CINEMA TICKETS

- The Cinema Tickets dataset, covering eight months in 2018, offers sales history and screening details from various cinemas. Ideal for predictive modeling, it supports forecasting, screening optimization, and ROI improvement. With anonymized locations, it aids decisions on cast, crew, and project planning.
- Emphasizing time series analysis, it provides insights for Cinema Clustering, sales forecasts, and movie genre recommendations. Evolving with additional movie data, it enhances strategic decision-making in the cinema industry.
- DOWNLOAD THE DATASET HERE

In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
df=pd.read_csv('cinemaTicket_Ref.csv')
df
```

Out[4]:

	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price	ticket_use	capacity	date	mo
0	1492	304	3900000	26	0	4	4.26	150000.0	26	610.328638	2018-05-05	
1	1492	352	3360000	42	0	5	8.08	80000.0	42	519.801980	2018-05-05	
2	1492	489	2560000	32	0	4	20.00	80000.0	32	160.000000	2018-05-05	
3	1492	429	1200000	12	0	1	11.01	100000.0	12	108.991826	2018-05-05	
4	1492	524	1200000	15	0	3	16.67	80000.0	15	89.982004	2018-05-05	
...	...	...	...	...	...	...	...	...	...	...	...	
142519	1569	495	1320000	22	0	2	3.86	60000.0	22	569.948187	2018-11-04	
142520	1569	474	1200000	15	0	1	65.22	80000.0	15	22.999080	2018-11-04	
142521	1569	524	1060000	8	0	3	9.20	132500.0	8	86.956522	2018-11-04	
142522	1569	529	600000	5	0	2	5.00	120000.0	5	100.000000	2018-11-04	
142523	1569	486	250000	5	0	1	1.79	50000.0	5	279.329609	2018-11-04	

142524 rows × 14 columns

In [5]:
```
df.head()
```

Out[5]:

	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price	ticket_use	capacity	date	month	c
0	1492	304	3900000	26	0	4	4.26	150000.0	26	610.328638	2018-05-05	5	
1	1492	352	3360000	42	0	5	8.08	80000.0	42	519.801980	2018-05-05	5	
2	1492	489	2560000	32	0	4	20.00	80000.0	32	160.000000	2018-05-05	5	
3	1492	429	1200000	12	0	1	11.01	100000.0	12	108.991826	2018-05-05	5	
4	1492	524	1200000	15	0	3	16.67	80000.0	15	89.982004	2018-05-05	5	

In [6]:
```
df.tail()
```

Out[6]:

	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price	ticket_use	capacity	date	mo
142519	1569	495	1320000	22	0	2	3.86	60000.0	22	569.948187	2018-11-04	
142520	1569	474	1200000	15	0	1	65.22	80000.0	15	22.999080	2018-11-04	
142521	1569	524	1060000	8	0	3	9.20	132500.0	8	86.956522	2018-11-04	
142522	1569	529	600000	5	0	2	5.00	120000.0	5	100.000000	2018-11-04	
142523	1569	486	250000	5	0	1	1.79	50000.0	5	279.329609	2018-11-04	

In [7]:
```
df.shape
```

Out[7]:
```
(142524, 14)
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['film_code', 'cinema_code', 'total_sales', 'tickets_sold',
       'tickets_out', 'show_time', 'occu_perc', 'ticket_price', 'ticket_use',
       'capacity', 'date', 'month', 'quarter', 'day'],
      dtype='object')
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 142524 entries, 0 to 142523
Data columns (total 14 columns):
 #   Column        Non-Null Count   Dtype  
---  ------        --------------   -----  
 0   film_code     142524 non-null  int64  
 1   cinema_code   142524 non-null  int64  
 2   total_sales   142524 non-null  int64  
 3   tickets_sold  142524 non-null  int64  
 4   tickets_out   142524 non-null  int64  
 5   show_time     142524 non-null  int64  
 6   occu_perc     142399 non-null  float64
 7   ticket_price  142524 non-null  float64
 8   ticket_use    142524 non-null  int64  
 9   capacity      142399 non-null  float64
 10  date          142524 non-null  object 
 11  month         142524 non-null  int64  
 12  quarter       142524 non-null  int64  
 13  day           142524 non-null  int64  
dtypes: float64(3), int64(10), object(1)
memory usage: 15.2+ MB
```

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
film_code         0
cinema_code       0
total_sales       0
tickets_sold      0
tickets_out       0
show_time         0
occu_perc       125
ticket_price      0
ticket_use        0
capacity        125
date              0
month             0
quarter           0
day               0
dtype: int64
```

