Overview

The objective of this project is to analyze, and compare the box office performance of movies from different regional Indian cinema industries (Tollywood, Mollywood, Bollywood, Gollywood, Marathi, and Bengali) for the years 2019–2024. The analysis aims to derive insights on box office trends, earnings correlation, budget efficiency, and movie success metrics.

```
pip install ydata-profiling
```

```
Requirement already satisfied: tqdm<5,>=4.48.2 in /usr/local/lib/python3.10/dist-packages (from yda
Requirement already satisfied: seaborn<0.14,>=0.10.1 in /usr/local/lib/python3.10/dist-packages (fro
Collecting multimethod<2,>=1.4 (from ydata-profiling)
  Downloading multimethod-1.12-py3-none-any.whl.metadata (9.6 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: typeguard<5,>=3 in /usr/local/lib/python3.10/dist-packages (from ydat
Collecting imagehash==4.3.1 (from ydata-profiling)
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl.metadata (8.0 kB)
Requirement already satisfied: wordcloud>=1.9.3 in /usr/local/lib/python3.10/dist-packages (from yda
Collecting dacite>=1.8 (from ydata-profiling)
  Downloading dacite-1.8.1-py3-none-any.whl.metadata (15 kB)
Requirement already satisfied: numba<1,>=0.56.0 in /usr/local/lib/python3.10/dist-packages (from yda
Collecting PyWavelets (from imagehash==4.3.1->ydata-profiling)
  Downloading pywavelets-1.7.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from imagehash==4
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jin
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from mat
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplot
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from ma
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from ma
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matp
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from mat
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (fron
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from panda
Requirement already satisfied: joblib>=0.14.1 in /usr/local/lib/python3.10/dist-packages (from phik<
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/python3.10/dist-packages (fro
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from request
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from r
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from r
Requirement already satisfied: patsy>=0.5.6 in /usr/local/lib/python3.10/dist-packages (from statsmo
Requirement already satisfied: attrs>=19.3.0 in /usr/local/lib/python3.10/dist-packages (from vision
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.10/dist-packages (from vision
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-date
Downloading ydata_profiling-4.12.0-py2.py3-none-any.whl (390 kB)
```

```
import seaborn as sns
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from ydata_profiling import ProfileReport
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# Import the Movies GDP dataset from the specified CSV file using Pandas
df2 = pd.read_csv('/content/drive/MyDrive/all_data_combined_box_office_data.csv')
df2
```

Released Date	Movie	Worldwide	India Telugu Net	India Gross	Overseas	Budget	Verdict	Industry	Original Languages
25 Dec	Solo Brathuke So Better	22.3	18.10	21.1	1.2	20.0	Hit	Tollywood	Telugu
13 Mar	Shivan	0.25	0.20	0.23	0.02	2.0	NaN	Tollywood	Telugu
13 Mar	Eureka	0.35	0.25	0.3	0.05	3.0	NaN	Tollywood	Telugu
13 Mar	Madha	0.5	0.35	0.42	0.08	3.0	NaN	Tollywood	Telugu
13 Mar	Arjuna	0.35	0.25	0.3	0.05	3.0	NaN	Tollywood	Telugu
...	...	...	...	...	...	...	...	...	...
19 Jan	Asha	0	NaN	0	-	1.0	NaN	Bengali	Bengali
19 Jan	Sentimentaaal	0.12	NaN	0.12	-	1.0	NaN	Bengali	Bengali
19 Jan	Hubba	0.36	NaN	0.36	-	1.0	NaN	Bengali	Bengali
12 Jan	Bijoyar Pore	0.28	NaN	0.28	-	1.0	NaN	Bengali	Bengali
12 Jan	Shri Swapankumarer Badami Hyenar Kobole	0.53	NaN	0.53	-	1.0	NaN	Bengali	Bengali

vs × 17 columns

Next steps: View recommended plots | New interactive sheet

Data Analysis

Basic Metrics

```
# Checking Shape of data
df2.shape
```

```
(1904, 17)
```

```
# Columns in data
df2.columns
```

```
Index(['Released Date', 'Movie', 'Worldwide', 'India Telugu Net',
       'India Gross', 'Overseas', 'Budget', 'Verdict', 'Industry',
       'Original Languages', 'Dubbed Languages', 'Status',
       'India Malayalam Net', 'India Hindi Net', 'India Gujarati Net',
       'India Marathi Net', 'India Bengali Net'],
      dtype='object')
```

```
# Overview of the dataset structure
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1904 entries, 0 to 1903
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Released Date        1904 non-null   object
 1   Movie                1904 non-null   object
 2   Worldwide            1904 non-null   object
 3   India Telugu Net     726 non-null    float64
 4   India Gross          1904 non-null   object
 5   Overseas             1904 non-null   object
 6   Budget               1904 non-null   float64
 7   Verdict              654 non-null    object
 8   Industry             1904 non-null   object
 9   Original Languages   1904 non-null   object
 10  Dubbed Languages     1904 non-null   object
 11  Status               1904 non-null   object
 12  India Malayalam Net  355 non-null    float64
 13  India Hindi Net      414 non-null    float64
 14  India Gujarati Net   110 non-null    float64
 15  India Marathi Net    148 non-null    float64
 16  India Bengali Net    151 non-null    float64
dtypes: float64(7), object(10)
memory usage: 253.0+ KB
```

