41247057S 陳育渝

Apple Music Playlist Analysis

Dataset

`J.csv` is my Japanese song playlist from Apple Music. I output it to `.txt`, and let AI turn into `.csv`.

This dataset contains information such as song titles, singers, and playing times.

I want to analyze which singers are my favorites, and which songs I listen to the most.

Data Pre-Processing

The Japanese singer may have one more name. For example: `藤井風` is also `Fujii Kaze`. I need to merge these data.

```
artist_mapping = {
    'Fujii Kaze': '藤井風',
}
df['藝人'] = df['藝人'].replace(artist_mapping)
```

Analysis Objectives

1. Top 10 Artists I Most Frequently Listen To

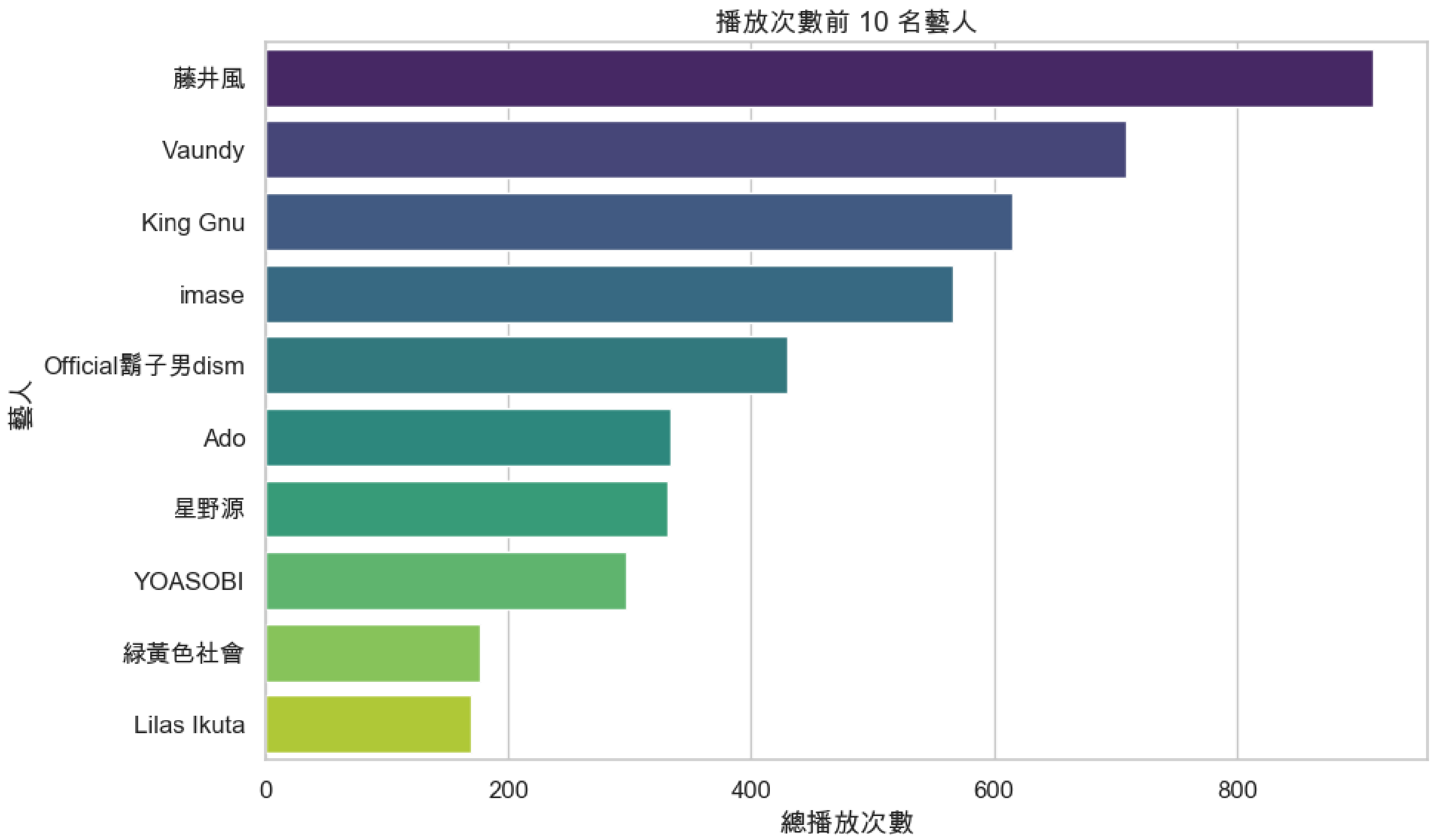
Pseudocode:

```
1. Group data by artist
2. Sum playing times for each artist
3. Sort by total playing times in descending order
4. Select top 10 artists
5. Create horizontal bar chart for visualization
```