In [11]:
```
df.dropna(inplace=True)
```

In [12]:
```
pd.isnull(df).sum()
```

Out[12]:
```
film_code       0
cinema_code     0
total_sales     0
tickets_sold    0
tickets_out     0
show_time       0
occu_perc       0
ticket_price    0
ticket_use      0
capacity        0
date            0
month           0
quarter         0
day             0
dtype: int64
```

In [13]:
```
print(df.duplicated().sum())
```

```
104
```

In [14]:
```
df.drop_duplicates(inplace=True)
```

In [15]:
```
print(df.duplicated().sum())
```

```
0
```

In [16]:
```
#let's check how many unique number of value present in our datasets
df['date'].nunique()
```

Out[16]:
```
234
```

In [17]:
```
df['date'].unique()
```

Out[17]:
```
array(['2018-05-05', '2018-05-04', '2018-05-03', '2018-05-02',
       '2018-05-06', '2018-08-04', '2018-08-03', '2018-08-02',
       '2018-08-01', '2018-07-31', '2018-07-30', '2018-07-29',
       '2018-07-28', '2018-07-27', '2018-07-26', '2018-07-25',
       '2018-07-24', '2018-07-23', '2018-07-22', '2018-07-20',
       '2018-07-19', '2018-07-18', '2018-07-17', '2018-07-16',
       '2018-08-12', '2018-07-15', '2018-07-14', '2018-07-13',
       '2018-07-12', '2018-07-11', '2018-07-10', '2018-07-08',
       '2018-07-07', '2018-07-06', '2018-07-05', '2018-07-04',
       '2018-07-03', '2018-07-02', '2018-07-01', '2018-06-30',
       '2018-06-29', '2018-06-28', '2018-06-27', '2018-06-26',
       '2018-08-10', '2018-06-25', '2018-06-24', '2018-06-23',
       '2018-06-22', '2018-06-21', '2018-06-20', '2018-06-19',
       '2018-06-18', '2018-06-17', '2018-06-16', '2018-06-15',
       '2018-06-14', '2018-06-13', '2018-06-12', '2018-06-11',
       '2018-06-10', '2018-06-09', '2018-06-08', '2018-06-07',
       '2018-06-03', '2018-06-02', '2018-06-01', '2018-05-31',
       '2018-05-30', '2018-05-29', '2018-05-28', '2018-05-27',
       '2018-08-07', '2018-05-26', '2018-05-25', '2018-05-24',
       '2018-05-23', '2018-05-22', '2018-05-21', '2018-05-20',
       '2018-05-19', '2018-05-18', '2018-05-17', '2018-08-06',
       '2018-05-16', '2018-05-15', '2018-05-14', '2018-05-13',
       '2018-05-12', '2018-05-11', '2018-05-10', '2018-05-09',
       '2018-05-08', '2018-05-07', '2018-08-05', '2018-09-18',
       '2018-09-17', '2018-09-16', '2018-09-15', '2018-09-14',
       '2018-09-13', '2018-09-12', '2018-09-11', '2018-09-10',
       '2018-09-09', '2018-09-08', '2018-09-07', '2018-09-06',
       '2018-09-05', '2018-09-30', '2018-09-29', '2018-09-28',
       '2018-09-27', '2018-09-26', '2018-09-25', '2018-09-24',
       '2018-09-23', '2018-10-02', '2018-07-21', '2018-08-08',
       '2018-10-24', '2018-10-23', '2018-10-21', '2018-10-20',
       '2018-10-19', '2018-10-18', '2018-10-17', '2018-10-16',
       '2018-10-15', '2018-11-03', '2018-10-14', '2018-10-13',
       '2018-10-12', '2018-10-11', '2018-10-10', '2018-10-09',
       '2018-10-08', '2018-10-07', '2018-10-06', '2018-10-05',
       '2018-11-02', '2018-10-04', '2018-10-03', '2018-10-01',
       '2018-10-31', '2018-10-29', '2018-09-04', '2018-09-03',
       '2018-09-02', '2018-09-01', '2018-08-31', '2018-08-30',
       '2018-08-29', '2018-08-28', '2018-08-27', '2018-08-26',
       '2018-10-28', '2018-08-25', '2018-08-24', '2018-08-23',
       '2018-08-22', '2018-08-21', '2018-08-20', '2018-08-19',
       '2018-08-18', '2018-08-17', '2018-08-16', '2018-10-27',
       '2018-08-15', '2018-08-14', '2018-08-13', '2018-08-11',
       '2018-08-09', '2018-10-26', '2018-10-25', '2018-11-04',
       '2018-07-09', '2018-06-05', '2018-03-16', '2018-03-15',
       '2018-03-14', '2018-03-17', '2018-05-01', '2018-04-30',
       '2018-04-29', '2018-04-28', '2018-04-27', '2018-04-26',
       '2018-04-25', '2018-04-24', '2018-04-23', '2018-04-22',
       '2018-04-21', '2018-04-20', '2018-04-19', '2018-04-18',
       '2018-04-17', '2018-04-16', '2018-04-15', '2018-04-14',
       '2018-04-13', '2018-04-12', '2018-04-11', '2018-04-10',
       '2018-04-09', '2018-04-08', '2018-04-07', '2018-04-06',
       '2018-04-05', '2018-04-04', '2018-04-03', '2018-04-02',
       '2018-04-01', '2018-03-31', '2018-03-30', '2018-03-29',
       '2018-03-28', '2018-03-27', '2018-03-26', '2018-03-25',
       '2018-03-24', '2018-03-23', '2018-03-22', '2018-03-21',
       '2018-03-20', '2018-03-19', '2018-03-18', '2018-09-22',
       '2018-11-01', '2018-10-30', '2018-10-22', '2018-02-23',
       '2018-02-21', '2018-09-21'], dtype=object)
```