Datatype Conversions

```
numeric_columns = df2.select_dtypes(include=['float64', 'int64']).columns
df2[numeric_columns] = df2[numeric_columns].fillna(0)
```

```
# Replace NaN values with 0 in the numeric columns
df2[numeric_columns] = df2[numeric_columns].fillna(0)

# Display the result
print(df2[['Worldwide', 'India Telugu Net', 'India Malayalam Net', 'India Hindi Net',
               'India Gujarati Net', 'India Marathi Net', 'India Bengali Net',
```

```
                'India Gross', 'Overseas', 'Budget']].head())
```

```
  Worldwide  India Telugu Net  India Malayalam Net  India Hindi Net  \
0      22.3             18.10                  0.0              0.0
1      0.25              0.20                  0.0              0.0
2      0.35              0.25                  0.0              0.0
3       0.5              0.35                  0.0              0.0
4      0.35              0.25                  0.0              0.0

   India Gujarati Net  India Marathi Net  India Bengali Net India Gross  \
0                 0.0                0.0                0.0        21.1
1                 0.0                0.0                0.0        0.23
2                 0.0                0.0                0.0         0.3
3                 0.0                0.0                0.0        0.42
4                 0.0                0.0                0.0         0.3

  Overseas  Budget
0      1.2    20.0
1     0.02     2.0
2     0.05     3.0
3     0.08     3.0
4     0.05     3.0
```

```
df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1904 entries, 0 to 1903
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Released Date        1904 non-null   object
 1   Movie                1904 non-null   object
 2   Worldwide            1904 non-null   object
 3   India Telugu Net     1904 non-null   float64
 4   India Gross          1904 non-null   object
 5   Overseas             1904 non-null   object
 6   Budget               1904 non-null   float64
 7   Verdict              654 non-null    object
 8   Industry             1904 non-null   object
 9   Original Languages   1904 non-null   object
 10  Dubbed Languages     1904 non-null   object
 11  Status               1904 non-null   object
 12  India Malayalam Net  1904 non-null   float64
 13  India Hindi Net      1904 non-null   float64
 14  India Gujarati Net   1904 non-null   float64
 15  India Marathi Net    1904 non-null   float64
 16  India Bengali Net    1904 non-null   float64
dtypes: float64(7), object(10)
memory usage: 253.0+ KB
```

Data Cleaning

```
df2.isna().sum()
```

	0
Released Date	0
Movie	0
Worldwide	0
India Telugu Net	0
India Gross	0
Overseas	0
Budget	0
Verdict	1250
Industry	0
Original Languages	0
Dubbed Languages	0
Status	0
India Malayalam Net	0
India Hindi Net	0
India Gujarati Net	0
India Marathi Net	0
India Bengali Net	0

dtype: int64

```
df2['Verdict'] = df2['Verdict'].fillna('Unknown')

# Verify if nulls are replaced
print(df2['Verdict'].isnull().sum())  # Should print 0
```

```
0
```

```
df2.isna().sum()
```

	0
Released Date	0
Movie	0
Worldwide	0
India Telugu Net	0
India Gross	0
Overseas	0
Budget	0
Verdict	0
Industry	0
Original Languages	0
Dubbed Languages	0
Status	0
India Malayalam Net	0
India Hindi Net	0
India Gujarati Net	0
India Marathi Net	0
India Bengali Net	0

dtype: int64

```
# Check the number of unique values in each column of the DataFrame 'df'
# This helps in understanding the diversity of data in each column.
df2.nunique()
```

	0
Released Date	408
Movie	1755
Worldwide	714
India Telugu Net	389
India Gross	740
Overseas	311
Budget	84
Verdict	10
Industry	6
Original Languages	6
Dubbed Languages	6
Status	1
India Malayalam Net	194
India Hindi Net	312
India Gujarati Net	73
India Marathi Net	82
India Bengali Net	76

dtype: int64

```
# Count the unique values in the 'Worldwide' column
worldwide_counts = df2['Worldwide'].value_counts()
# Display the result
print(worldwide_counts)
```

```
Worldwide
-        125
0.03      59
0.04      57
0.01      53
0.02      47
         ...
0.75       1
0.58       1
17.39      1
0.55       1
1.28       1
Name: count, Length: 714, dtype: int64
```

```
# Count the unique values in the 'India Gross' column
India_Gross_counts = df2['India Gross'].value_counts()
# Display the result
print(India_Gross_counts)
```

```
India Gross
-        63
0.04     59
0.03     58
0        57
0.01     53
         ..
12.23     1
14.53     1
```

```
41.54     1
4.1       1
2.98      1
Name: count, Length: 740, dtype: int64
```

```
# Count the unique values in the 'Overseas' column
Overseas_counts = df2['Overseas'].value_counts()
# Display the result
print(Overseas_counts)
```

```
Overseas
-       853
0       362
0.5      19
0.2      15
1        14
       ...
76        1
31        1
203       1
0.03      1
104       1
Name: count, Length: 311, dtype: int64
```

```
# Count the unique values in the 'Budget' column
Budget_counts = df2['Budget'].value_counts()
# Display the result
print(Budget_counts)
```

```
Budget
1.0       1427
15.0        37
2.0         34
10.0        31
5.0         27
           ...
820.0        1
1850.0       1
1150.0       1
135.0        1
165.0        1
Name: count, Length: 84, dtype: int64
```

```
# Count the unique values in the 'India Telugu Net' column
India_Telugu_Net_counts = df2['India Telugu Net'].value_counts()
# Display the result
print(India_Telugu_Net_counts)
```

```
India Telugu Net
0.00      1192
0.03        26
0.07        22
0.04        19
0.06        19
           ...
11.26        1
15.27        1
0.89         1
1.75         1
145.48       1
Name: count, Length: 389, dtype: int64
```

```
# Count the unique values in the 'India Malayalam  Net' column
India_Malayalam_Net_counts = df2['India Malayalam Net'].value_counts()
# Display the result
print(India_Malayalam_Net_counts)
```

```
India Malayalam Net
0.00     1557
0.04       19
0.02       14
0.01       13
0.03       13
          ...
13.81       1
1.06        1
25.63       1
49.74       1
1.22        1
Name: count, Length: 194, dtype: int64
```

```
# Count the unique values in the 'India Hindi   Net' column
India_Hindi_Net_counts = df2['India Hindi Net'].value_counts()
# Display the result
print(India_Hindi_Net_counts)
```

```
India Hindi Net
0.00     1501
0.01       14
0.04       11
0.07        9
0.03        9
          ...
20.42       1
4.00        1
45.60       1
52.15       1
0.14        1
Name: count, Length: 312, dtype: int64
```

```
# Count the unique values in the 'India Gujarati   Net' column
India_Gujarati_Net_counts = df2['India Gujarati Net'].value_counts()
# Display the result
print(India_Gujarati_Net_counts)
```

```
India Gujarati Net
0.00    1797
0.04       7
0.02       6
0.07       5
0.03       5
          ...
2.95       1
0.67       1
0.39       1
0.34       1
0.87       1
Name: count, Length: 73, dtype: int64
```

```
# Count the unique values in the 'India Marathi  Net' column
India_Marathi_Net_counts = df2['India Marathi Net'].value_counts()
# Display the result
print(India_Marathi_Net_counts)
```

```
India Marathi Net
0.00     1759
0.04        9
0.08        9
0.09        8
0.07        6
          ...
10.00       1
7.19        1
0.40        1
```

```
0.86        1
7.37        1
Name: count, Length: 82, dtype: int64
```

```
# Count the unique values in the 'India Bengali  Net' column
India_Bengali_Net_counts = df2['India Bengali Net'].value_counts()
# Display the result
print(India_Bengali_Net_counts)
```

```
India Bengali Net
0.00    1774
0.04      11
0.01      10
0.03       8
0.02       6
         ...
1.06       1
3.70       1
3.26       1
5.09       1
0.48       1
Name: count, Length: 76, dtype: int64
```

Descriptive Statistics

```
# Summary statistics for numerical columns
summary_statistics = df2.describe(include='all').T
# Display the statistics in the console
summary_statistics
```

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Released Date	1904	408	22 Nov	29	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Movie	1904	1755	Aho Vikramaarka! (Hindi)	3	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Worldwide	1904	714	-	125	NaN	NaN	NaN	NaN	NaN	NaN	NaN
India Telugu Net	1904.0	NaN	NaN	NaN	3.612474	19.44416	0.0	0.0	0.0	0.16	431.01
India Gross	1904	740	-	63	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Overseas	1904	311	-	853	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Budget	1904.0	NaN	NaN	NaN	34.758009	196.935486	1.0	1.0	1.0	1.625	3200.0
Verdict	1904	10	Unknown	1250	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Industry	1904	6	Tollywood	726	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Original Languages	1904	6	Telugu	726	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Dubbed Languages	1904	6	Hindi, Malayalam, Marathi, Gujarati, Bengali	726	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Status	1904	1	Released	1904	NaN	NaN	NaN	NaN	NaN	NaN	NaN
India Malayalam Net	1904.0	NaN	NaN	NaN	1.039279	6.38256	0.0	0.0	0.0	0.0	130.25
India Hindi Net	1904.0	NaN	NaN	NaN	7.275357	40.417719	0.0	0.0	0.0	0.0	1030.42
India Gujarati Net	1904.0	NaN	NaN	NaN	0.098944	1.098194	0.0	0.0	0.0	0.0	28.93

Next steps: Generate code with summary_statistics | View recommended plots | New interactive sheet

```
import pandas as pd

# Create a dictionary to store all statistical results
statistical = {}

# 1. Variance
statistical['Variance'] = df2[numerical_columns].var()

# 2. Covariance matrix
statistical['Covariance Matrix'] = df2[numerical_columns].cov()

# 3. Correlation table
statistical['Correlation Table'] = df2[numerical_columns].corr()

# 4. Mode
statistical['Mode Values'] = df2[numerical_columns].mode().iloc[0]

# 5. Range (max - min)
statistical['Range Values'] = df2[numerical_columns].max() - df2[numerical_columns].min()

# 6. Quartiles and IQR
```

```
# 6. Quartiles and IQR
q1_values = df2[numerical_columns].quantile(0.25)
q3_values = df2[numerical_columns].quantile(0.75)
iqr_values = q3_values - q1_values
statistical['Q1'] = q1_values
statistical['Q3'] = q3_values
statistical['IQR'] = iqr_values

# 7. Outliers detection based on IQR
outliers = {}
for col in numerical_columns:
    lower_bound = q1_values[col] - 1.5 * iqr_values[col]
    upper_bound = q3_values[col] + 1.5 * iqr_values[col]
    outliers[col] = df2[(df2[col] < lower_bound) | (df2[col] > upper_bound)].shape[0]
statistical['Outliers'] = pd.DataFrame.from_dict(outliers, orient='index', columns=['Outliers Count'])

# Combine all results into a single DataFrame for easy display
combined_stats = pd.DataFrame()

for stat_name, stat_values in statistical.items():
    if isinstance(stat_values, pd.DataFrame):
        combined_stats = pd.concat([combined_stats, stat_values], axis=1)
    else:
        combined_stats[stat_name] = stat_values

# Display the combined results
print(combined_stats)
```

```
Overseas             247701.084970       -428.467529          -229.352673
Budget                38783.585689        360.628668            -0.489398
India Telugu Net        360.628668        378.075354            -3.756343
India Malayalam Net      -0.489398         -3.756343            40.737078
India Hindi Net         862.940691        -26.295849            -7.565102
India Gujarati Net       -3.194234         -0.357620            -0.102884
India Marathi Net        -7.564518         -0.936813            -0.269514
India Bengali Net        -1.956538         -0.221648            -0.063766

                      India Hindi Net  India Gujarati Net  India Marathi Net  \
Worldwide                 4073.530714          -18.888119         -47.171287
India Gross               2371.361960           -1.655391           1.337276
Overseas                  1530.986673          -29.565094         -94.108996
Budget                     862.940691           -3.194234          -7.564518
India Telugu Net           -26.295849           -0.357620          -0.936813
India Malayalam Net         -7.565102           -0.102884          -0.269514
India Hindi Net           1633.592010           -0.720230          -1.886699
India Gujarati Net          -0.720230            1.206029          -0.025659
India Marathi Net           -1.886699           -0.025659           7.640776
India Bengali Net           -0.446389           -0.006071          -0.015903

                      ...  India Hindi Net  India Gujarati Net  \
Worldwide             ...         0.078784           -0.013565
India Gross           ...         0.578614           -0.014991
Overseas              ...         0.022348           -0.016143
Budget                ...         0.108414           -0.014769
India Telugu Net      ...        -0.033460           -0.016748
```

```
India Marathi Net            1.000000          -0.010275         0.00
India Bengali Net           -0.010275           1.000000         0.00

                     Range Values    Q1      Q3     IQR  Outliers Count
Worldwide                17380.00  0.08  27.830  27.750             305
India Gross               1416.90  0.08  18.350  18.270             292
Overseas                 16902.50  0.00  16.500  16.500             171
Budget                    3199.00  1.00   1.625   0.625             441
India Telugu Net           431.01  0.00   0.160   0.160             384
India Malayalam Net        130.25  0.00   0.000   0.000             347
India Hindi Net           1030.42  0.00   0.000   0.000             403
India Gujarati Net          28.93  0.00   0.000   0.000             107
India Marathi Net           76.28  0.00   0.000   0.000             145
India Bengali Net           13.18  0.00   0.000   0.000             130

[10 rows x 27 columns]
```

```
combined_stats
```

