

Film Patron satisfaction EDA

Film on the Rocks, hosted at the iconic Red Rocks Amphitheatre in Colorado, offers a unique cinematic experience with classic films, live entertainment, and breathtaking views. Co-promoted by the Denver Film Society and the City and County of Denver, the series is supported by corporate sponsorships, allowing for affordable ticket prices. While the venue is exceptional, challenges like accessibility and weather dependence exist. The promoters aim to enhance patron satisfaction and increase attendance. To achieve this, they conducted surveys during a recent season to understand the demographics, satisfaction levels, and effective media outlets for their audience, facilitating better-targeted marketing efforts and potential growth.

In [1]:

```
#importing packages
import pandas as pd
```

In [2]:

```
# Loading the survey data
data = pd.read_csv("C:\\Users\\sujoydutta\\Desktop\\Data analysis\\Datasets for ML\\Misc types\\Films.csv")
data.head()
```

Out[2]:

	rowstate	Movie	Gender	Marital_Status	Sinage	Parking	Clean	Overall	Age	Income	Hear_About
0	0	Ferris Buellers Day Off	Female	Married	2.0	2.0	2.0	2.0	3.0	1.0	5
1	0	Ferris Buellers Day Off	Female	Single	1.0	1.0	1.0	1.0	2.0	1.0	5
2	0	Ferris Buellers Day Off	Male	Married	2.0	4.0	3.0	2.0	4.0	1.0	5
3	0	Ferris Buellers Day Off	Female	Married	1.0	3.0	2.0	2.0	4.0	1.0	5
4	0	Ferris Buellers Day Off	Female	Married	1.0	1.0	1.0	1.0	3.0	3.0	1

In [7]:

```
#Calculating overall customer satisfaction
likert_columns = ["Sinage", "Parking", "Clean"]
data['Overall_Satisfaction'] = data[likert_columns].mean(axis=1)
```

In [8]:

```
# Printing the results
print("Overall Customer Satisfaction:")
print(data['Overall_Satisfaction'].mean())
```

Overall Customer Satisfaction:
1.7740628166160084

In [9]:

```
# Analyze factors linked to satisfaction
correlation_matrix = data[likert_columns].corr()
```

In [10]:

```
print("Factors Linked to Satisfaction:")
print(correlation_matrix)
```

Factors Linked to Satisfaction:

	Sinage	Parking	Clean
Sinage	1.000000	0.470322	0.349163
Parking	0.470322	1.000000	0.444368
Clean	0.349163	0.444368	1.000000

In [14]:

```
# Recoding values
data['Gender'] = data['Gender'].replace({'1': 'Male', '2': 'Female'})
data['Marital_Status'] = data['Marital_Status'].replace({'1': 'Married', '2': 'Single', 'Slnge': 'Single'})
```

In [16]:

```
# Analyzing the demographic profile

# Gender
gender_profile = data['Gender'].value_counts()
# Marital Status
marital_status_profile = data['Marital_Status'].value_counts()
# Age
age_profile = data['Age'].value_counts()
# Income
income_profile = data['Income'].value_counts()
```

In [17]:

```
# Printing the results
print("Demographic Profile:")
print("Gender Profile:")
print(gender_profile)
print("\nMarital Status Profile:")
print(marital_status_profile)
print("\nAge Profile:")
print(age_profile)
print("\nIncome Profile:")
print(income_profile)
```

Demographic Profile:
Gender Profile:
Female 213
Male 117
Name: Gender, dtype: int64

Marital Status Profile:
Single 228
Married 100
Name: Marital_Status, dtype: int64

Age Profile:
2.0 175
3.0 117
1.0 26
4.0 10
Name: Age, dtype: int64

Income Profile:
1.0 142
3.0 90
2.0 82
Name: Income, dtype: int64

In [23]:

```
# Extracting the primary source from 'Hear About' column
data['Primary_Source'] = data['Hear_About'].str.split(',').str[0]
data['Primary_Source']
```

Out[23]:

0	5
1	5
2	5
3	5
4	1
..	
325	1
326	5
327	5
328	3
329	5

Name: Primary_Source, Length: 330, dtype: object

In [31]:

```
#Calculating the frequency of each source
source_counts = data['Primary_Source'].value_counts().sort_index()
```

In [32]:

```
#percentage counts for each source
percentage_counts = source_counts / source_counts.sum() * 100
```

In [33]:

```
# Determining effective media outlets
source_mapping = {
    1: 'Television',
    2: 'Newspaper',
    3: 'Radio',
    4: 'Website',
    5: 'Word of Mouth'
}
```

In [36]:

```
# Initializing counts and percentages
source_counts = {source_name: 0 for source_name in source_mapping.values()}
percentage_counts = {source_name: 0 for source_name in source_mapping.values()}
```

In [37]:

```
# Counting and calculating percentages for each source
for code, source_name in source_mapping.items():
    count = data['Primary_Source'].apply(lambda x: x.split(',') if isinstance(x, str) else []).apply(lambda x:
percentage = (count / len(data) * 100) if len(data) > 0 else 0
    source_counts[source_name] = count
    percentage_counts[source_name] = percentage
```

In [38]:

```
# Printing the results
print("How Patrons Heard About Film on the Rocks:")
for source_name in source_mapping.values():
    count = source_counts[source_name]
    percentage = percentage_counts[source_name]
    print(f"{source_name}: Count={count}, Percentage={percentage:.2f}%")
```

How Patrons Heard About Film on the Rocks:
Television: Count=23, Percentage=6.97%
Newspaper: Count=14, Percentage=4.24%
Radio: Count=17, Percentage=5.15%
Website:Count=42, Percentage=12.73%
Word of Mouth: Count=227, Percentage=68.79%