
usbank
stadium

Whether or not you like football, the Super Bowl is a spectacle. There's a little something for everyone at your Super Bowl party. Drama in the form of blowouts, comebacks, and controversy for the sports fan. There are the ridiculously expensive ads, some hilarious, others gut-wrenching, thought-provoking, and weird. The half-time shows with the biggest musicians in the world, sometimes riding giant mechanical tigers or leaping from the roof of the stadium.

The dataset we'll use was scraped and polished from Wikipedia. It is made up of three CSV files, one with game data, one with TV data, and one with halftime musician data for 52 Super Bowls through 2018.

The Data

Three datasets have been provided, and summaries and previews of each are presented below.

1. halftime_musicians.csv

This dataset contains information about the musicians who performed during the halftime shows of various Super Bowl games. The structure is shown below, and it applies to all remaining files.

Column	Description
`'super_bowl'`	The Super Bowl number (e.g., 52 for Super Bowl LII).
`'musician'`	The name of the musician or musical group that performed during the halftime show.
`'num_songs'`	The number of songs performed by the musician or group during the halftime show.

2. super_bowls.csv

This dataset provides details about each Super Bowl game, including the date, location, participating teams, and scores, including the points difference between the winning and losing team (`'difference_pts'`).

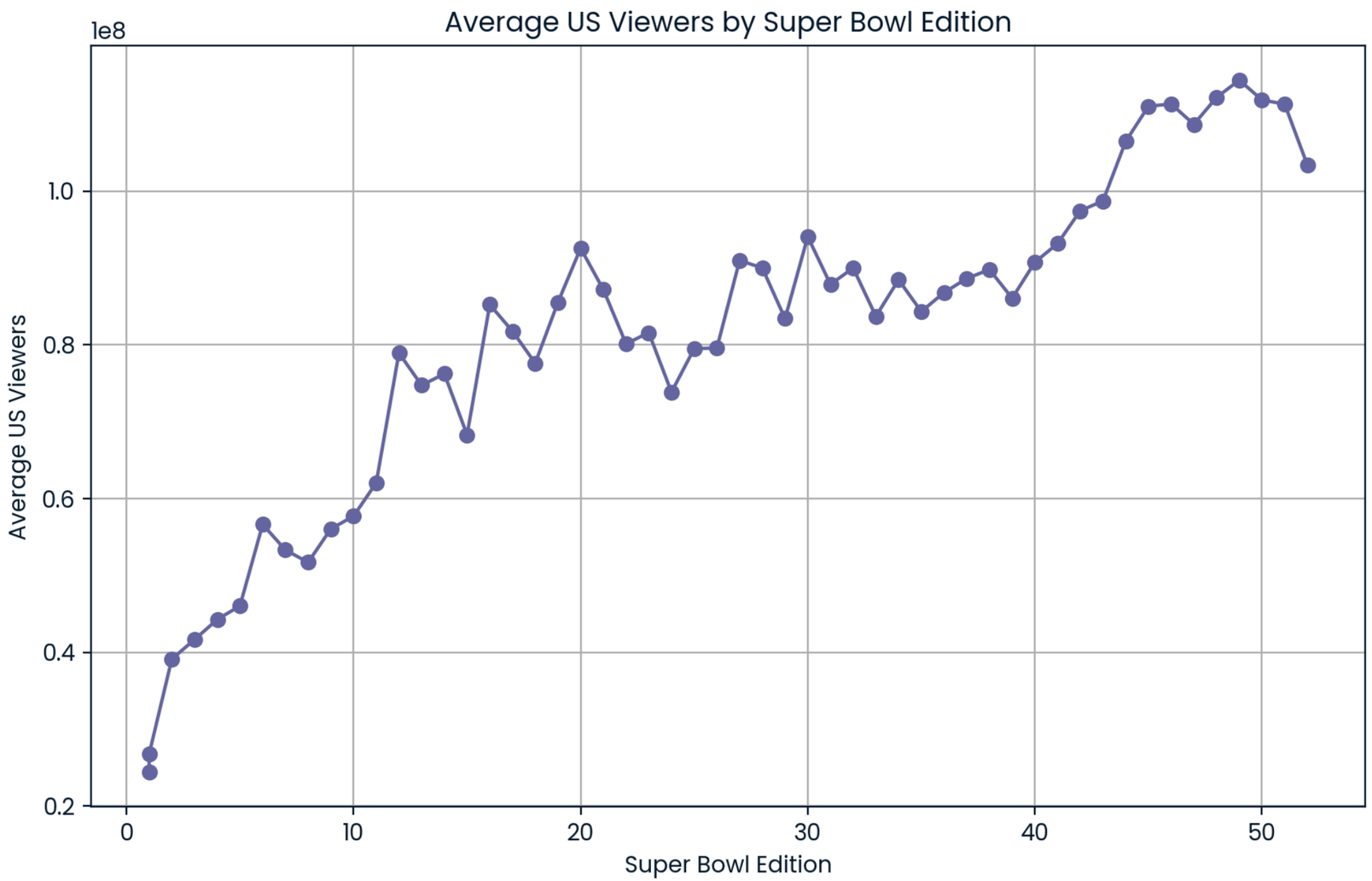
3. tv.csv

This dataset contains television viewership statistics and advertisement costs related to each Super Bowl.