	Variance	Worldwide	India Gross	Overseas	Budget	India Telugu Net	Indi Malayala Ne
Worldwide	1.532359e+06	1.532359e+06	33961.543591	1.959754e+06	175566.928088	728.341400	-15.53742
India Gross	9.952577e+03	3.396154e+04	9952.576549	2.991727e+04	6797.454386	744.858152	75.41928
Overseas	1.627048e+06	1.959754e+06	29917.271233	1.627048e+06	247701.084970	-428.467529	-229.35267
Budget	3.878359e+04	1.755669e+05	6797.454386	2.477011e+05	38783.585689	360.628668	-0.48939
India Telugu Net	3.780754e+02	7.283414e+02	744.858152	-4.284675e+02	360.628668	378.075354	-3.75634
India Malayalam Net	4.073708e+01	-1.553743e+01	75.419281	-2.293527e+02	-0.489398	-3.756343	40.73707
India Hindi Net	1.633592e+03	4.073531e+03	2371.361960	1.530987e+03	862.940691	-26.295849	-7.56510
India Gujarati Net	1.206029e+00	-1.888812e+01	-1.655391	-2.956509e+01	-3.194234	-0.357620	-0.10288
India Marathi Net	7.640776e+00	-4.717129e+01	1.337276	-9.410900e+01	-7.564518	-0.936813	-0.26951
India Bengali Net	3.135424e-01	-1.364592e+01	-1.665254	-2.103662e+01	-1.956538	-0.221648	-0.06376

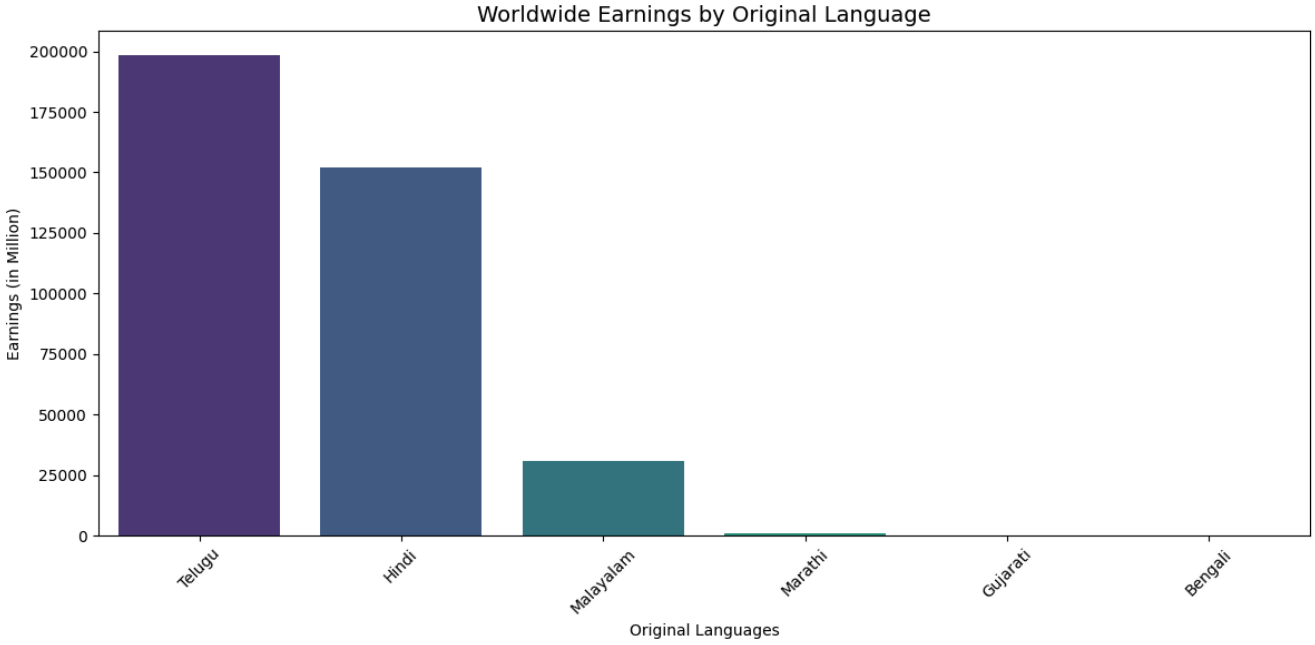
10 rows × 27 columns

Data Visualization

```
#  1. Worldwide Earnings by Original Language
language_earnings = df2.groupby("Original Languages").sum()[['Worldwide']]
language_earnings_sorted = language_earnings.sort_values(by="Worldwide", ascending=False)

# Plot: Contribution to Worldwide Earnings by Language
plt.figure(figsize=(12, 6))
sns.barplot(x=language_earnings_sorted.index, y=language_earnings_sorted["Worldwide"], palette="viridis")
plt.title("Worldwide Earnings by Original Language", fontsize=14)
```

