INFO 430: Database Design and Management
Project 2
Movie Data Collection and Analysis Using MongoDB
Muhamed Hashi
May 22, 2025

Part 1:

For my INFO 430 Group Project 2, I chose to focus on movies and will use MongoDB to build a NoSQL database. I plan to collect data from sources like IMDb (https://www.imdb.com/), including movie titles, genres, directors, cast, release dates, runtimes, ratings, and box office performance. This data will be extracted using web scraping, processed, and stored in MongoDB. My goal is to explore patterns in the film industry and answer questions such as which genres perform best, how ratings correlate with revenue, and which directors or studios produce consistently successful films. MongoDB's flexible schema makes it ideal for handling the semi-structured nature of movie data. MongoDB is also widely used in media, including by companies like MTV and Shutterfly, for managing complex and variable data such as multimedia content and user interactions. This aligns well with my movie data project, which includes a variety of fields that can differ from one movie to another. This project will demonstrate my skills in web scraping, data processing, and querying NoSQL databases to uncover useful insights.

Part 2:

Part 3:

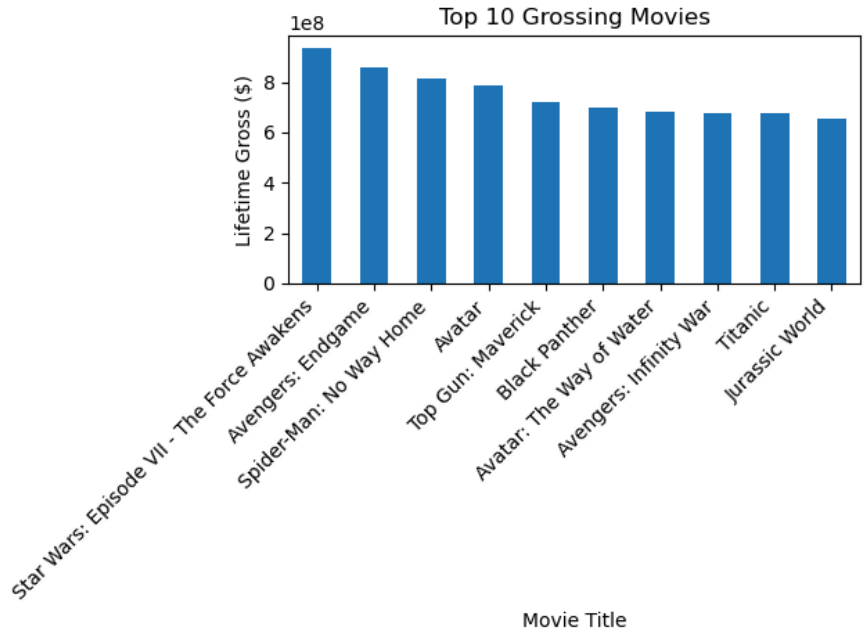
Query 1: Top 10 Grossing Movies

- Purpose: Identify the top 10 movies with the highest lifetime gross

Code:

```
top_10 = collection.find(
    {},
    {"_id": 0, "title": 1, "lifetime_gross": 1}
).sort("lifetime_gross", -1).limit(10)
```

```
df_top_10 = pd.DataFrame(list(top_10))
df_top_10

df_top_10.plot(kind='bar', x='title', y='lifetime_gross', legend=False)
plt.ylabel('Lifetime Gross ($)')
plt.xlabel('Movie Title')
plt.title('Top 10 Grossing Movies')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Discussion

- The bar chart highlights the highest-grossing films in the dataset, led by Star Wars: Episode VII – The Force Awakens, followed closely by Avengers: Endgame and Spider-Man: No Way Home. All top performers are franchise blockbusters released in the last two decades, underscoring a strategic trend in the film industry: sequels, reboots, and superhero films dominate box office earnings.