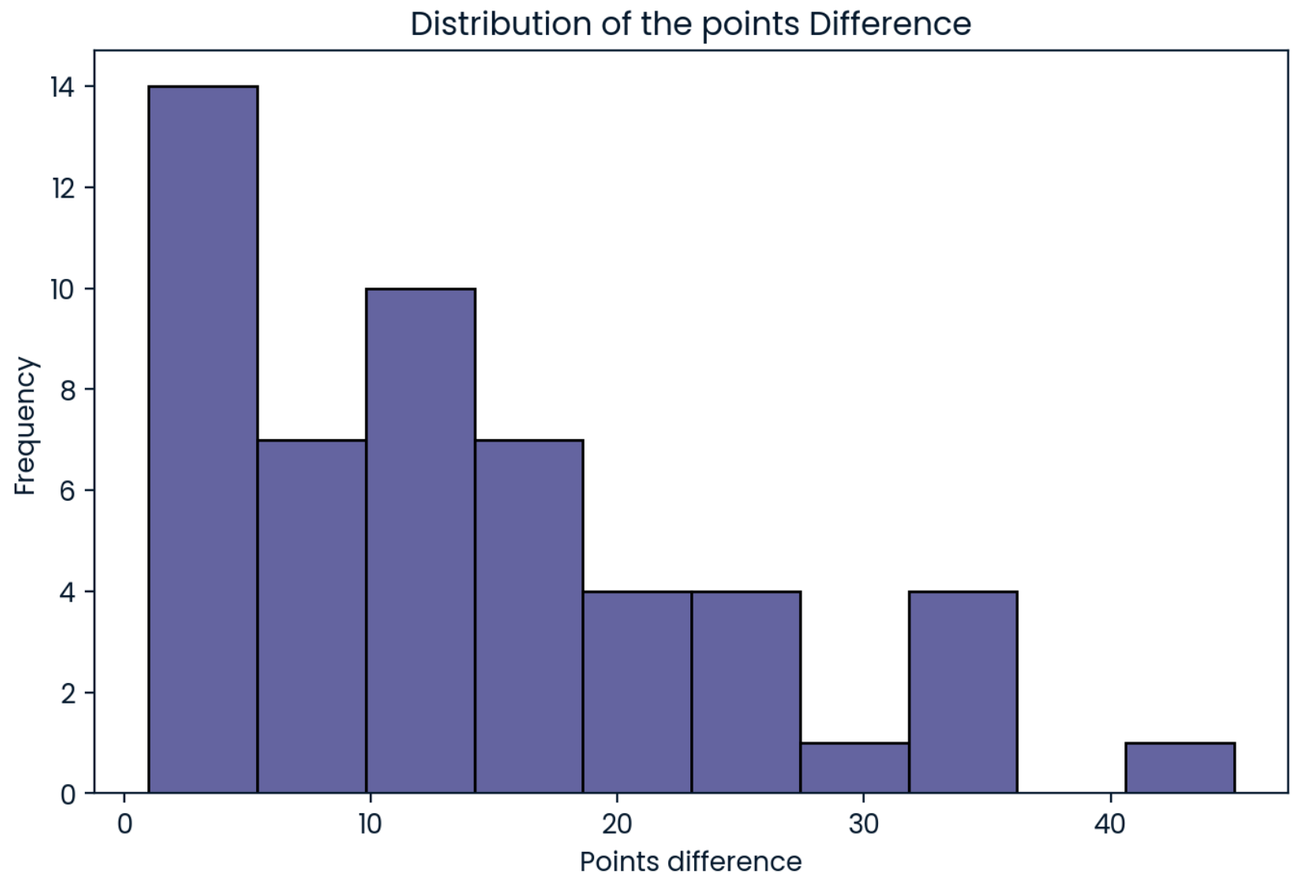
```python
# Import libraries
import pandas as pd
from matplotlib import pyplot as plt

# Load data
tv = pd.read_csv("datasets/tv.csv")
super_bowls = pd.read_csv("datasets/super_bowls.csv")
halftime_musicians = pd.read_csv("datasets/halftime_musicians.csv")

# Merge the 'year' information from super_bowls into tv DataFrame
tv = tv.merge(super_bowls[["super_bowl", "date"]], on="super_bowl", how="left")

# Extract year from the 'date' column and create a new 'year' column
tv["year"] = pd.to_datetime(tv["date"]).dt.year

# Group by year and calculate the mean average US viewers per year
yearly_viewership = tv.groupby("year")["avg_us_viewers"].mean().sort_index()

# Determine if viewership increased from the first to the last year (boolean)
viewership_increased = bool(yearly_viewership.iloc[-1] > yearly_viewership.iloc[0])

print("Has TV viewership increased over time?", viewership_increased)

# Plot the trend of avg_us_viewers by Super Bowl edition
plt.figure(figsize=(10, 6))
plt.plot(tv.sort_values("super_bowl")["super_bowl"], tv.sort_values("super_bowl")["avg_us_viewers"], marker='o')
plt.xlabel("Super Bowl Edition")
plt.ylabel("Average US Viewers")
plt.title("Average US Viewers by Super Bowl Edition")
plt.grid(True)
plt.show()

# Count the number of Super Bowls with a point difference greater than 40
difference = int((super_bowls['difference_pts'] > 40).sum())
print("Number of Super Bowl matches with a difference greater than 40 is =", difference)

# Plot of the difference in points between matches
plt.figure(figsize=(8,5))
plt.hist(super_bowls["difference_pts"], bins=10, edgecolor="black")
plt.xlabel("Points difference")
plt.ylabel("Frequency")
plt.title("Distribution of the points Difference")
```

```
plt.show()

# Aggregate first, then sort
songs_per_musician = halftime_musicians.groupby("musician")["num_songs"].sum()
most_songs1 = songs_per_musician.sort_values(ascending=False)
most_songs = most_songs1.idxmax()
print("Musicians with the most songs in the Super Bowl:", most_songs)
```

```
Has TV viewership increased over time? True
```




```
Number of Super Bowl matches with a difference greater than 40 is = 1
```




```
Musicians with the most songs in the Super Bowl: Justin Timberlake
```