Algorithm:

```
# Group by artist and sum playing times
top_artists = (
    df.groupby('藝人')['播放次數']
    .sum()
    .sort_values(ascending=False)
    .head(10)
)

# Visualize with horizontal bar chart
sns.barplot(x=top_artists.values, y=top_artists.index, palette="viridis")
```



2. Artist Song Count Distribution

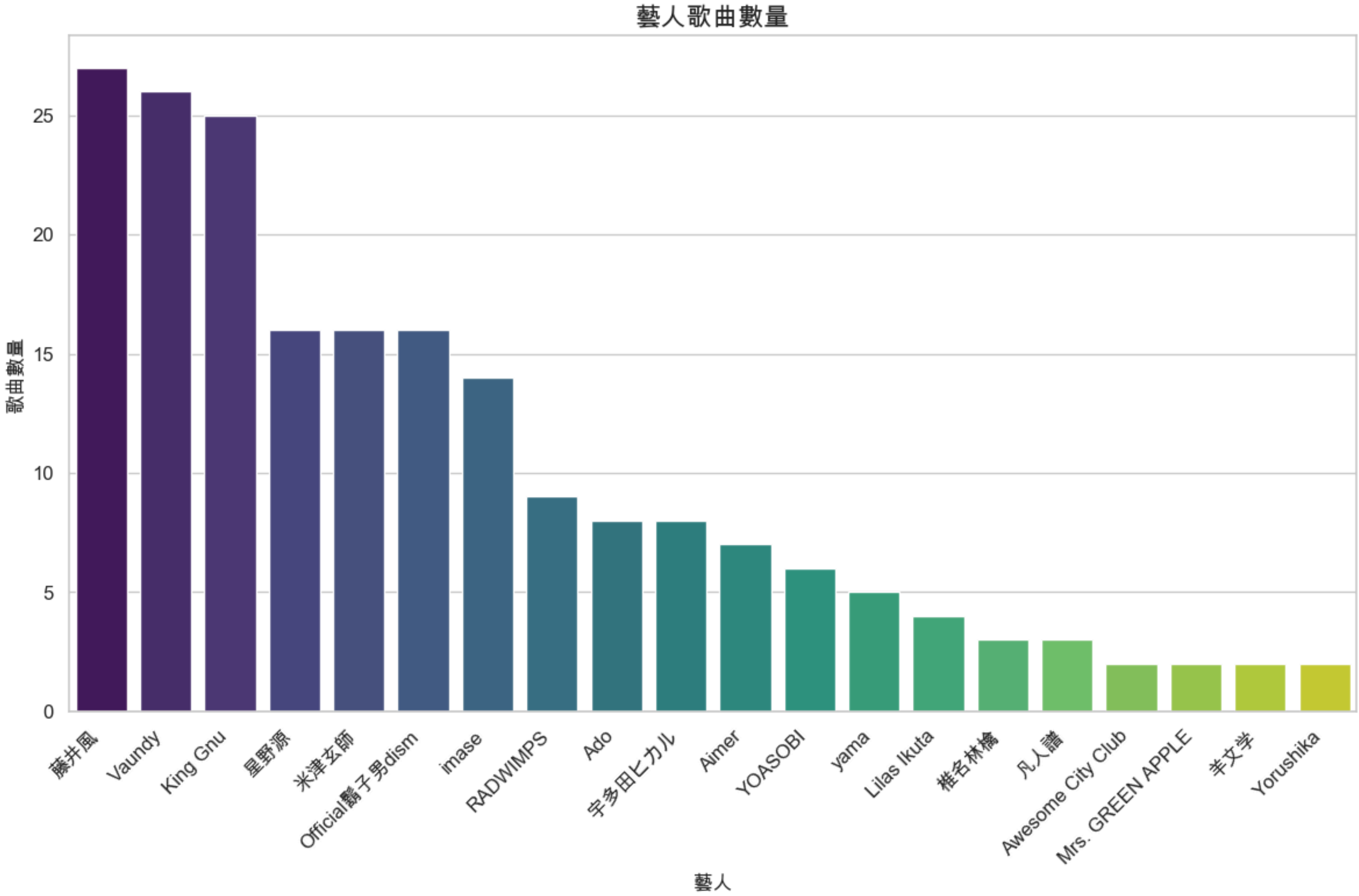
Pseudocode:

```
1. Count number of songs per artist
2. Select top N artists (configurable)
3. Create bar chart showing song counts
4. Calculate percentage distribution
```