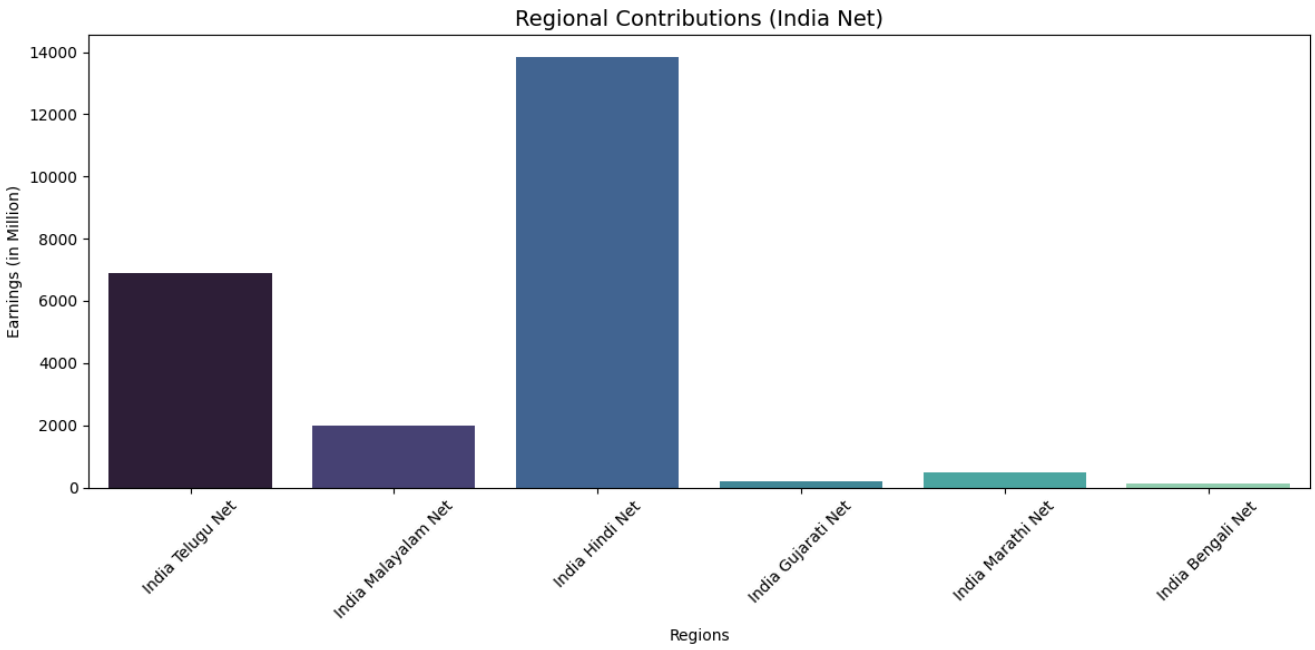
```
plt.xticks(rotation=45)
plt.ylabel("Earnings (in Million)")
plt.xlabel("Original Languages")
plt.tight_layout()
plt.show()
```

```
<ipython-input-32-ea098e244e0c>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

  sns.barplot(x=language_earnings_sorted.index, y=language_earnings_sorted["Worldwide"], palette="viric
```

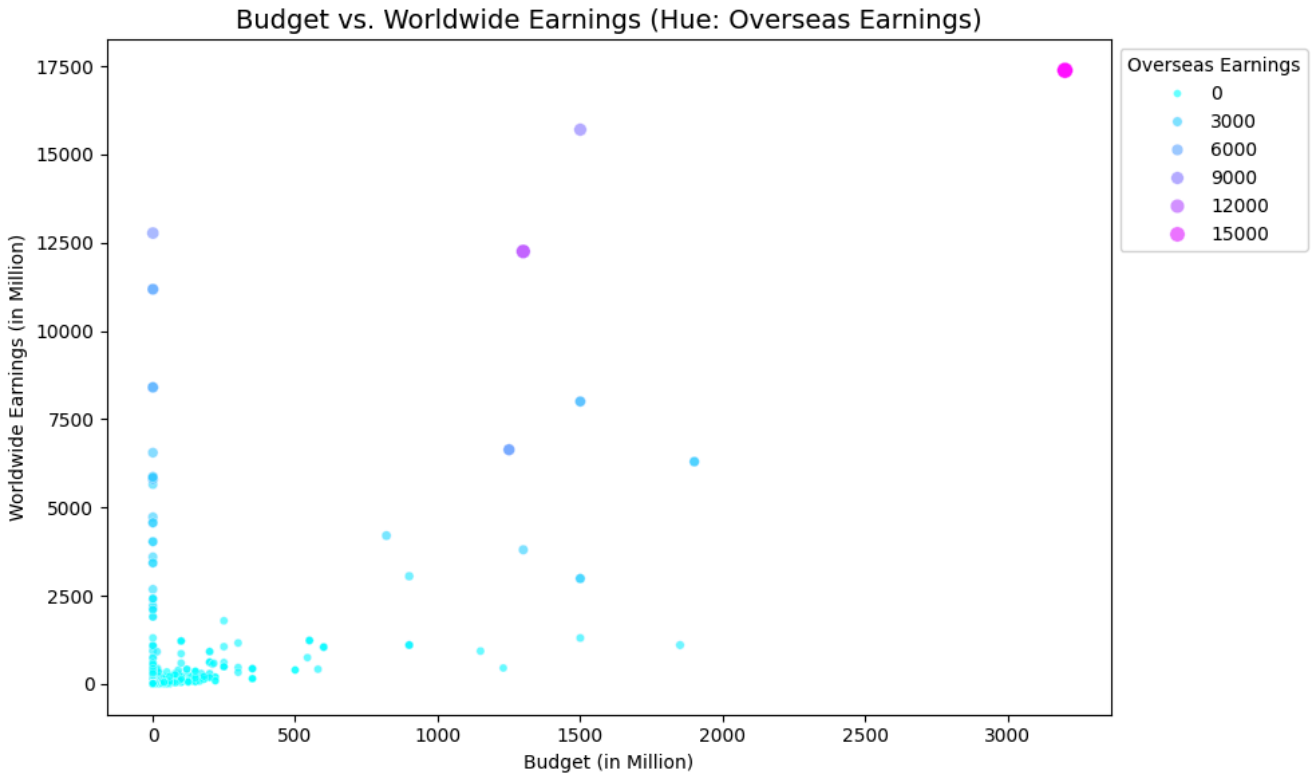



```
# 2. Regional Contributions (India Net)

regional_totals = df2[regional_columns].sum()

# Plot: Regional earnings contributions
plt.figure(figsize=(12, 6))
sns.barplot(x=regional_columns, y=regional_totals, palette="mako")
plt.title("Regional Contributions (India Net)", fontsize=14)
plt.xticks(rotation=45)
plt.ylabel("Earnings (in Million)")
plt.xlabel("Regions")
plt.tight_layout()
plt.show()
```

```
<ipython-input-34-4982b47a3032>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x`

  sns.barplot(x=regional_columns, y=regional_totals, palette="mako")
```