In [18]:
```
df.describe()
```

Out[18]:

	film_code	cinema_code	total_sales	tickets_sold	tickets_out	show_time	occu_perc	ticket_price	ticket_u
count	142295.000000	142295.000000	1.422950e+05	142295.000000	142295.000000	142295.000000	142295.000000	142295.000000	142295.0000
mean	1519.014871	320.383295	1.233833e+07	140.097944	0.237493	3.933357	19.955645	81239.000455	139.8604
std	36.185012	159.655445	3.062706e+07	279.704392	2.925276	3.056759	22.648393	33239.397517	279.5101
min	1471.000000	32.000000	2.000000e+04	1.000000	0.000000	1.000000	0.000000	483.870968	-219.0000
25%	1485.000000	181.000000	1.270000e+06	18.000000	0.000000	2.000000	3.750000	60000.000000	18.0000
50%	1498.000000	324.000000	3.720000e+06	50.000000	0.000000	3.000000	10.340000	79448.266667	50.0000
75%	1556.000000	474.000000	1.110000e+07	143.000000	0.000000	5.000000	28.210000	100000.000000	143.0000
max	1589.000000	566.000000	1.262820e+09	8499.000000	311.000000	60.000000	147.500000	700000.000000	8499.0000

In [19]:
```
df.shape
```

Out[19]:
```
(142295, 14)
```

In [20]:
```
df.columns
```

Out[20]:
```
Index(['film_code', 'cinema_code', 'total_sales', 'tickets_sold',
       'tickets_out', 'show_time', 'occu_perc', 'ticket_price', 'ticket_use',
       'capacity', 'date', 'month', 'quarter', 'day'],
      dtype='object')
```

In [21]:
```
df['date']=pd.to_datetime(df['date'])
df['Year']=df['date'].dt.year
```

In [22]:
```
time_sn=df[["date","total_sales"]]
time_sn.head()
```

Out[22]:

	date	total_sales
0	2018-05-05	3900000
1	2018-05-05	3360000
2	2018-05-05	2560000
3	2018-05-05	1200000
4	2018-05-05	1200000

In [23]:
```
time_sn = time_sn.sort_values('date')
time_sn.head()
```

Out[23]:

	date	total_sales
46143	2018-02-21	32030000
46142	2018-02-23	180000
120952	2018-03-14	240000
104023	2018-03-14	7050000
104024	2018-03-14	6900000