Algorithm:

```
# Count songs per artist
artist_counts = df['藝人'].value_counts()
top_artists = artist_counts.head(20)

# Create bar chart
sns.barplot(x='藝人', y='歌曲數量', data=plot_data, palette="viridis")
```



3. Top 20 Most Played Songs

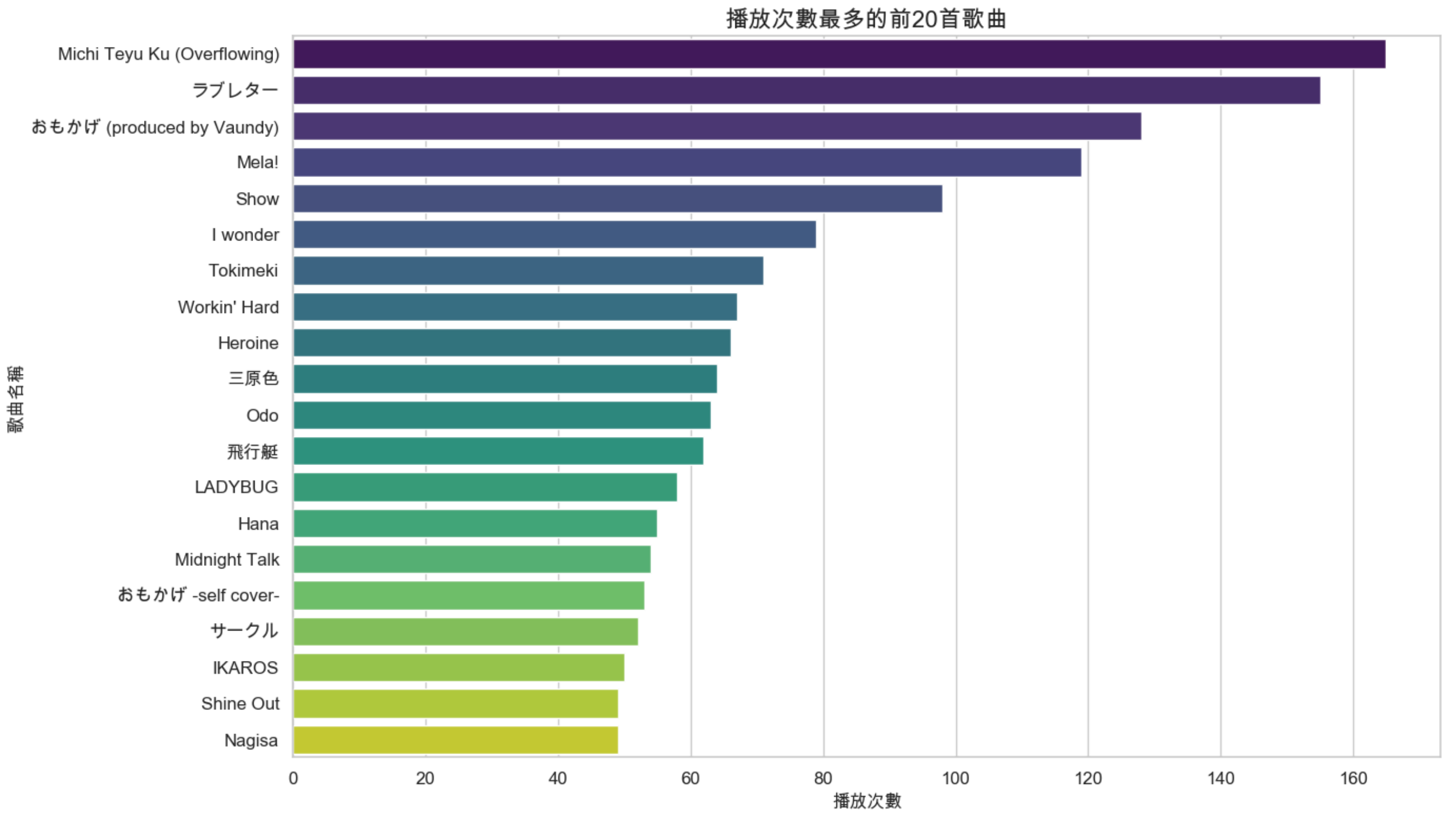
Pseudocode:

```
1. Sort all songs by playing times
2. Select top 20 songs
3. Create horizontal bar chart
4. Display detailed ranking list
```

Algorithm:

```
# Sort by playing times and get top 20
top_songs = df.sort_values('播放次數', ascending=False).head(20)

# Create horizontal bar chart
sns.barplot(x='播放次數', y='名稱', data=top_songs, palette="viridis")
```



