perilaku pengguna:

- Pada Springfield, jumlah lagu yang diputar mencapai puncaknya pada hari Senin dan Jumat, sedangkan pada hari Rabu terjadi penurunan aktivitas.
- Di Shelbyville, sebaliknya, pengguna lebih banyak mendengarkan musik pada hari Rabu.

Aktivitas pengguna pada hari Senin dan Jumat lebih sedikit.

[Kembali ke Daftar Isi](#)

## Hipotesis 2: Musik di Awal dan Akhir Minggu

Menurut hipotesis kedua, pada Senin pagi dan Jumat malam, warga Springfield mendengarkan genre yang berbeda dari yang dinikmati warga Shelbyville.

```
In [69]: # mendapatkan tabel spr_general dari baris df,
# di mana nilai dari kolom 'city' adalah 'Springfield'
spr_general = df[df['city']=='Springfield']
```

```
In [70]: # mendapatkan shel_general dari baris df,
# di mana nilai dari kolom 'city' adalah 'Shelbyville'
shel_general = df[df['city']=='Shelbyville']
```

Fungsi `genre_weekday()` dengan empat parameter:

- Sebuah tabel untuk data
- Nama hari
- Tanda waktu pertama, dalam format 'hh:mm'
- Tanda waktu terakhir, dalam format 'hh: mm'

```
In [75]: def genre_weekday(data, day, time1, time2):
genre_df = data[data['day']==day]
genre_df = genre_df[genre_df['time']<time2]
genre_df = genre_df[genre_df['time']>time1]
genre_df_grouped = genre_df.groupby('genre')['genre'].count()
genre_df_sorted = genre_df_grouped.sort_values(ascending=False)
return genre_df_sorted[:15]
```

Membandingkan hasil fungsi `genre_weekday()` untuk Springfield dan Shelbyville pada Senin pagi (dari pukul 07.00 hingga 11.00) dan pada Jumat malam (dari pukul 17:00 hingga 23:00):

```
In [76]: # memanggil fungsi untuk Senin pagi di Springfield
genre_weekday (spr_general, 'Monday', '07:00:00', '11:00:00')
```

```
Out[76]: genre
pop      781
dance    549
electronic 480
rock     474
hiphop   286
ruspop   186
world    181
rusrap   175
alternative 164
classical 157
metal    120
jazz     100
folk     97
soundtrack 95
latin    79
Name: genre, dtype: int64
```

```
In [77]: # memanggil fungsi untuk Senin pagi di Shelbyville
genre_weekday (shel_general, 'Monday', '07:00:00', '11:00:00')
```

```
Out[77]: genre
pop      218
dance    182
rock     162
electronic 147
hiphop    80
ruspop    64
alternative 58
rusrap    55
jazz      44
classical 40
world     36
rap       32
soundtrack 31
rnb       27
metal     27
Name: genre, dtype: int64
```

```
In [78]: # memanggil fungsi untuk Jumat malam di Springfield
genre_weekday (spr_general, 'Friday', '17:00:00', '23:00:00')
```

```
Out[78]: genre
pop      713
rock     517
dance    495
electronic 482
hiphop   273
world    208
ruspop   170
alternative 163
classical 163
rusrap   142
jazz     111
soundtrack 105
rnb      90
metal    88
folk     83
Name: genre, dtype: int64
```

```
In [79]: # memanggil fungsi untuk Jumat malam di Shelbyville  
genre_weekday (shel_general, 'Friday', '17:00:00', '23:00:00')
```

```
Out[79]: genre  
pop      256  
electronic 216  
rock      216  
dance     210  
hiphop     97  
alternative 63  
jazz       61  
classical  60  
rusrap     59  
world      54  
ruspop     47  
soundtrack 40  
metal      39  
rap        36  
latin      35  
Name: genre, dtype: int64
```

## Kesimpulan

Setelah membandingkan 15 genre teratas pada Senin pagi, kita dapat menarik kesimpulan berikut:

1. Pengguna dari Springfield dan Shelbyville mendengarkan musik dengan genre yang sama. Lima genre teratas sama, hanya rock dan elektronik yang bertukar tempat.
2. Di Springfield, jumlah nilai yang hilang ternyata sangat besar sehingga nilai 'unknown' berada di urutan ke-10. Ini berarti bahwa nilai-nilai yang hilang memiliki jumlah data yang cukup besar, yang mungkin menjadi dasar untuk mempertanyakan ketepatan kesimpulan kita.