Managerial Implications

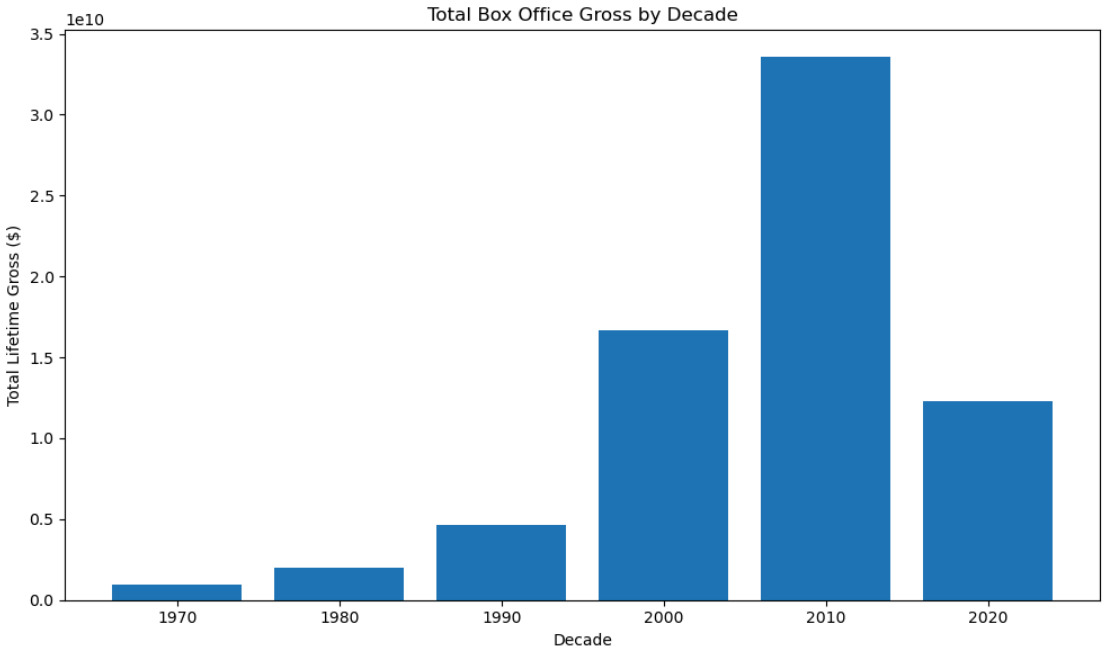
- Brand Recognition Drives Revenue: Titles with strong brand identity (Marvel, Star Wars) attract larger audiences and sustain high earnings globally.
- Future Development Strategy: Emerging franchises or new IPs may benefit from being tied to existing cinematic universes or released during peak seasons to capitalize on this trend.

Query 2: Gross Revenue by Decade

- Purpose: Show how total box office revenue trends change over decades.

Code:

```
pipeline = [
   {"$addFields": {
      "decade": {"$subtract": ["$year", {"$mod": ["$year", 10]}]}
   }},
   {"$group": {
      "_id": "$decade",
      "total_gross": {"$sum": "$lifetime_gross"}
   }},
   {"$sort": {"_id": 1}}
]

df_by_decade = pd.DataFrame(list(collection.aggregate(pipeline))).rename(columns={"_id":
"decade"})
df_by_decade

plt.figure(figsize=(10, 6))
plt.bar(df_by_decade["decade"].astype(str), df_by_decade["total_gross"])
plt.xlabel("Decade")
plt.ylabel("Total Lifetime Gross ($)")
plt.title("Total Box Office Gross by Decade")
plt.tight_layout()
plt.show()
```



Discussion

- This shows a clear upward trend in total box office revenue across decades, peaking in the 2010s with over $30 billion in gross earnings. This dramatic growth reflects not only an increase in global film distribution but also the rise of high-budget franchises and international markets. Notably, the 2020s show a significant drop, likely due to the COVID-19 pandemic's impact on theater releases.

Managerial Implications

- Decade-Based Investment Strategy: Historical performance by decade can guide future investment decisions, especially for planning long-term content portfolios.
- 2010s as a Benchmark: Studios can look to the 2010s as a benchmark for high-grossing decades, especially due to streaming crossovers and global box office expansion.

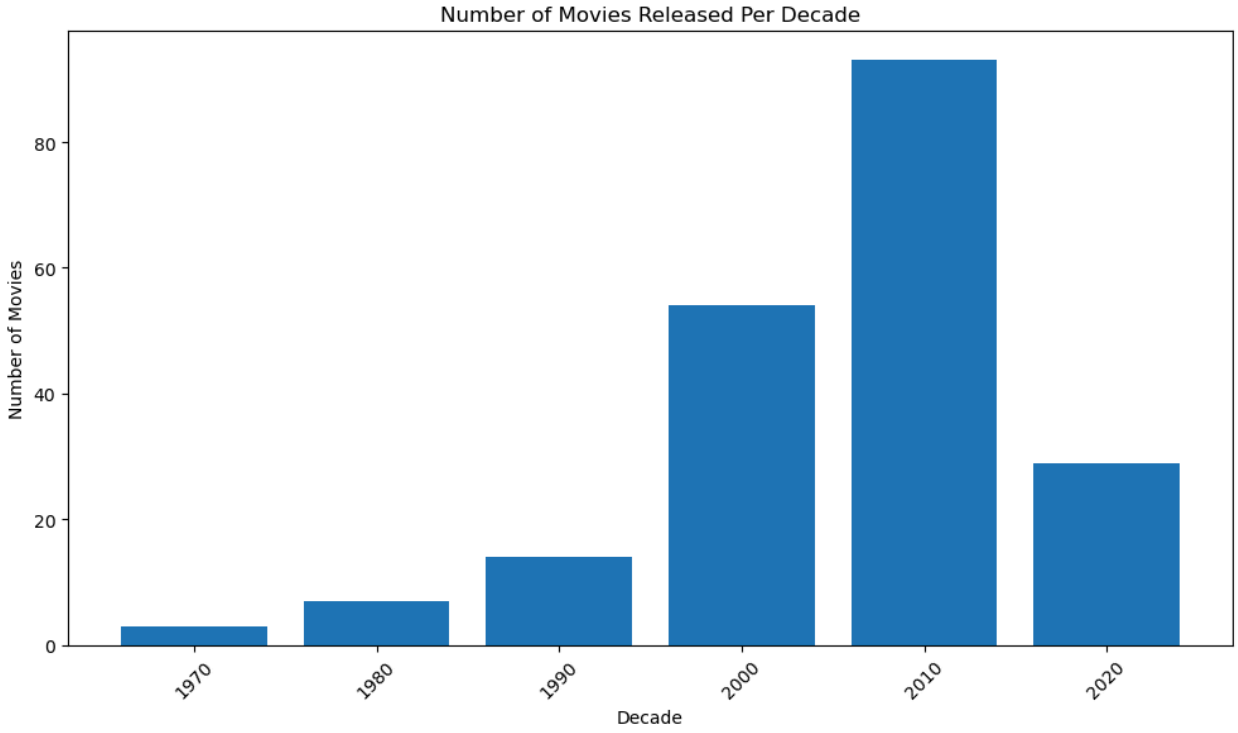
Query 3: Number of Movies per Decade

- Purpose: Counts the number of movies released in each decade to examine industry production trends and assess how content volume has evolved over time.

Code:

```
pipeline_movie_count_per_decade = [
  {"$addFields": {
    "decade": {"$subtract": ["$year", {"$mod": ["$year", 10]}]}
  }},
  {"$group": {
    "_id": "$decade",
```

```
      "movie_count": {"$sum": 1}
    }},
    {"$sort": {"_id": 1}}
]

df_movies_per_decade =
pd.DataFrame(list(collection.aggregate(pipeline_movie_count_per_decade))).rename(columns={
"_id": "decade"})
df_movies_per_decade

plt.figure(figsize=(10, 6))
plt.bar(df_movies_per_decade['decade'].astype(str), df_movies_per_decade['movie_count'])
plt.xlabel('Decade')
plt.ylabel('Number of Movies')
plt.title('Number of Movies Released Per Decade')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Discussion

- The number of movie releases steadily increased from the 1970s through the 2010s, peaking in the 2010s with over 90 films in the dataset. This growth reflects broader trends such as lower production barriers, international co-productions, and rising demand for

content in theaters and streaming platforms. The 2020s show a notable decline, likely due to pandemic-related disruptions, production halts, and shifts in distribution strategy.

Managerial Implications

- Peak Content Era: The 2010s marked an era of maximum content output, signaling the height of studio production and market saturation.
- Post-2020 Recovery Opportunity: The drop in the 2020s presents a potential opportunity for studios to fill content gaps and meet pent-up audience demand.