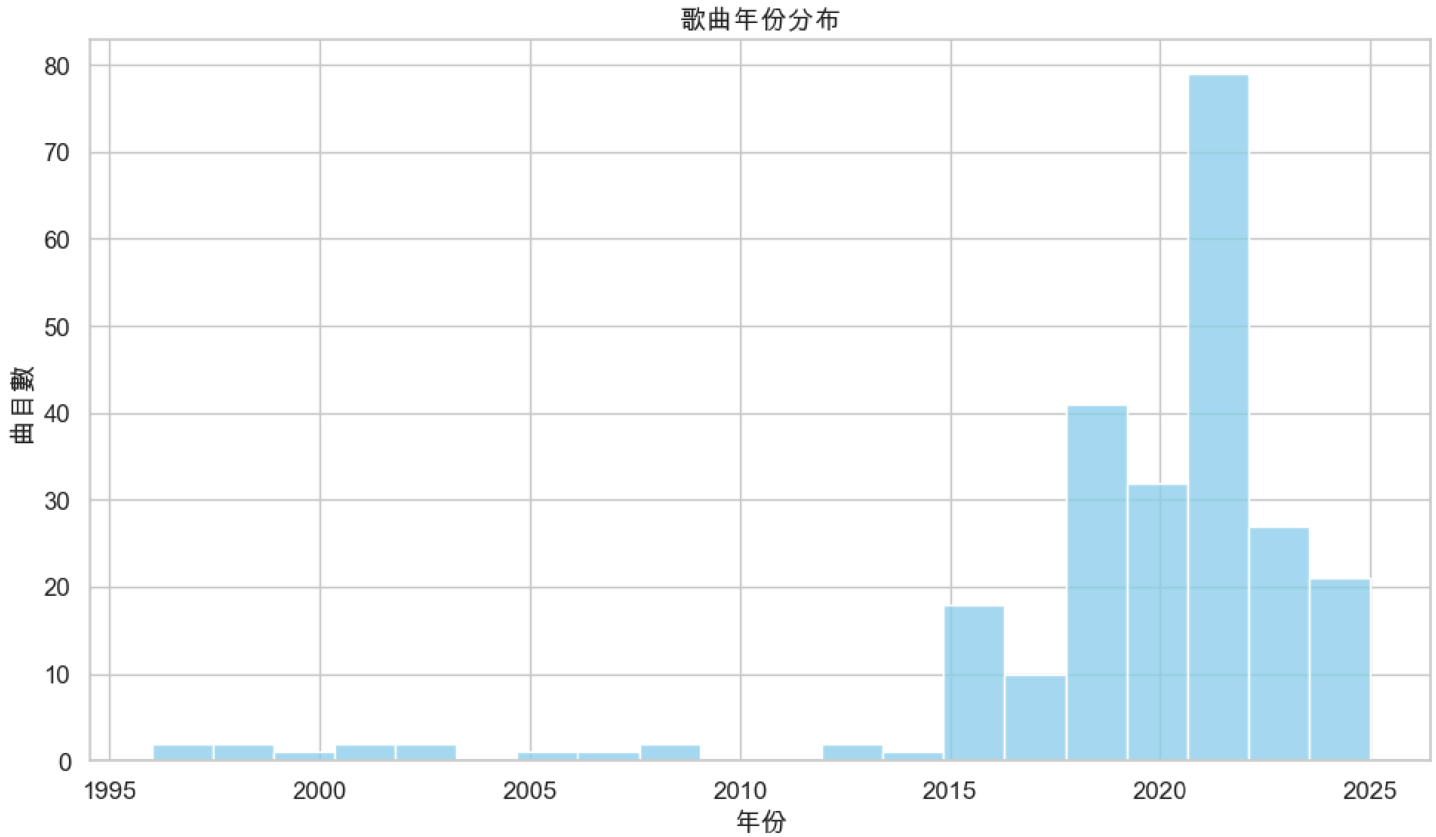
4. Song Year Distribution

Pseudocode:

```
1. Extract year information from dataset
2. Remove null values
3. Create histogram with appropriate bins
4. Show distribution pattern
```

Algorithm:

```
# Create histogram for year distribution
sns.histplot(df['年份'].dropna(), bins=20, kde=False, color='skyblue')
```



Key Findings

- The most played artist is **藤井風**.
- The most listened-to song is **"Michi Teyu Ku (Overflowing)"** by **藤井風**, with 165 plays.
- Most songs in this playlist are from 2018–2022, indicating a recent music preference.

Files Structure

```
myMusic/
├── apple_music_analysis.ipynb
├── J.csv
├── README.md
└── .gitignore
```

GitHub Repository