Untuk Jumat malam, situasinya serupa. Genre individual cukup bervariasi, tetapi secara keseluruhan, 15 besar genre untuk kedua kota sama.

Dengan demikian, hipotesis kedua sebagian terbukti benar:

- Pengguna mendengarkan musik yang sama di awal dan akhir minggu.
- Tidak ada perbedaan yang mencolok antara Springfield dan Shelbyville. Pada kedua kota tersebut, pop adalah genre yang paling populer.

Namun, jumlah nilai yang hilang membuat hasil ini dipertanyakan. Di Springfield, ada begitu banyak yang memengaruhi 15 teratas kita. Jika kita tidak mengabaikan nilai-nilai ini, hasilnya mungkin akan berbeda.

[Kembali ke Daftar Isi](#)

## Hipotesis 3: Preferensi Genre di Springfield dan Shelbyville

Hipotesis: Shelbyville menyukai musik rap. Warga Springfield lebih menyukai pop.



```
In [50]: spr_group = spr_general.groupby('genre')['genre'].count()
spr_genres = spr_group.sort_values(ascending=False)
```

```
In [51]: # menampilkan 10 baris pertama dari spr_genres
print(spr_genres[:10])
```

```
genre
pop          5892
dance        4435
rock         3965
electronic   3786
hiphop       2096
classical    1616
world        1432
alternative  1379
ruspop       1372
rusrap       1161
Name: genre, dtype: int64
```

```
In [52]: shel_group = shel_general.groupby('genre')['genre'].count()
shel_genres = shel_group.sort_values(ascending=False)
```

```
In [53]: # menampilkan 10 baris pertama dari shel_genres
print(shel_genres[:10])
```

```
genre
pop          2431
dance        1932
rock         1879
electronic   1736
hiphop        960
alternative   649
classical     646
rusrap        564
ruspop        538
world         515
Name: genre, dtype: int64
```

## Kesimpulan

Hipotesis terbukti benar sebagian:

- Musik pop adalah genre paling populer di Springfield, seperti yang diharapkan.
- Namun, musik pop ternyata sama populernya baik di Springfield maupun di Shelbyville, dan musik rap tidak berada di 5 besar untuk kedua kota tersebut.

[Kembali ke Daftar Isi](#)

## Temuan

Kita telah menguji tiga hipotesis berikut:

1. Aktivitas pengguna berbeda-beda tergantung pada hari dan kotanya.

2. Pada senin pagi, penduduk Springfield dan Shelbyville mendengarkan genre yang berbeda. Hal ini juga berlaku untuk Jumat malam.
3. Pendengar di Springfield dan Shelbyville memiliki preferensi yang berbeda. Baik Springfield maupun di Shelbyville, mereka lebih suka musik pop.

Setelah menganalisis data, kita dapat menyimpulkan:

1. Aktivitas pengguna di Springfield dan Shelbyville bergantung pada harinya, walaupun kotanya berbeda.

Hipotesis pertama dapat diterima sepenuhnya.

1. Preferensi musik tidak terlalu berbeda selama seminggu di Springfield dan Shelbyville. Kita dapat melihat perbedaan kecil dalam urutan pada hari Senin, tetapi:
  - Baik di Springfield maupun di Shelbyville, orang paling banyak mendengarkan musik pop.

Jadi hipotesis ini tidak dapat kita terima. Kita juga harus ingat bahwa hasilnya mungkin saja akan berbeda jika bukan karena nilai yang hilang.

1. Ternyata preferensi musik pengguna dari Springfield dan Shelbyville sangat mirip.

Hipotesis ketiga ditolak. Jika memang ada perbedaan preferensi, ia tidak dapat dilihat dari data ini.

[Kembali ke Daftar Isi](#)