In [24]:
```
time_sn = time_sn.reset_index(drop=True)
time_sn.head()
```

Out[24]:

	date	total_sales
0	2018-02-21	32030000
1	2018-02-23	180000
2	2018-03-14	240000
3	2018-03-14	7050000
4	2018-03-14	6900000

In [25]:
```
time_sn = time_sn.set_index("date")
```

In [26]:
```
time_sn.tail()
```

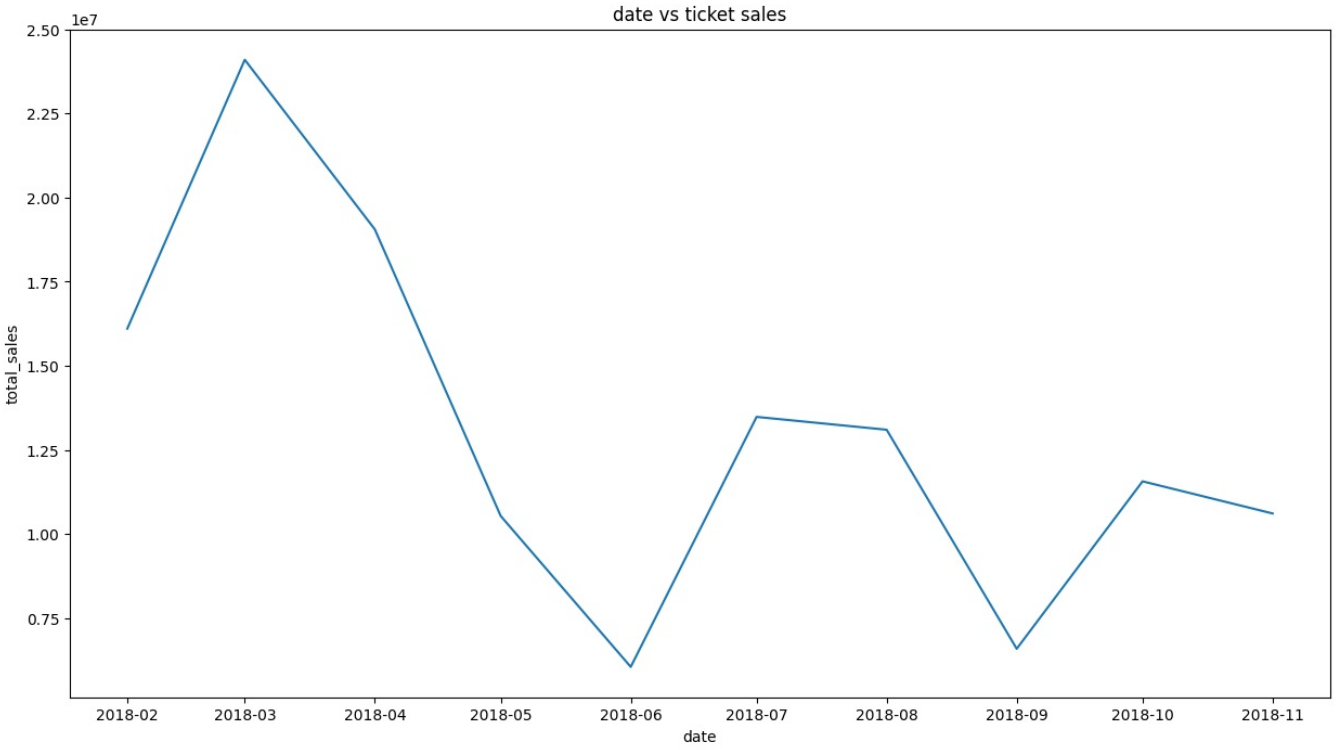
Out[26]:

date	total_sales
2018-11-04	770000
2018-11-04	1040000
2018-11-04	1520000
2018-11-04	3880000
2018-11-04	250000

In [27]:
```
time_sn_x=time_sn["total_sales"].resample('MS').mean()
```

In [28]:
```
plt.figure(figsize=(15,8))
sns.lineplot(time_sn_x)
plt.title("date vs ticket sales")
```