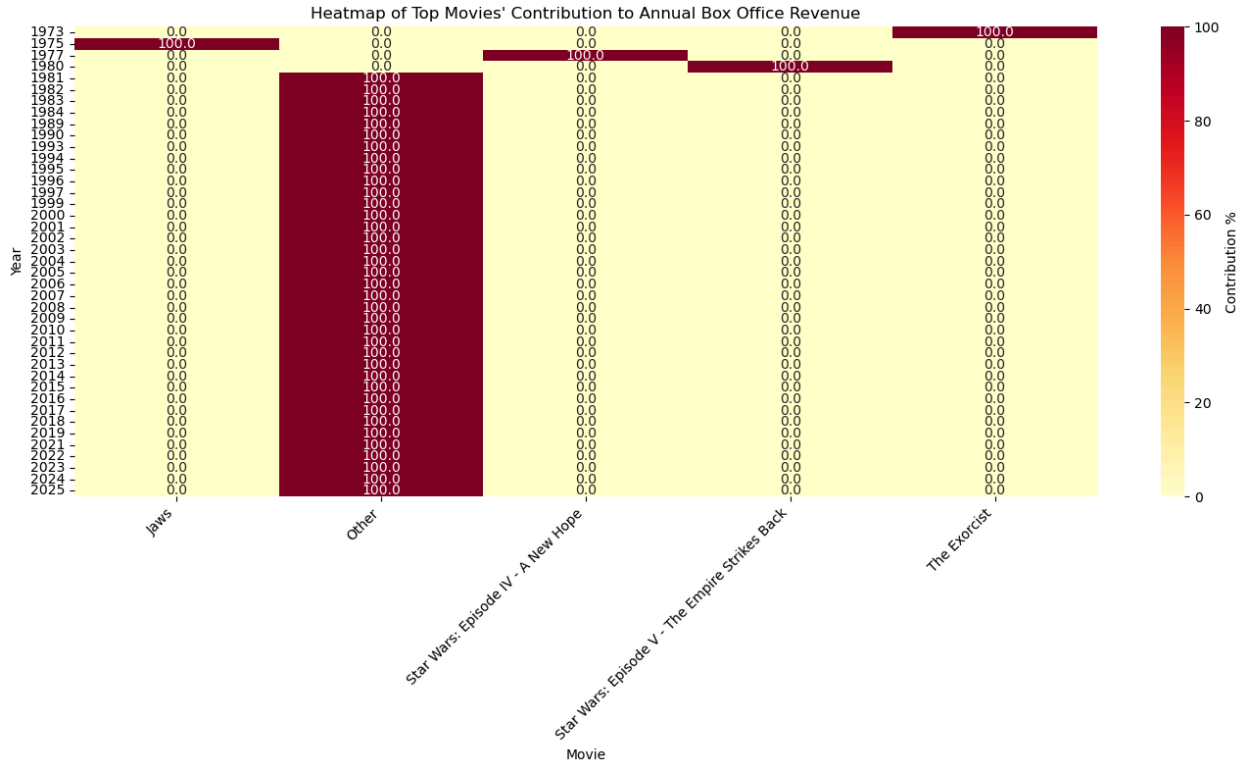
Query 4: Movie Contribution to Annual Box Office Revenue

- Purpose: Calculates each movie's percentage contribution to its release year's total box office revenue to identify the most dominant films annually.

Code:

```
pipeline_contribution_per_movie = [
  {
    "$group": {
      "_id": "$year",
      "total_year_gross": {"$sum": "$lifetime_gross"},
      "movies": {"$push": {"title": "$title", "gross": "$lifetime_gross"}}
    }
  },
  {"$unwind": "$movies"},
  {
    "$project": {
      "year": "$_id",
      "title": "$movies.title",
      "lifetime_gross": "$movies.gross",
      "contribution_pct": {
        "$multiply": [
          {"$divide": ["$movies.gross", "$total_year_gross"]},
          100
        ]
      }
    }
  },
  {"$sort": {"year": 1, "contribution_pct": -1}}
]
df_contribution_per_movie =
pd.DataFrame(list(collection.aggregate(pipeline_contribution_per_movie)))
df_contribution_per_movie
```

```
filtered = df_contribution_per_movie[df_contribution_per_movie["year"] >= 2010]

grouped = filtered.groupby(["year", "title"]).agg({
    "contribution_pct": "sum"
}).reset_index()

heatmap_df = grouped.pivot(index="year", columns="title", values="contribution_pct").fillna(0)

import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 6))
sns.heatmap(heatmap_df, annot=True, fmt=".1f", cmap="YlOrRd", cbar_kws={'label':
'Contribution %'})
plt.title("Heatmap of Movie Contribution to Yearly Gross")
plt.xlabel("Movie")
plt.ylabel("Year")
plt.tight_layout()
plt.show()
```



Discussion

- The heatmap shows that for most years, the “Other” category contributes 100% to annual box office revenue, indicating that the individually named movies either weren’t released in those years or didn’t contribute meaningfully to the overall gross. Notable exceptions include 1975, when Jaws accounted for 100% of the recorded revenue, and early entries from the Star Wars and Exorcist franchises, which also show full dominance in select years. This suggests either strong outlier performance in those years or limited data coverage for other movies. The sparse presence of named movies highlights how only a few films have historically reached blockbuster status significant enough to shape yearly totals.

Managerial Implications

- Blockbuster Dependency: Studios increasingly rely on a few major titles to drive annual revenue, suggesting a high-risk, high-reward model.
- Portfolio Diversification: The heatmap reinforces the value of investing in a balanced portfolio to avoid overdependence on a few titles.
- Data Completeness Matters: Decision-makers should ensure comprehensive box office reporting before drawing conclusions, as missing data may overstate the influence of certain titles.

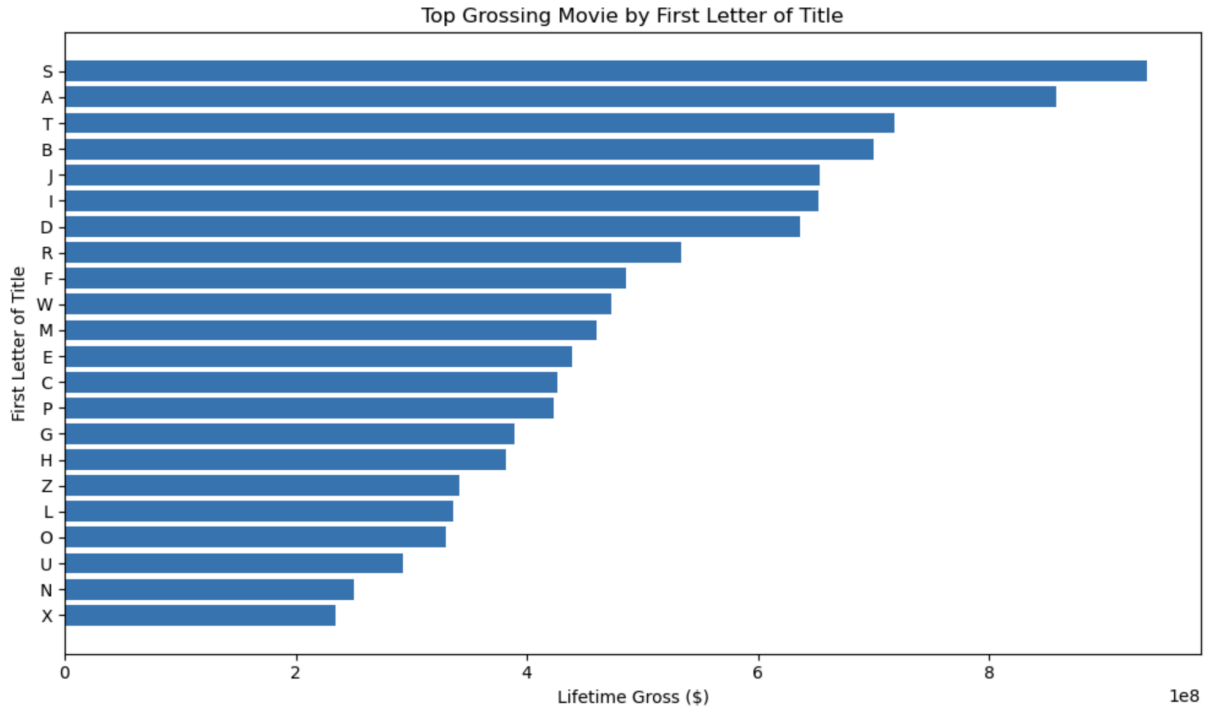
Query 5: Highest Grossing Movie by First Letter of Title

- Purpose: This query identifies the top-grossing movie for each starting letter (A–Z).

Code:

```
import string
results = []

for letter in string.ascii_uppercase:
    pipeline = [
        {"$match": {"title": {"$regex": f"^{letter}", "$options": "i"}}},
        {"$sort": {"lifetime_gross": -1}},
        {"$limit": 1}
    ]
    top_movie = list(collection.aggregate(pipeline))
    if top_movie:
        results.append({
            "letter": letter,
            "title": top_movie[0]["title"],
            "lifetime_gross": top_movie[0]["lifetime_gross"]
        })
```

```
df = pd.DataFrame(results).sort_values("lifetime_gross", ascending=True)

plt.figure(figsize=(10, 12))
plt.barh(df["letter"], df["lifetime_gross"])
plt.title("Top Grossing Movie by First Letter of Title")
plt.xlabel("Lifetime Gross ($)")
plt.ylabel("First Letter of Title")
plt.tight_layout()
plt.show()
```



Discussion

- Certain starting letters, such as "S", "A", and "B", correspond to widely successful movie franchises like Star Wars, Avengers, and Barbie. While this correlation may be coincidental, the recurrence of phonetically strong or visually recognizable letters in blockbuster titles is notable. These patterns may reflect branding decisions that favor simplicity, memorability, and cultural familiarity in naming conventions.

& Managerial Implications:

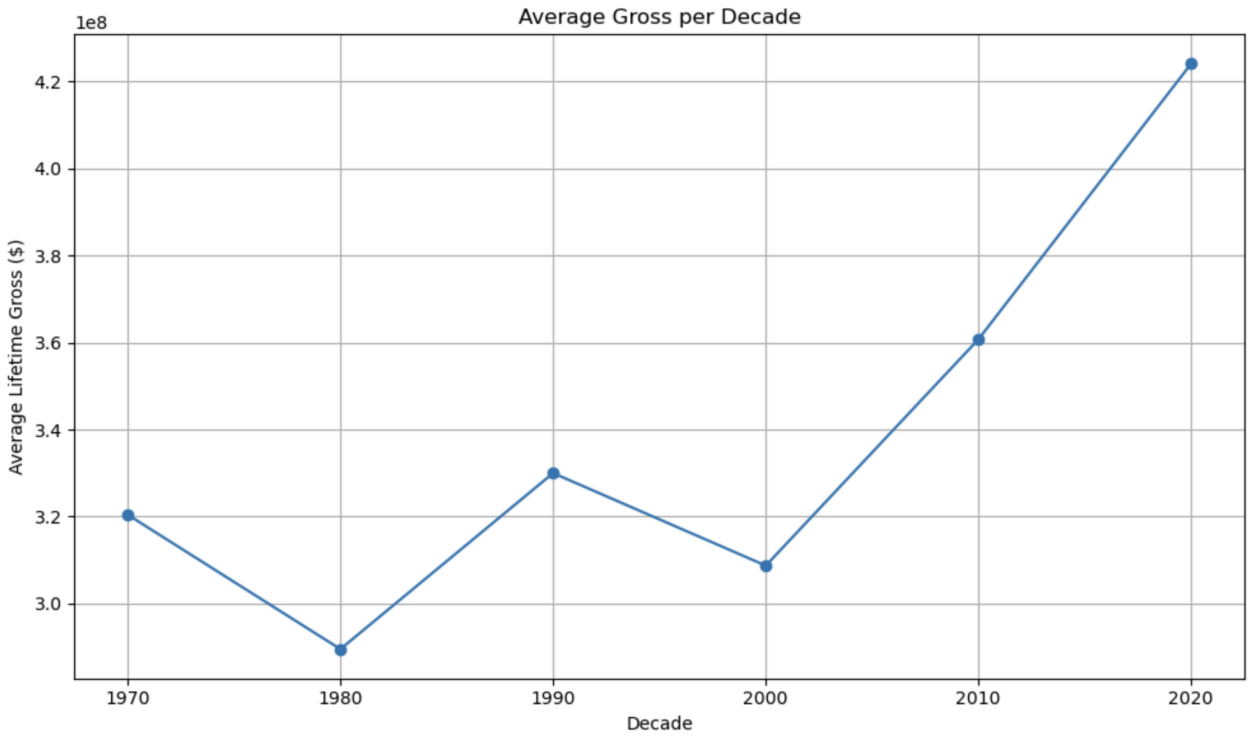
- Brand Recognition: Leveraging familiar starting letters or franchise-linked patterns could subtly influence audience recall and marketability.
- Naming as a Marketing Lever: While not a standalone factor, naming choices can complement broader marketing strategies aimed at positioning films as recognizable and culturally resonant.

Query 6: Average Gross Per Decade

- Purpose: This query calculates the average box office gross for movies released in each decade. It helps identify how profitability has evolved over time.

Code:

```
pipeline = [
    {"$project": {
        "decade": {"$subtract": ["$year", {"$mod": ["$year", 10]}]},
        "lifetime_gross": 1
    }},
    {"$group": {
        "_id": "$decade",
        "avg_gross": {"$avg": "$lifetime_gross"}
    }},
    {"$sort": {"_id": 1}}
]

results = list(collection.aggregate(pipeline))
df = pd.DataFrame(results)

plt.figure(figsize=(10, 6))
plt.plot(df["_id"], df["avg_gross"], marker='o')
plt.title("Average Lifetime Gross per Decade")
plt.xlabel("Decade")
plt.ylabel("Average Gross ($)")
plt.grid(True)
plt.tight_layout()
plt.show()
```



Discussion

- The trend reveals steadily increasing average box office grosses, particularly from the 2000s onward. This growth can be attributed to several factors, including inflation, the expansion of international markets, and the adoption of premium viewing formats like IMAX and 3D. These developments have amplified the revenue potential of theatrical releases, especially for large-scale, globally distributed productions.

Managerial Implications

- Higher Revenue Potential: Studios can forecast greater returns for blockbuster films, especially when leveraging global distribution and premium formats.
- Investment Justification: Data trends support continued investment in tentpole films, provided studios carefully manage scale, audience targeting, and global rollout strategies.

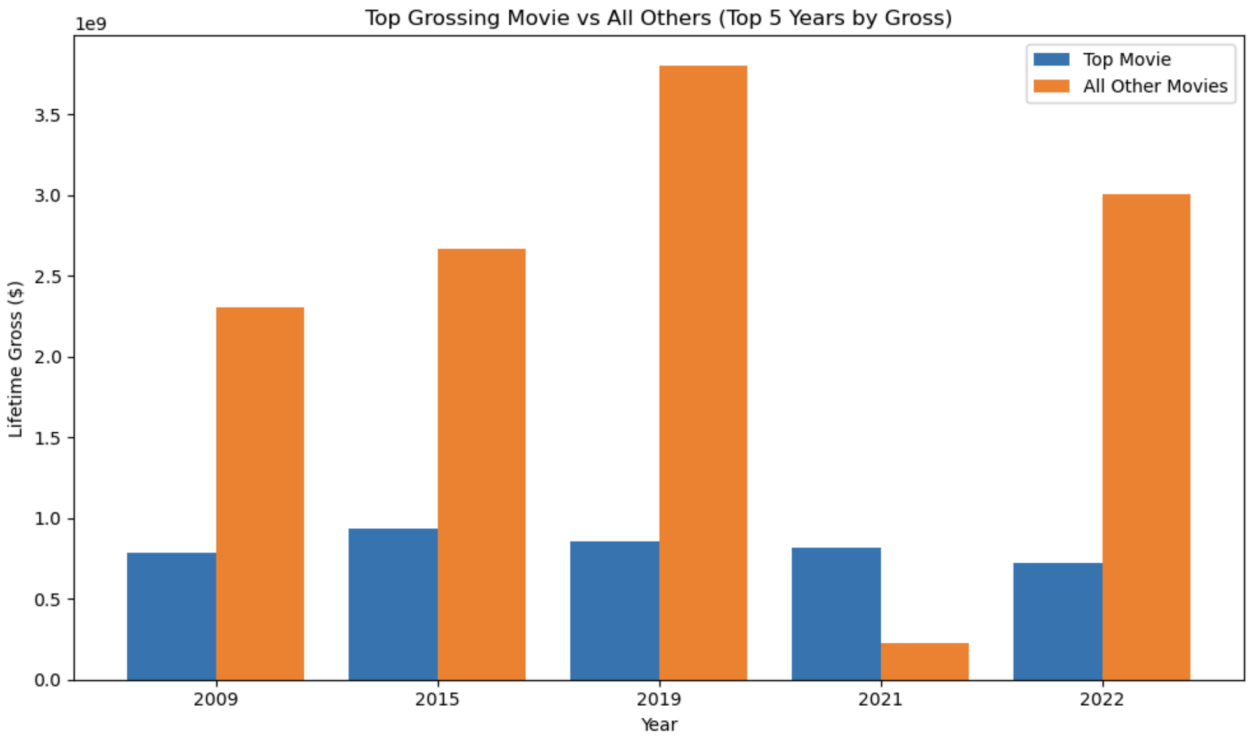
Query 7: Top Grossing Movie vs. All Others (Top 5 Years by Gross)

- Purpose: This query compares the top-earning film in each of the 5 highest-grossing years with the combined gross of all other movies released in those years.

Code:

```
top5_movies = list(collection.find().sort("lifetime_gross", -1).limit(5))

top5_years = list({movie["year"] for movie in top5_movies})
```

```
all_movies = list(collection.find({"year": {"$in": top5_years}}))

data = {}

for movie in all_movies:

    year = movie["year"]

    title = movie["title"]

    gross = movie["lifetime_gross"]

    if year not in data:

        data[year] = []

    data[year].append((title, gross))

records = []

for year in sorted(data):

    movies = sorted(data[year], key=lambda x: x[1], reverse=True)  top_movie = movies[0]
total_others = sum(g for _, g in movies[1:])

    records.append({    "year": year, "top_movie": top_movie[0],"top_gross":
top_movie[1],"others_gross": total_others

    })

df = pd.DataFrame(records)

x = range(len(df))

plt.figure(figsize=(10, 6)) plt.bar(x, df["top_gross"], width=0.4, label="Top Movie",
align='center') plt.bar([i + 0.4 for i in x], df["others_gross"], width=0.4, label="All Other
Movies", align='center') plt.xticks([i + 0.2 for i in x], df["year"]) plt.xlabel("Year")
plt.ylabel("Lifetime Gross ($)") plt.title("Top Grossing Movie vs All Others (Top 5 Years by
Gross)") plt.legend() plt.tight_layout() plt.show()
```



Discussion

- Top films in peak years often dominate the total box office, as seen with Avengers: Endgame in 2019. This indicates a high level of market concentration where mega-franchises capture a substantial share of audience attention and spending. These blockbusters can overshadow other releases, making it difficult for smaller titles to gain traction during the same periods.

Managerial Implications

- Strategic Release Timing: Studios without comparable intellectual property should avoid releasing films alongside major franchise events and instead target quieter windows to maximize visibility.
- Market Positioning: Understanding blockbuster impact helps studios make data-informed decisions about scheduling, marketing budgets, and competitive positioning.

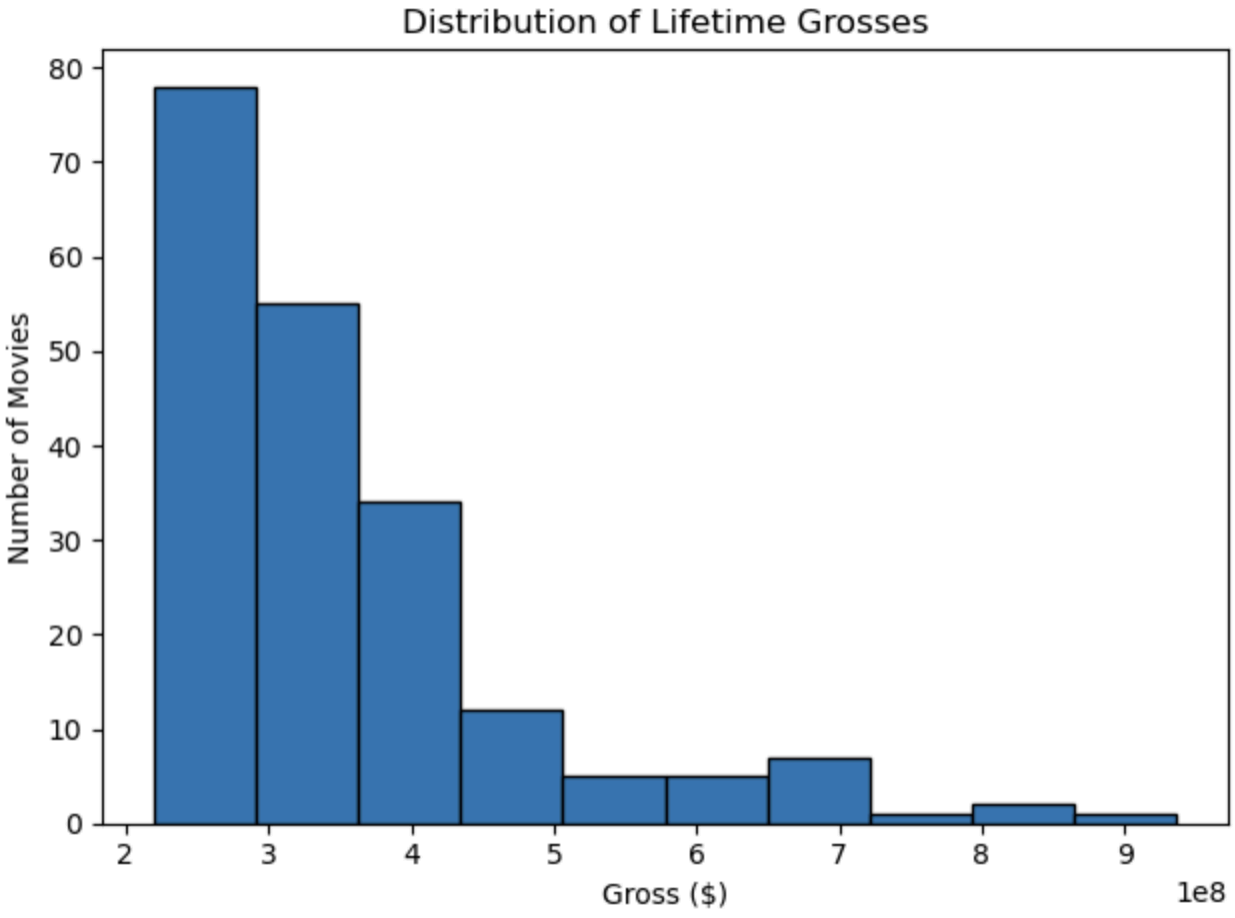
Query 8: Distribution of Lifetime Grosses

- Purpose: To examine how movie revenue is distributed across the dataset and whether a few films dominate the market.

Code:

```
cursor = collection.find({}, {"lifetime_gross": 1, "_id": 0})
```

```
grosses = [doc["lifetime_gross"] for doc in cursor if doc.get("lifetime_gross")]


plt.figure(figsize=(10, 6))
plt.hist(grosses, bins=20, edgecolor='black')
plt.title("Distribution of Lifetime Grosses")
plt.xlabel("Lifetime Gross ($)")
plt.ylabel("Number of Movies")
plt.tight_layout()
plt.show()
```



Discussion

- The histogram reveals a right-skewed distribution of box office grosses, where the majority of films earn significantly less than the top performers. This pattern reflects the dynamics of a “blockbuster economy,” in which a small number of high-grossing films generate the majority of total revenue, while most releases underperform in comparison.

Managerial Implications

- Portfolio Planning: Diversifying across genres, budgets, and audiences can help manage financial risk in a market defined by revenue outliers.
- Budget Discipline: Understanding the skewed distribution encourages more realistic budget setting and ROI expectations, particularly for mid- and low-tier productions.