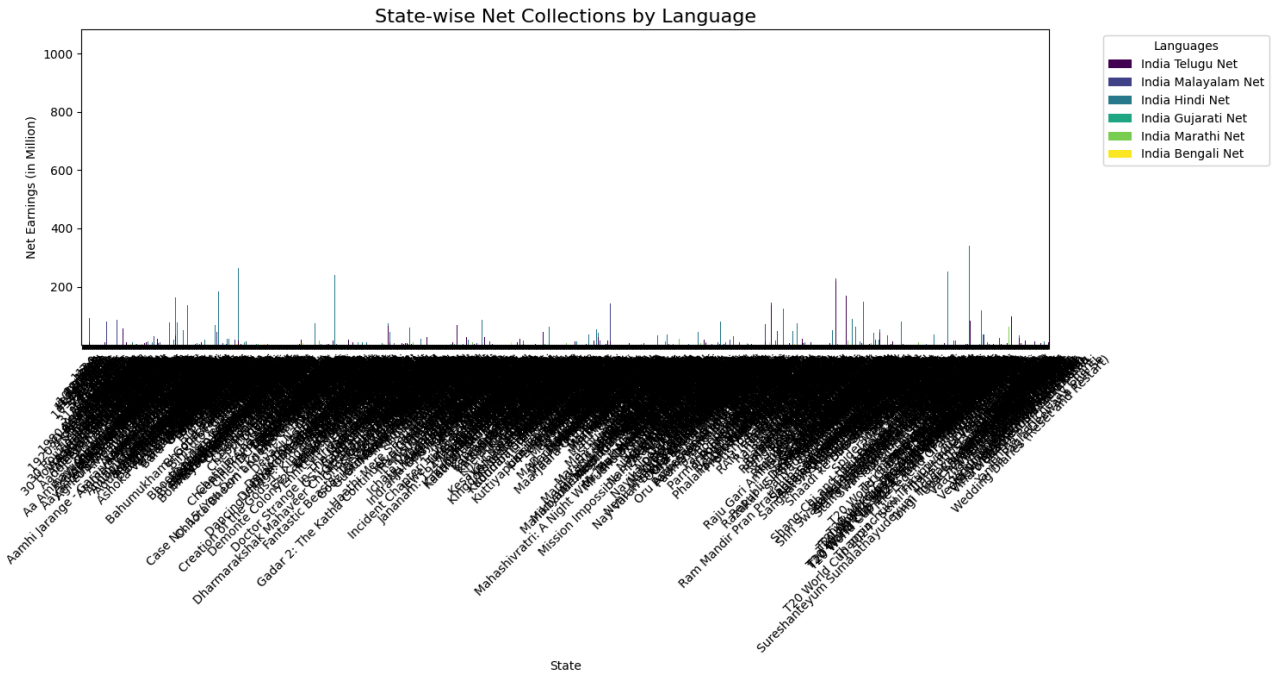

```
# 3. Budget vs. Worldwide Earnings (Hue: Overseas)
plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=df2, x="Budget", y="Worldwide", hue="Overseas", size="Overseas",
    palette="cool", alpha=0.6
)
plt.title("Budget vs. Worldwide Earnings (Hue: Overseas Earnings)", fontsize=14)
plt.xlabel("Budget (in Million)")
plt.ylabel("Worldwide Earnings (in Million)")
plt.legend(title="Overseas Earnings", loc="upper left", bbox_to_anchor=(1, 1))
plt.tight_layout()
plt.show()
```



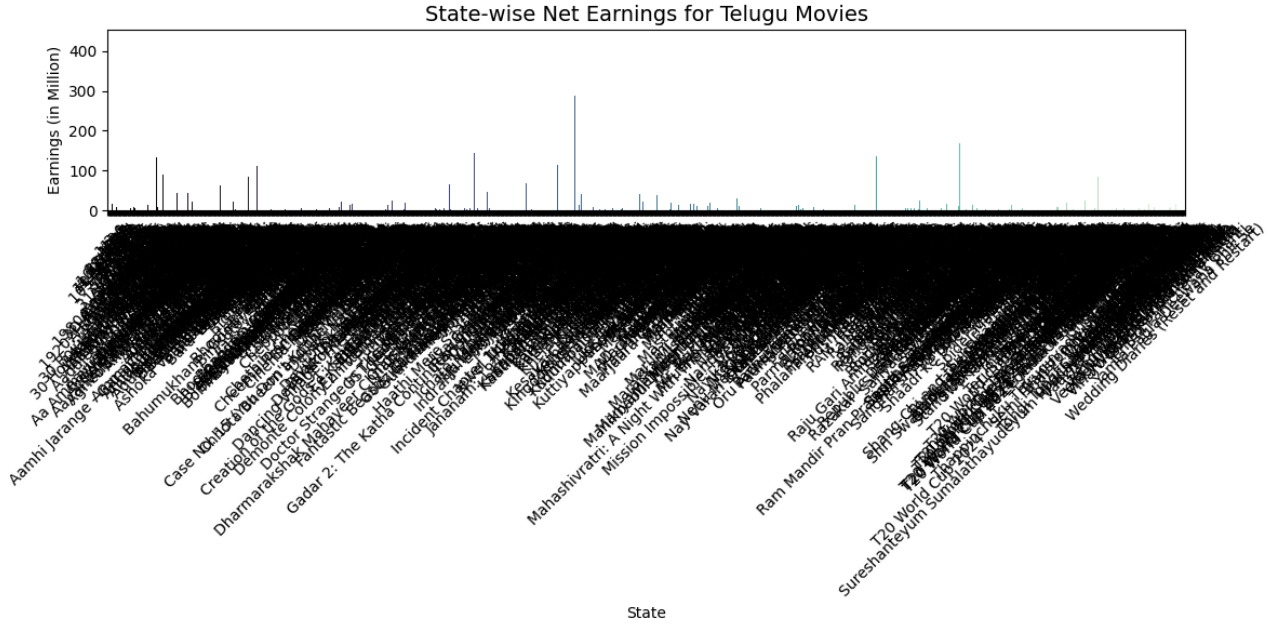

```
# Group and sum by state for regional nets
regional_data = df2[[
    'Movie', 'India Telugu Net', 'India Malayalam Net',
    'India Hindi Net', 'India Gujarati Net',
    'India Marathi Net', 'India Bengali Net'
]]

# Fill missing values with 0
regional_data.fillna(0, inplace=True)

# Aggregate by state
statewise_totals = regional_data.groupby('Movie').sum()

# Plot: State-wise Telugu Net, Malayalam Net, etc.
plt.figure(figsize=(15, 8))
statewise_totals.plot(kind='bar', stacked=True, figsize=(15, 8), colormap='viridis')
plt.title("State-wise Net Collections by Language", fontsize=16)
plt.ylabel("Net Earnings (in Million)")
plt.xlabel("State")
plt.xticks(rotation=45)
plt.legend(title="Languages", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.tight_layout()
plt.show()

# Individual plots for each language
languages = [
    'India Telugu Net', 'India Malayalam Net',
    'India Hindi Net', 'India Gujarati Net',
    'India Marathi Net', 'India Bengali Net'
]

for language in languages:
    plt.figure(figsize=(12, 6))
    sns.barplot(x=statewise_totals.index, y=statewise_totals[language], palette="mako")
    plt.title(f"State-wise Net Earnings for {language.split()[-2]} Movies", fontsize=14)
    plt.ylabel("Earnings (in Million)")
```

```
        plt.xlabel("Movies")
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
```

```
<ipython-input-38-2455f7609db3>:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/index
  regional_data.fillna(0, inplace=True)
<Figure size 1500x800 with 0 Axes>
```




```
<ipython-input-38-2455f7609db3>:34: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `

  sns.barplot(x=statewise_totals.index, y=statewise_totals[language], palette="mako")
```




```
<ipython-input-38-2455f7609db3>:34: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `

  sns.barplot(x=statewise_totals.index, y=statewise_totals[language], palette="mako")
```