Out[28]:
```
Text(0.5, 1.0, 'date vs ticket sales')
```



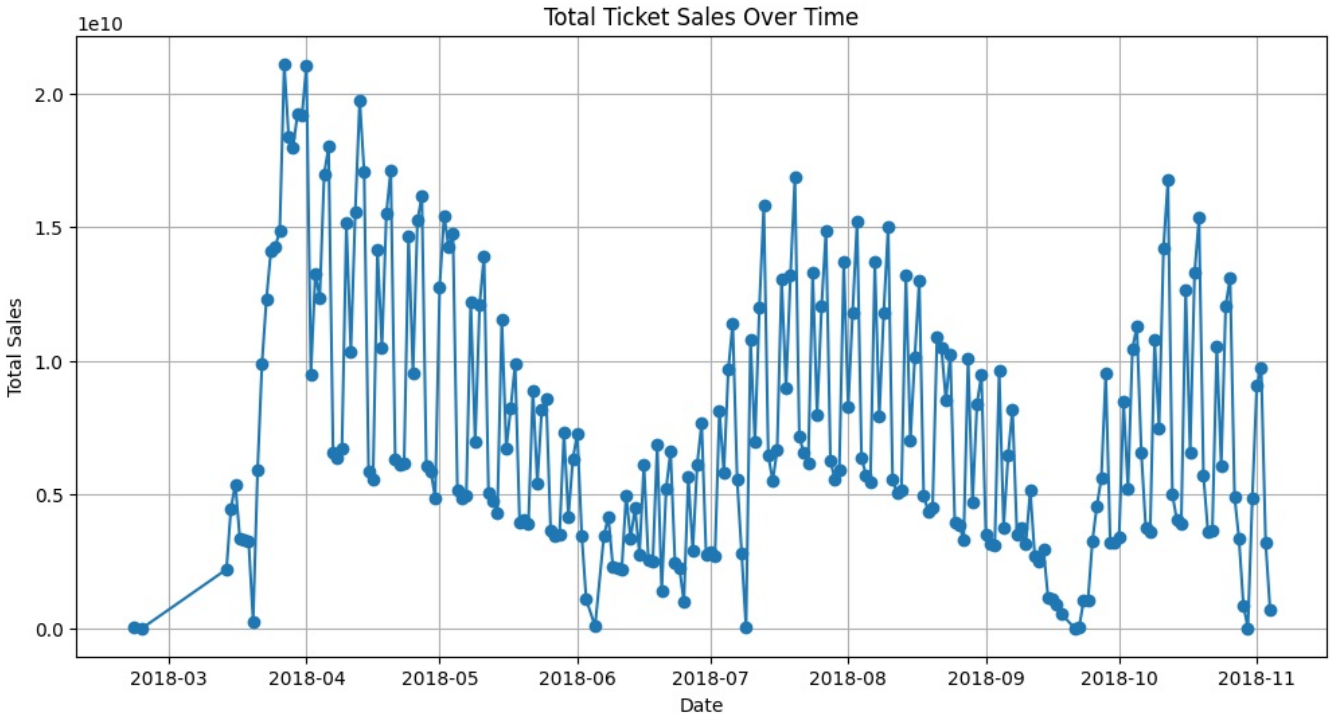
In [29]:
```
# Aggregate total sales by date
daily_sales=df.groupby("date")["total_sales"].sum().reset_index()
daily_sales
```

Out[29]:

	date	total_sales
0	2018-02-21	32030000
1	2018-02-23	180000
2	2018-03-14	2202300484
3	2018-03-15	4479008974
4	2018-03-16	5386645987
...	...	...
229	2018-10-31	4867668000
230	2018-11-01	9060334998
231	2018-11-02	9740225000
232	2018-11-03	3208490000
233	2018-11-04	682985000

234 rows × 2 columns

In [30]:
```
# Plot the total sales over time
plt.figure(figsize=(12, 6))
plt.plot(daily_sales['date'], daily_sales['total_sales'], marker='o', linestyle='-')
plt.title('Total Ticket Sales Over Time')
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.grid(True)
plt.show()
```



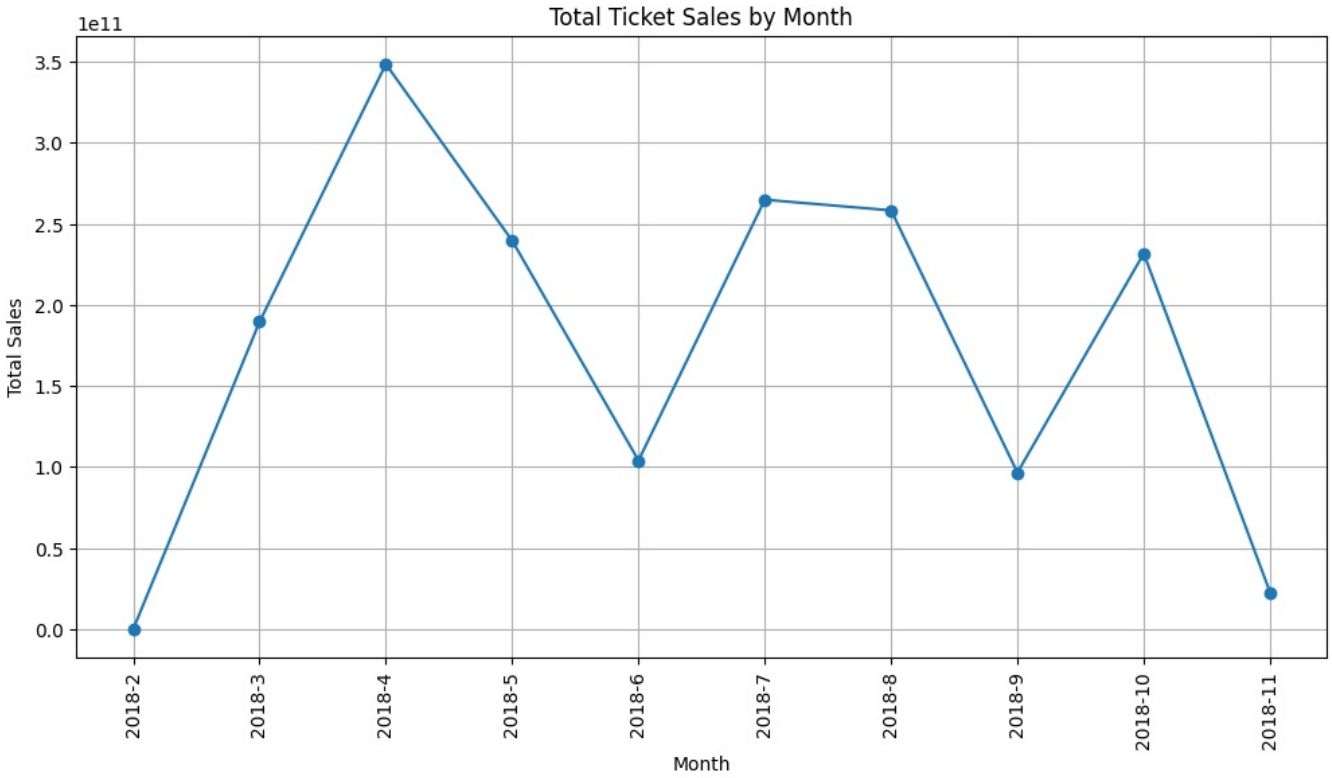
In [31]:
```
# Aggregate total sales by year and month
monthly_sales = df.groupby(['Year','month'])['total_sales'].sum().reset_index()
monthly_sales
```

Out[31]:

	Year	month	total_sales
0	2018	2	32210000
1	2018	3	189502975738
2	2018	4	348510864007
3	2018	5	239585133317
4	2018	6	104001473893
5	2018	7	264855222688
6	2018	8	258275380764
7	2018	9	96566384964
8	2018	10	231660299820
9	2018	11	22692034998

In [32]:
```
# Create a 'year_month' column for better visualization
monthly_sales['year_month'] = monthly_sales['Year'].astype(str) + '-' + monthly_sales['month'].astype(str)
```

In [33]:
```
# Plot the monthly total sales
plt.figure(figsize=(12, 6))
plt.plot(monthly_sales['year_month'], monthly_sales['total_sales'], marker='o', linestyle='-')
plt.title('Total Ticket Sales by Month')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.xticks(rotation=90)
plt.grid(True)
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: