In [4]:
```
import pandas as pd
import numpy as np
import matplotlib as mpl
```

In [6]:
```
lobster = pd.read_csv('lobster24.csv')
```

In [10]:
```
#Question A
lobster.head()
```

Out[10]:

	Date	Day_of_Week	Total_Visitors	Arcade_Visitors	Total_Purchases	Arcade_
0	5/27/2024	Monday	1060	267	22575.71	
1	5/28/2024	Tuesday	1494	552	17142.52	
2	5/29/2024	Wednesday	1330	447	10229.99	
3	5/30/2024	Thursday	1295	442	6442.65	
4	5/31/2024	Friday	1838	256	28409.06	

Question B - there are five rows

In [12]:
```
#Question C shape attribute
print(lobster.shape)
```

```
(99, 15)
```

Question C -there are 99 rows and 15 columns

In [14]:
```
#Question D
lobster.describe()
```

Out[14]:

	Total_Visitors	Arcade_Visitors	Total_Purchases	Arcade_Revenue	Total_Labor_
count	99.000000	99.000000	99.000000	99.000000	99.0
mean	1399.737374	408.050505	15712.011818	3120.240808	309.0
std	651.157801	182.460137	5737.163102	1744.706699	107.
min	221.000000	51.000000	5271.130000	568.870000	115.5
25%	924.500000	268.000000	11069.275000	1638.020000	233.3
50%	1380.000000	426.000000	16028.240000	3047.170000	311.3
75%	1924.500000	559.500000	19003.970000	4511.465000	393.7
max	2478.000000	698.000000	28599.960000	7088.790000	498.3

In [16]:
```
#Question D
lobster.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 15 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Date                     99 non-null     object 
 1   Day_of_Week              99 non-null     object 
 2   Total_Visitors           99 non-null     int64  
 3   Arcade_Visitors          99 non-null     int64  
 4   Total_Purchases          99 non-null     float64
 5   Arcade_Revenue           99 non-null     float64
 6   Total_Labor_Hours        99 non-null     float64
 7   Weather_Type             99 non-null     object 
 8   Special_Events           21 non-null     object 
 9   Customer_Complaints      99 non-null     int64  
 10  Passholder_Percentage    99 non-null     float64
 11  International_Visitors   99 non-null     int64  
 12  High_Temperature         96 non-null     float64
 13  General_Weather          99 non-null     object 
 14  Precipitation            99 non-null     int64  
dtypes: float64(5), int64(5), object(5)
memory usage: 11.7+ KB
```

Question D - Categorical: Day_of_Week, Weather_Type, Special_Events, General_Weather, Date Numeric: Total_Visitors, Arcade_Visitors, Total_Purchases, Arcade_Revenue, Total_Labor_Hours, Customer_Complaints, Passholder_Percentage, International_Visitors, High_Temperature, Precipitation

In [18]:
```
#Question E
nan_values = lobster.isna().sum()
print(nan_values)
```

```
Date                       0
Day_of_Week                0
Total_Visitors             0
Arcade_Visitors            0
Total_Purchases            0
Arcade_Revenue             0
Total_Labor_Hours          0
Weather_Type               0
Special_Events            78
Customer_Complaints        0
Passholder_Percentage      0
International_Visitors     0
High_Temperature           3
General_Weather            0
Precipitation              0
dtype: int64
```

Question E - Columns with NaN values are Special_Events and High_Temperature

In [20]:
```
#Question E part c. View NaN cells
nan_rows = lobster[lobster['High_Temperature'].isna()]
print(nan_rows)
```

```
         Date Day_of_Week  Total_Visitors  Arcade_Visitors  Total_Purchases  
\
22  6/18/2024     Tuesday            1282              627          7830.61   
44  7/10/2024   Wednesday             234              408         12408.41   
67   8/2/2024      Friday            2435              162          8052.76   

    Arcade_Revenue  Total_Labor_Hours Weather_Type     Special_Events  \
22         1864.69             151.08        Rainy  Arcade Tournament   
44         2427.42             297.21       Stormy                NaN   
67         1707.11             258.66        Sunny                NaN   

    Customer_Complaints  Passholder_Percentage  International_Visitors  \
22                   13                   0.23                     441   
44                   11                   0.25                      78   
67                   14                   0.24                     973   

    High_Temperature General_Weather  Precipitation  
22               NaN    Rainy/Stormy              1  
44               NaN    Rainy/Stormy              1  
67               NaN            Mild              0  
```

In [22]:
```
#Question E part c. Replace NaN values with No Event as provided in the colu
lobster['Special_Events']= lobster['Special_Events'].fillna('No Event')
```

From the data set description, NaN event mean No Event

In [24]:
```
#Question E part b and c confirm Spacial_Events has no NaN
nan_values =lobster.isna().sum()
print(nan_values)
```

```
Date                      0
Day_of_Week               0
Total_Visitors            0
Arcade_Visitors           0
Total_Purchases           0
Arcade_Revenue            0
Total_Labor_Hours         0
Weather_Type              0
Special_Events            0
Customer_Complaints       0
Passholder_Percentage     0
International_Visitors    0
High_Temperature          3
General_Weather           0
Precipitation             0
dtype: int64
```

In [26]:
```
#Questio E part b and c . Replace NaN values in High_Temperatur column and c
rep_values = [85,86,86]
```

In [28]:
```
nan_indices = lobster[lobster['High_Temperature'].isna()].index[:len(rep_val
lobster.loc[nan_indices, 'High_Temperature'] = rep_values
```

A google search of high temperature weather in Portland, Maine on 6/18/2024 ,7/10/2024 and 8/2/2024 showed that high temperature for the 3 days were 85,86,86 consecutevily

In [30]:
```
#Question E cont...replace and confirm
nan_values =lobster.isna().sum()
print(nan_values)
```

```
Date                       0
Day_of_Week                0
Total_Visitors             0
Arcade_Visitors            0
Total_Purchases            0
Arcade_Revenue             0
Total_Labor_Hours          0
Weather_Type               0
Special_Events             0
Customer_Complaints        0
Passholder_Percentage      0
International_Visitors     0
High_Temperature           0
General_Weather            0
Precipitation              0
dtype: int64
```

In [32]:
```
#Question F- # Rearrange the columns to move 'Day_of_Week' next to 'Date'
columns = ['Date', 'Day_of_Week'] + [col for col in lobster.columns if col r
lobster = lobster[columns]
print(lobster.head())
```

```
        Date Day_of_Week  Total_Visitors  Arcade_Visitors  Total_Purchases  
\
0  5/27/2024      Monday            1060              267         22575.71
1  5/28/2024     Tuesday            1494              552         17142.52
2  5/29/2024   Wednesday            1330              447         10229.99
3  5/30/2024    Thursday            1295              442          6442.65
4  5/31/2024      Friday            1838              256         28409.06

   Arcade_Revenue  Total_Labor_Hours Weather_Type     Special_Events  \
0         3524.30             185.84        Sunny  Arcade Tournament
1         1652.93             250.90       Cloudy           No Event
2         3251.81             115.59       Cloudy           No Event
3          867.17             347.30       Stormy           No Event
4          747.60             234.62        Sunny  Arcade Tournament

   Customer_Complaints  Passholder_Percentage  International_Visitors  \
0                    9                   0.22                     357
1                    3                   0.23                     424
2                    0                   0.22                     441
3                    7                   0.29                     338
4                    0                   0.21                     615

   High_Temperature General_Weather  Precipitation
0              65.4            Cool              0
1              66.4            Cool              0
2              90.2             Hot              0
3              93.1     Rainy/Stormy              1
4              73.1            Mild              0
```

In [34]:
```
#Question F part b - # Rename the column 'Total_Purchases' to 'Total_Revenue
lobster = lobster.rename(columns={'Total_Purchases': 'Total_Revenue'})

print(lobster.head())
```

```
        Date Day_of_Week  Total_Visitors  Arcade_Visitors  Total_Revenue  \
0  5/27/2024      Monday            1060              267       22575.71
1  5/28/2024     Tuesday            1494              552       17142.52
2  5/29/2024   Wednesday            1330              447       10229.99
3  5/30/2024    Thursday            1295              442        6442.65
4  5/31/2024      Friday            1838              256       28409.06

   Arcade_Revenue  Total_Labor_Hours Weather_Type     Special_Events  \
0         3524.30             185.84        Sunny  Arcade Tournament
1         1652.93             250.90       Cloudy           No Event
2         3251.81             115.59       Cloudy           No Event
3          867.17             347.30       Stormy           No Event
4          747.60             234.62        Sunny  Arcade Tournament

   Customer_Complaints  Passholder_Percentage  International_Visitors  \
0                    9                   0.22                     357
1                    3                   0.23                     424
2                    0                   0.22                     441
3                    7                   0.29                     338
4                    0                   0.21                     615

   High_Temperature General_Weather  Precipitation
0              65.4            Cool              0
1              66.4            Cool              0
2              90.2             Hot              0
3              93.1   Rainy/Stormy              1
4              73.1            Mild              0
```

In [36]:
```
#Question G -create staff_efficiency variable
lobster['staff_efficiency'] = lobster['Total_Revenue'] / lobster['Total_Labo

print(lobster[['Total_Revenue', 'Total_Labor_Hours', 'staff_efficiency']].he
```

```
   Total_Revenue  Total_Labor_Hours  staff_efficiency
0       22575.71             185.84        121.479283
1       17142.52             250.90         68.324113
2       10229.99             115.59         88.502379
3        6442.65             347.30         18.550677
4       28409.06             234.62        121.085415
```

In [38]:
```
## Sort the dataset by 'staff_efficiency' in descending order
lobster_sorted = lobster.sort_values(by='staff_efficiency', ascending=False)

print(lobster_sorted[['Total_Revenue', 'Total_Labor_Hours', 'staff_efficienc
```

```
    Total_Revenue  Total_Labor_Hours  staff_efficiency
36       19604.96             139.99        140.045432
97       18142.97             133.16        136.249399
0        22575.71             185.84        121.479283
4        28409.06             234.62        121.085415
69       17466.68             158.64        110.102622
```

In [40]:
```
# Isolate the 10 most efficient days
top_10_days = lobster_sorted.head(10)

# Display the 10 most efficient days
print(top_10_days)

# Analyze differences between these days and the overall dataset
overall_mean_efficiency = lobster['staff_efficiency'].mean()
top_10_mean_efficiency = top_10_days['staff_efficiency'].mean()

print(f"Overall Mean Efficiency: {overall_mean_efficiency}")
print(f"Top 10 Mean Efficiency: {top_10_mean_efficiency}")
```

```
         Date Day_of_Week  Total_Visitors  Arcade_Visitors  Total_Revenue  \
36   7/2/2024     Tuesday            2261              239       19604.96   
97   9/1/2024      Sunday            1685              398       18142.97   
0   5/27/2024      Monday            1060              267       22575.71   
4   5/31/2024      Friday            1838              256       28409.06   
69   8/4/2024      Sunday             692              691       17466.68   
46  7/12/2024      Friday            2155              460       14499.29   
56  7/22/2024      Monday            1195              546       12301.13   
52  7/18/2024    Thursday            1779              657       14544.99   
18  6/14/2024      Friday             659              574       26427.39   
30  6/26/2024   Wednesday            1728              380       19034.52   

    Arcade_Revenue  Total_Labor_Hours Weather_Type     Special_Events  \
36         3715.68             139.99        Rainy          Fireworks   
97         2066.83             133.16        Sunny      Food Festival   
0          3524.30             185.84        Sunny  Arcade Tournament   
4           747.60             234.62        Sunny  Arcade Tournament   
69         5818.36             158.64       Stormy           No Event   
46         1116.58             136.68        Sunny           No Event   
56         2460.53             122.35       Stormy          Fireworks   
52         4705.76             145.39       Stormy           No Event   
18         5783.81             269.39        Rainy     Music Festival   
30         1374.71             195.29       Cloudy           No Event   

    Customer_Complaints  Passholder_Percentage  International_Visitors  \
36                    9                   0.16                     928   
97                    3                   0.21                     657   
0                     9                   0.22                     357   
4                     0                   0.21                     615   
69                    1                   0.23                     305   
46                    6                   0.18                     952   
56                   12                   0.26                     346   
52                   13                   0.29                     504   
18                    7                   0.26                     224   
30                   13                   0.22                     529   

    High_Temperature General_Weather  Precipitation  staff_efficiency  
36              66.4    Rainy/Stormy              1        140.045432  
97              73.6            Mild              0        136.249399  
0               65.4            Cool              0        121.479283  
4               73.1            Mild              0        121.085415  
69              69.5    Rainy/Stormy              1        110.102622  
46              81.8            Mild              0        106.082016  
56              91.3    Rainy/Stormy              1        100.540499  
52              86.7    Rainy/Stormy              1        100.041200  
18              88.0    Rainy/Stormy              1         98.100857  
30              70.3            Mild              0         97.467971  
Overall Mean Efficiency: 57.15846321994519
Top 10 Mean Efficiency: 113.11946939862636
```

The 10 most efficient days in the dataset appear to be strongly influenced by special events. These days include events such as Fireworks, Food Festival, Music Festival, and Arcade Tournament, which likely attracted more visitors and significantly boosted revenue. The average staff efficiency for these top 10 days is 113.12, which is nearly double the overall dataset average of 57.16, indicating that these days stand out in terms of performance. Special events seem to drive higher efficiency by increasing visitor numbers and revenue without a proportionate increase in labor hours. This suggests that organizing or promoting similar events could be a strategic way for Lobster Land to optimize efficiency and maximize revenue.

In [43]:
```
#Question H Create the 'international_percentage' variable
lobster['international_percentage'] = (lobster['International_Visitors'] / l

# Display the updated DataFrame with the new column
print(lobster[['Date', 'Total_Visitors', 'International_Visitors', 'internat
```

```
        Date  Total_Visitors  International_Visitors  international_percenta
ge
0  5/27/2024            1060                     357                 33.6792
45
1  5/28/2024            1494                     424                 28.3801
87
2  5/29/2024            1330                     441                 33.1578
95
3  5/30/2024            1295                     338                 26.1003
86
4  5/31/2024            1838                     615                 33.4602
83
```

In [45]:
```
# Calculate the correlation between international_percentage and Passholder_
correlation = lobster['international_percentage'].corr(lobster['Passholder_P

# Display the correlation
print(f"Correlation between international_percentage and Passholder_Percenta
```

```
Correlation between international_percentage and Passholder_Percentage: -0.6
721775465715214
```

The correlation between international_percentage and Passholder_Percentage is -0.67, indicating a moderate to strong negative relationship. This means that as the percentage of international visitors increases, the percentage of passholders tends to decrease, and vice versa. This likely happens because passholders are typically local or regional visitors who purchase season passes, while international visitors are more likely to visit once during their travels. Days with higher international visitors naturally have a smaller proportion of local passholders, and days dominated by locals see fewer international visitors. This suggests a clear distinction between these two groups, highlighting the need for tailored marketing strategies, such as one-day promotions for international visitors and season pass incentives for local customers.

In [48]:
```
#Question I
print(lobster.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 17 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Date                      99 non-null     object 
 1   Day_of_Week               99 non-null     object 
 2   Total_Visitors            99 non-null     int64  
 3   Arcade_Visitors           99 non-null     int64  
 4   Total_Revenue             99 non-null     float64
 5   Arcade_Revenue            99 non-null     float64
 6   Total_Labor_Hours         99 non-null     float64
 7   Weather_Type              99 non-null     object 
 8   Special_Events            99 non-null     object 
 9   Customer_Complaints       99 non-null     int64  
 10  Passholder_Percentage     99 non-null     float64
 11  International_Visitors    99 non-null     int64  
 12  High_Temperature          99 non-null     float64
 13  General_Weather           99 non-null     object 
 14  Precipitation             99 non-null     int64  
 15  staff_efficiency          99 non-null     float64
 16  international_percentage  99 non-null     float64
dtypes: float64(7), int64(5), object(5)
memory usage: 13.3+ KB
None
```

In [50]:
```
# Convert the 'Date' column to a datetime object
lobster['Date'] = pd.to_datetime(lobster['Date'], errors='coerce')
```

In [52]:
```
# Verify the conversion
print(lobster.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99 entries, 0 to 98
Data columns (total 17 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Date                      99 non-null     datetime64[ns]
 1   Day_of_Week               99 non-null     object
 2   Total_Visitors            99 non-null     int64
 3   Arcade_Visitors           99 non-null     int64
 4   Total_Revenue             99 non-null     float64
 5   Arcade_Revenue            99 non-null     float64
 6   Total_Labor_Hours         99 non-null     float64
 7   Weather_Type              99 non-null     object
 8   Special_Events            99 non-null     object
 9   Customer_Complaints       99 non-null     int64
 10  Passholder_Percentage     99 non-null     float64
 11  International_Visitors    99 non-null     int64
 12  High_Temperature          99 non-null     float64
 13  General_Weather           99 non-null     object
 14  Precipitation             99 non-null     int64
 15  staff_efficiency          99 non-null     float64
 16  international_percentage  99 non-null     float64
dtypes: datetime64[ns](1), float64(7), int64(5), object(4)
memory usage: 13.3+ KB
None
```

Converting the Date variable to a datetime64[ns] format allows Python to recognize it as an actual date rather than a string. This enables the dataset to be used more effectively in any analysis involving time. For instance, Python can now extract components like the year, month, or day directly from the date, which would be difficult with a string format. It also allows for easier sorting and filtering of the data based on time periods, such as retrieving rows within a specific date range or identifying trends over time. Additionally, this conversion ensures consistency in the way dates are handled, avoiding potential errors from mixed formats. Overall, having Python see the Date variable as a true date makes it much more useful for time-based analysis, especially for tasks like calculating time differences, identifying seasonality, or preparing data for time-series analysis.

In [55]:
```python
#Question J - Group by 'Day_of_Week' and use describe() on 'Total_Revenue'
day_revenue_stats = lobster.groupby('Day_of_Week')['Total_Revenue'].describe

# Display the statistics
print(day_revenue_stats)
```

```
             count          mean          std       min         25%  \
Day_of_Week
Friday        14.0  17133.101429  6309.542051   8052.76  12365.9700
Monday        15.0  13525.969333  5161.017456   7624.32   9002.9300
Saturday      14.0  21448.985000  4161.667899  15353.92  18287.3075
Sunday        14.0  18807.897143  4946.299831  11984.34  15227.6475
Thursday      14.0  12125.405000  4653.834452   6056.24   7931.7700
Tuesday       14.0  14155.262143  5128.831358   5271.13  10193.9950
Wednesday     14.0  12943.608571  3422.424946   9183.07   9936.1925

                   50%         75%       max
Day_of_Week
Friday       15628.680  21562.4450  28409.06
Monday       12301.130  18278.0500  22575.71
Saturday     21655.475  23664.5925  28599.96
Sunday       17804.825  21440.6825  28132.69
Thursday     12138.675  16272.5750  19296.08
Tuesday      16138.935  18130.2900  19732.53
Wednesday    12520.310  15416.1950  19034.52
```

The higher revenue on Saturdays and Sundays likely results from increased leisure time, making these days prime for family outings and entertainment. In contrast, weekdays, especially Thursday and Wednesday, show lower average revenues, as fewer people are free to visit. The variability in Friday's revenue may indicate that this day benefits from both weekend-level crowds and weekday promotions or events. These trends suggest that Lobster Land could benefit from targeted promotions during weekdays to drive more visitors and maintain consistent revenue throughout the week.

In [60]:
```
#Question K- Select only numeric columns for correlation
numeric_columns = lobster.select_dtypes(include=['number'])

# Calculate the correlation matrix
correlation_matrix = numeric_columns.corr()

# Display the correlation matrix
print(correlation_matrix)
```

```
                          Total_Visitors  Arcade_Visitors  Total_Revenue  \
Total_Visitors                  1.000000         0.091358      -0.101004   
Arcade_Visitors                 0.091358         1.000000      -0.083089   
Total_Revenue                  -0.101004        -0.083089       1.000000   
Arcade_Revenue                  0.119088         0.008905       0.206017   
Total_Labor_Hours              -0.191928        -0.082671       0.192167   
Customer_Complaints             0.139288         0.085257      -0.192131   
Passholder_Percentage          -0.220726         0.063360      -0.409345   
International_Visitors          0.955973         0.095152       0.007228   
High_Temperature               -0.055025         0.041316       0.019759   
Precipitation                  -0.102971         0.049386      -0.064037   
staff_efficiency                0.095876         0.010835       0.540774   
international_percentage        0.350451         0.053921       0.386487   

                          Arcade_Revenue  Total_Labor_Hours  \
Total_Visitors                  0.119088          -0.191928   
Arcade_Visitors                 0.008905          -0.082671   
Total_Revenue                   0.206017           0.192167   
Arcade_Revenue                  1.000000          -0.058063   
Total_Labor_Hours              -0.058063           1.000000   
Customer_Complaints            -0.162349          -0.153790   
Passholder_Percentage          -0.294743           0.028926   
International_Visitors          0.205847          -0.179584   
High_Temperature                0.063346          -0.085215   
Precipitation                   0.105889          -0.051462   
staff_efficiency                0.171905          -0.655341   
international_percentage        0.344139          -0.036125   

                          Customer_Complaints  Passholder_Percentage  \
Total_Visitors                       0.139288              -0.220726   
Arcade_Visitors                      0.085257               0.063360   
Total_Revenue                       -0.192131              -0.409345   
Arcade_Revenue                      -0.162349              -0.294743   
Total_Labor_Hours                   -0.153790               0.028926   
Customer_Complaints                  1.000000              -0.029790   
Passholder_Percentage               -0.029790               1.000000   
International_Visitors               0.143018              -0.380185   
High_Temperature                     0.002081               0.049463   
Precipitation                        0.065717              -0.017832   
staff_efficiency                    -0.052985              -0.316841   
international_percentage            -0.007313              -0.672178   

                          International_Visitors  High_Temperature  \
Total_Visitors                          0.955973         -0.055025   
Arcade_Visitors                         0.095152          0.041316   
Total_Revenue                           0.007228          0.019759   
Arcade_Revenue                          0.205847          0.063346   
Total_Labor_Hours                      -0.179584         -0.085215   
Customer_Complaints                     0.143018          0.002081   
Passholder_Percentage                  -0.380185          0.049463   
International_Visitors                  1.000000         -0.017057   
High_Temperature                       -0.017057          1.000000   
Precipitation                          -0.037891          0.066644   
staff_efficiency                        0.155964          0.037056   
international_percentage                0.587681          0.071913   
```

```
                          Precipitation  staff_efficiency  \
Total_Visitors                -0.102971          0.095876
Arcade_Visitors                0.049386          0.010835
Total_Revenue                 -0.064037          0.540774
Arcade_Revenue                 0.105889          0.171905
Total_Labor_Hours             -0.051462         -0.655341
Customer_Complaints            0.065717         -0.052985
Passholder_Percentage         -0.017832         -0.316841
International_Visitors        -0.037891          0.155964
High_Temperature               0.066644          0.037056
Precipitation                  1.000000         -0.006221
staff_efficiency              -0.006221          1.000000
international_percentage       0.148108          0.285361

                          international_percentage
Total_Visitors                            0.350451
Arcade_Visitors                           0.053921
Total_Revenue                             0.386487
Arcade_Revenue                            0.344139
Total_Labor_Hours                        -0.036125
Customer_Complaints                      -0.007313
Passholder_Percentage                    -0.672178
International_Visitors                    0.587681
High_Temperature                          0.071913
Precipitation                             0.148108
staff_efficiency                          0.285361
international_percentage                  1.000000
```

Based on the correlation matrix, Total_Visitors and International_Visitors are highly correlated with a coefficient of 0.96. This indicates that these two variables are not identical but are mostly redundant, as the number of international visitors is a subset of total visitors.The high correlation suggests that knowing the value of one variable essentially gives you an idea of the other. For example, on days with more total visitors, there are generally more international visitors, as international visitors make up a fixed or predictable proportion of total visitors.

In [63]:
```python
#Question K con...remove International_Visitors
lobster = lobster.drop(columns=['International_Visitors'])
print(lobster.head())
```

```
         Date Day_of_Week  Total_Visitors  Arcade_Visitors  Total_Revenue  \
0 2024-05-27      Monday            1060              267       22575.71   
1 2024-05-28     Tuesday            1494              552       17142.52   
2 2024-05-29   Wednesday            1330              447       10229.99   
3 2024-05-30    Thursday            1295              442        6442.65   
4 2024-05-31      Friday            1838              256       28409.06   

   Arcade_Revenue  Total_Labor_Hours Weather_Type     Special_Events  \
0         3524.30             185.84        Sunny  Arcade Tournament   
1         1652.93             250.90       Cloudy           No Event   
2         3251.81             115.59       Cloudy           No Event   
3          867.17             347.30       Stormy           No Event   
4          747.60             234.62        Sunny  Arcade Tournament   

   Customer_Complaints  Passholder_Percentage  High_Temperature  \
0                    9                   0.22              65.4   
1                    3                   0.23              66.4   
2                    0                   0.22              90.2   
3                    7                   0.29              93.1   
4                    0                   0.21              73.1   

  General_Weather  Precipitation  staff_efficiency  international_percentage  
0            Cool              0        121.479283                 33.679245  
1            Cool              0         68.324113                 28.380187  
2             Hot              0         88.502379                 33.157895  
3    Rainy/Stormy              1         18.550677                 26.100386  
4            Mild              0        121.085415                 33.460283  
```

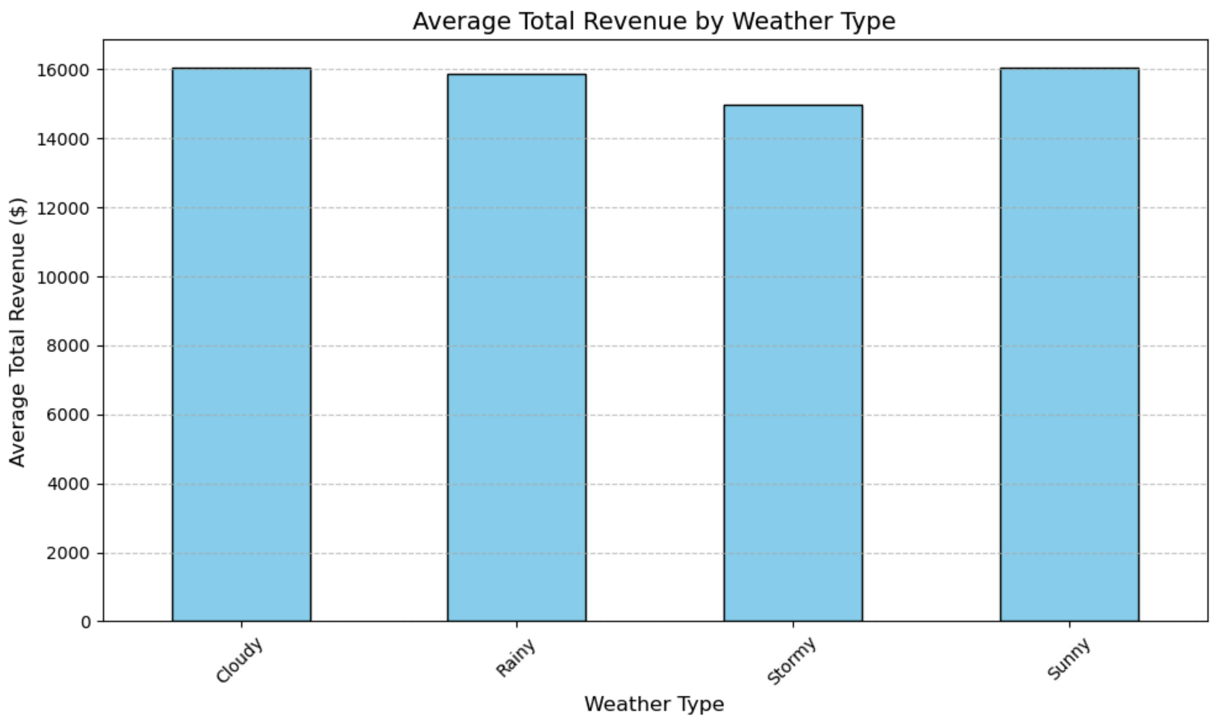
In [70]:
```
#####PART 2
import matplotlib.pyplot as plt
```

In [72]:
```
# Group by 'Weather_Type' and calculate the mean of 'Total_Revenue'
weather_revenue = lobster.groupby('Weather_Type')['Total_Revenue'].mean()

# Create a bar plot
plt.figure(figsize=(10, 6))
weather_revenue.plot(kind='bar', color='skyblue', edgecolor='black')

# Add title and labels
plt.title('Average Total Revenue by Weather Type', fontsize=14)
plt.xlabel('Weather Type', fontsize=12)
plt.ylabel('Average Total Revenue ($)', fontsize=12)
plt.xticks(rotation=45, fontsize=10)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show the plot
plt.tight_layout()
```



This bar plot shows the average total revenue for each weather type. The revenue appears relatively consistent across different weather conditions, with no dramatic peaks or dips. This suggests that visitor attendance and spending at Lobster Land are not heavily influenced by the weather. This result aligns with the expectation for venues that provide diverse attractions and experiences, regardless of outdoor conditions. It might indicate that indoor attractions, such as arcades, play a significant role in stabilizing revenue during less favorable weather.

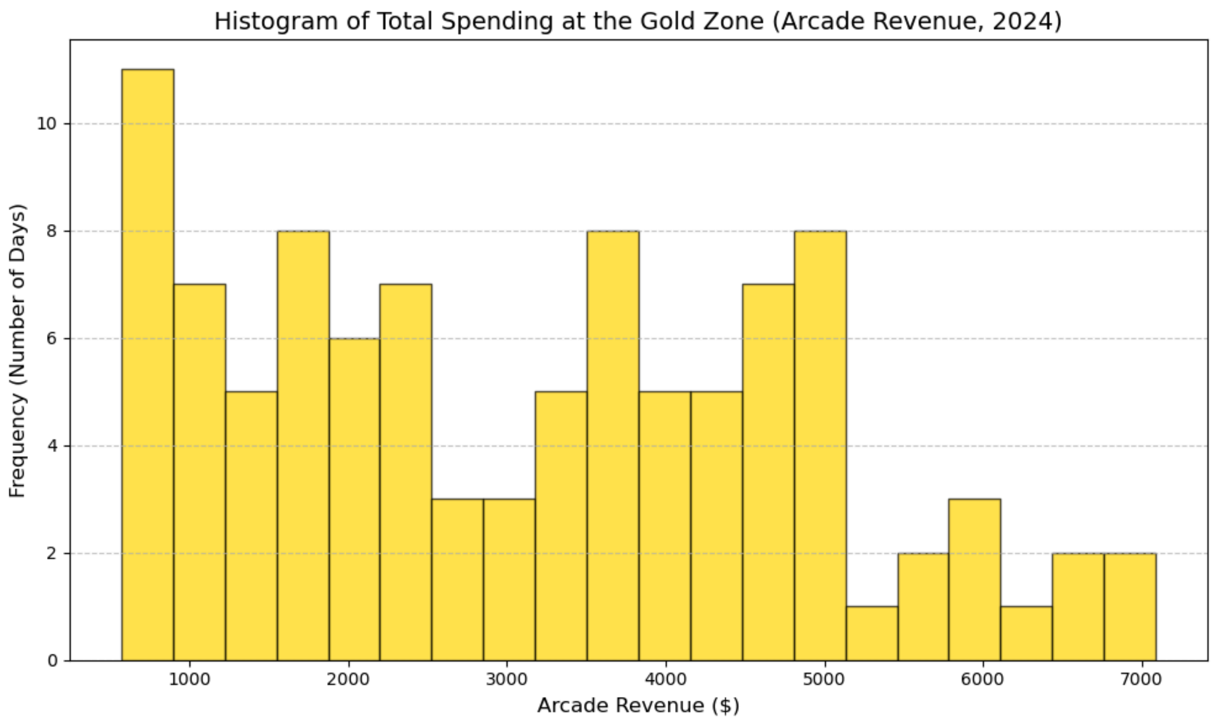
In [77]:
```
#Question M- Filter for the 2024 season
lobster_2024 = lobster[lobster['Date'].dt.year == 2024]

# Create a histogram for Gold Zone spending (Arcade Revenue) in 2024
plt.figure(figsize=(10, 6))
plt.hist(lobster_2024['Arcade_Revenue'], bins=20, color='gold', edgecolor='b

# Add title and labels
plt.title('Histogram of Total Spending at the Gold Zone (Arcade Revenue, 202
plt.xlabel('Arcade Revenue ($)', fontsize=12)
plt.ylabel('Frequency (Number of Days)', fontsize=12)

# Add gridlines for better readability
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show the plot
plt.tight_layout()
plt.show()
```



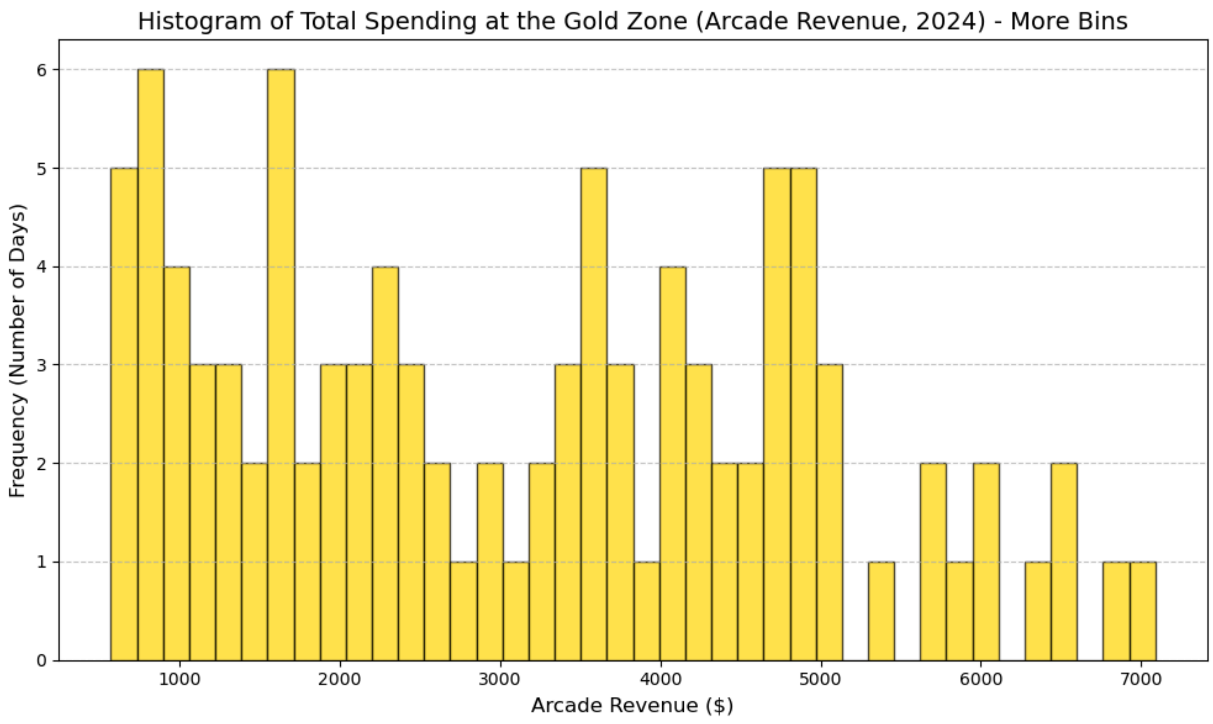
The histogram depicts the distribution of total spending in the Gold Zone (Arcade Revenue) during the 2024 season. The distribution shows a significant number of days with arcade revenue clustered around lower values, particularly between 500 and 2,000, suggesting that lower spending is more common. However, there are also a few days with much higher revenue, up to $7,000, which indicates some outliers or exceptionally high-performing days. This pattern may reflect typical daily spending trends with occasional spikes during special events or busy periods.

In [81]:
```
#Question M part b-  Create a histogram for Gold Zone spending (Arcade Rever
plt.figure(figsize=(10, 6))
plt.hist(lobster_2024['Arcade_Revenue'], bins=40, color='gold', edgecolor='b

# Add title and labels
plt.title('Histogram of Total Spending at the Gold Zone (Arcade Revenue, 202
plt.xlabel('Arcade Revenue ($)', fontsize=12)
plt.ylabel('Frequency (Number of Days)', fontsize=12)

# Add gridlines for better readability
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Show the plot
plt.tight_layout()
plt.show()
```



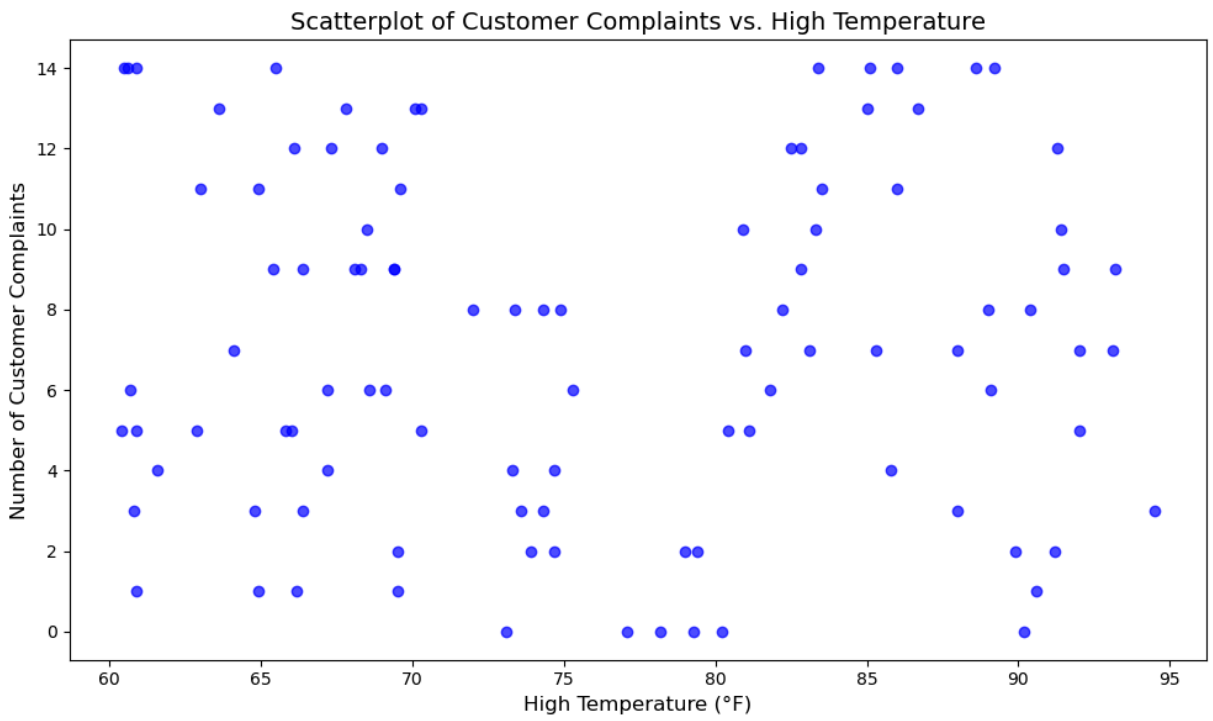
In this second histogram, which has more bins than the first, we can observe that the increased number of bins provides a finer-grained view of the distribution of arcade revenue. With the higher number of bins, we can see more variability in spending across different ranges, revealing more detailed patterns. For example, the revenue appears to be more evenly distributed across different revenue intervals, which may have been less obvious in the first histogram with fewer bins.

In [84]:
```
#Question N Ensure the 'High_Temperature' and 'Customer_Complaints' columns
lobster['High_Temperature'] = pd.to_numeric(lobster['High_Temperature'], err
lobster['Customer_Complaints'] = pd.to_numeric(lobster['Customer_Complaints'

# Create a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(lobster['High_Temperature'], lobster['Customer_Complaints'], col

# Add title and labels
plt.title('Scatterplot of Customer Complaints vs. High Temperature', fontsiz
plt.xlabel('High Temperature (°F)', fontsize=12)
plt.ylabel('Number of Customer Complaints', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```



Based on the scatterplot, it appears that there is no strong relationship between the high temperature and the number of customer complaints. Complaints seem to be distributed fairly evenly across the range of temperatures. While there are some days with more complaints at higher temperatures, this pattern is not consistent.The number of customer complaints might be influenced by other factors besides just the temperature, such as the presence of special events, customer expectations, or the quality of service on a given day.

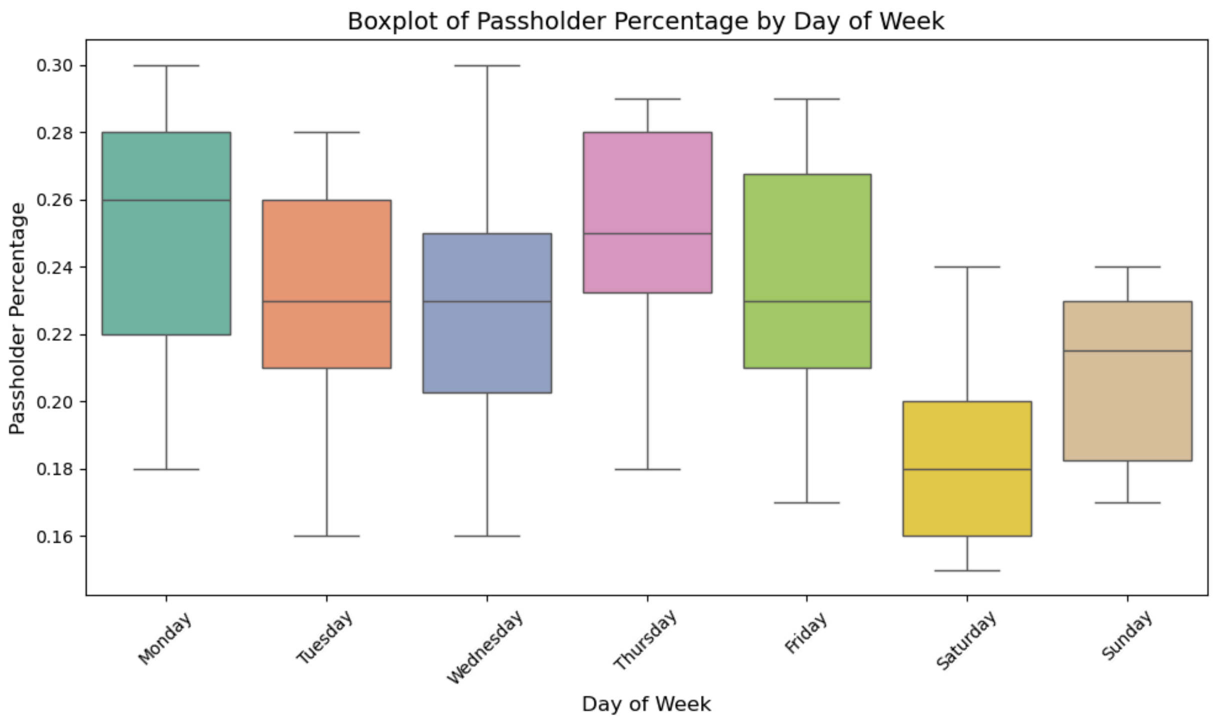
In [105…

```
#Question O
import seaborn as sns
# Create a boxplot for Passholder_Percentage by Day_of_Week
plt.figure(figsize=(10, 6))
sns.boxplot(x='Day_of_Week', y='Passholder_Percentage', data=lobster, palett

# Add title and labels
plt.title('Boxplot of Passholder Percentage by Day of Week', fontsize=14)
plt.xlabel('Day of Week', fontsize=12)
plt.ylabel('Passholder Percentage', fontsize=12)

# Rotate x-axis labels for readability
plt.xticks(rotation=45, fontsize=10)

# Show the plot
plt.tight_layout()
plt.show()
```



Each box represents the interquartile range (IQR), with the horizontal line indicating the median passholder percentage for that day.Monday and Friday show a higher median passholder percentage compared to the other days, with Monday having a wider IQR, indicating more variability in the data. Saturday and Sunday have lower passholder percentages with smaller IQRs, suggesting less variation in the data. Portland, Maine, is a popular tourist destination, especially during the summer months. Fridays and Mondays may see higher passholder percentages because of weekend visitors. People often plan trips around weekends, and those staying in the area may purchase passes to visit the park on these days. This is supported by the higher passholder percentages seen on those days. On Saturdays and Sundays, families may prefer day passes instead of season passes, especially if they’re only in town for the weekend. This could result in lower passholder percentages. Additionally, with more one-off visitors and less frequent local visits, weekend passholder percentages tend to be lower but more stable, reflecting a combination of local families and tourists who prefer short-term visits.

In [108…

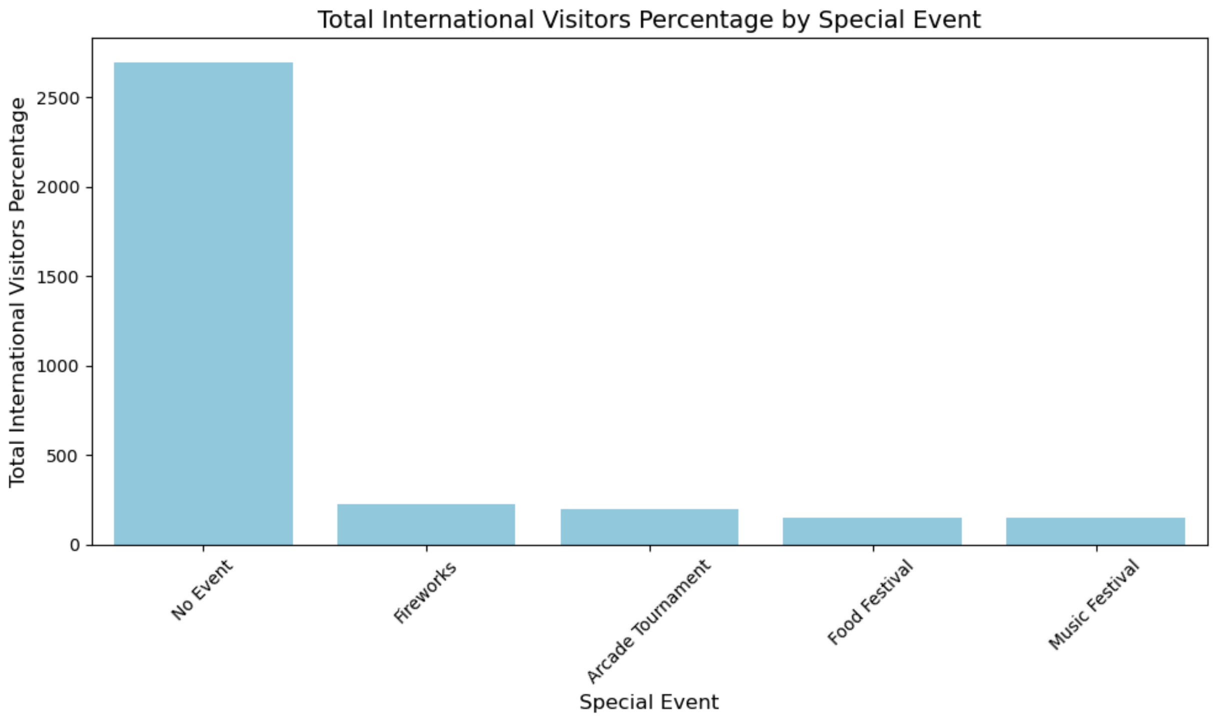
```
#Question P -  Group by 'Special_Events' and calculate the sum of 'internati
event_percentage = lobster.groupby('Special_Events')['international_percenta

# Create a barplot without error bars
plt.figure(figsize=(10, 6))
sns.barplot(x=event_percentage.index, y=event_percentage.values, color='skyb

# Add title and labels
plt.title('Total International Visitors Percentage by Special Event', fontsi
plt.xlabel('Special Event', fontsize=12)
plt.ylabel('Total International Visitors Percentage', fontsize=12)

# Rotate x-axis labels for readability
plt.xticks(rotation=45, fontsize=10)
```

```
# Show the plot
plt.tight_layout()
plt.show()
```



The bar plot shows that most of the total international visitors percentage is concentrated under the category "No Event", with much smaller percentages for the other events like Fireworks, Arcade Tournament, Food Festival, and Music Festival.While this plot shows a correlation between "No Event" and a high number of international visitors, we cannot conclude causality based solely on this data. There are likely other factors contributing to this trend, such as the overall tourist population in Portland, Maine, and the general appeal of the park regardless of events. The presence of special events could influence the number of local or regular visitors but might not be as significant in attracting international tourists. A more thorough investigation would be required to establish whether these events directly cause an increase in international visitors.

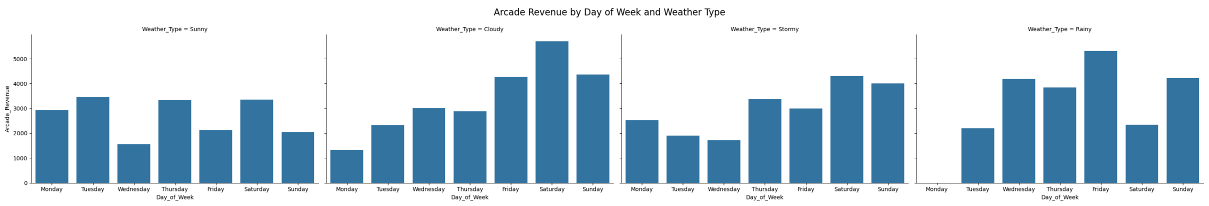
In [111…

```
#Question Q Create a faceted bar plot with facets for weather types and bars
sns.catplot(
    data=lobster,
    x='Day_of_Week',  # Days of the week on x-axis
    y='Arcade_Revenue',  # Arcade revenue on y-axis
    col='Weather_Type',  # Facets for different weather types
    kind='bar',  # Bar plot type
    errorbar=None,  # Do not show error bars (instead of ci=None)
    height=5,  # Height of each facet
    aspect=1.5  # Aspect ratio (width/height) for each facet
)

# Add title and labels
```

```
plt.subplots_adjust(top=0.85)  # Adjust the title to avoid overlap with the
plt.suptitle('Arcade Revenue by Day of Week and Weather Type', fontsize=16)

# Show the plot
plt.show()
```



Saturday (Cloudy) has notably high arcade revenue compared to other days, possibly due to the combination of weekend traffic and favorable weather.Monday (Rainy) has a low arcade revenue compared to other days, which could be a sign that fewer visitors come to the arcade at the beginning of the week, especially when weather conditions aren't ideal.While this plot helps visualize patterns, we cannot definitively conclude that weather directly influences arcade revenue. There could be other factors, such as holidays, local events, or external influences (like promotions), that are not accounted for in this plot. The observed patterns should be interpreted cautiously, and further analysis considering other factors (e.g., event schedules or local tourism trends) would be needed to draw stronger conclusions.

In [100…

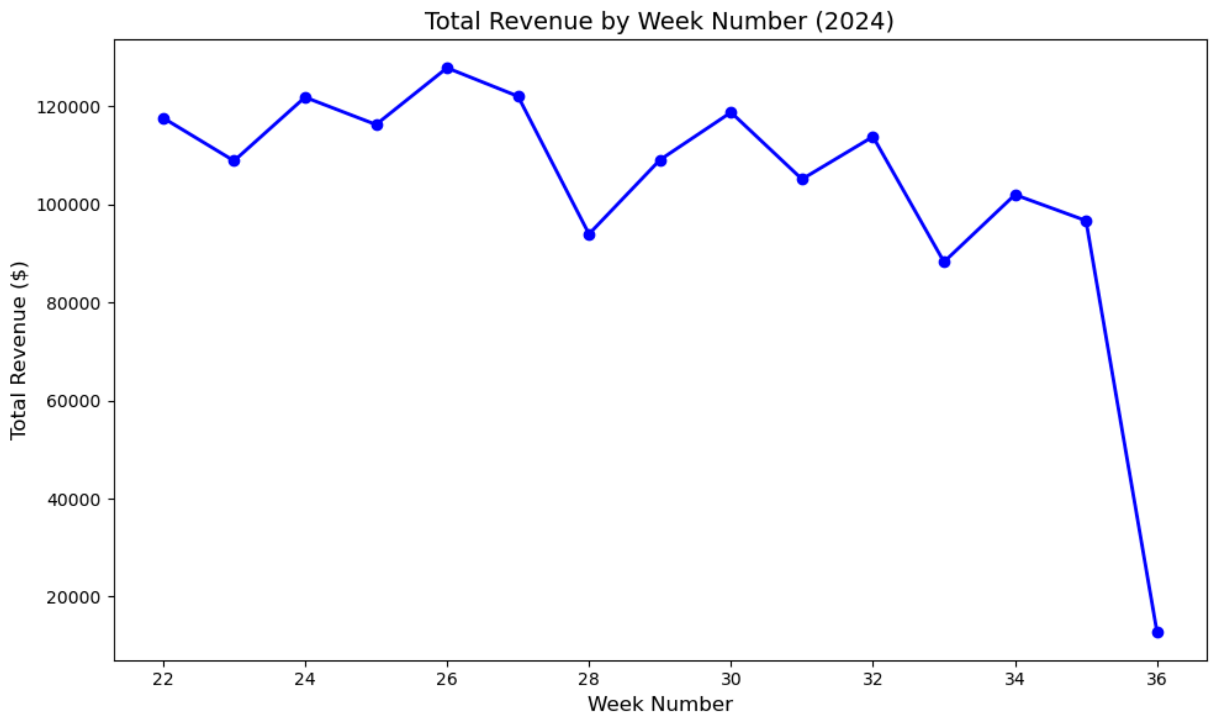
```
#Question R- Add a new column for the week number of the year
lobster['Week_Number'] = lobster['Date'].dt.isocalendar().week

# Group by 'Week_Number' and calculate the total revenue for each week
weekly_revenue = lobster.groupby('Week_Number')['Total_Revenue'].sum()

# Create a line plot
plt.figure(figsize=(10, 6))
plt.plot(weekly_revenue.index, weekly_revenue.values, marker='o', color='b',

# Add title and labels
plt.title('Total Revenue by Week Number (2024)', fontsize=14)
plt.xlabel('Week Number', fontsize=12)
plt.ylabel('Total Revenue ($)', fontsize=12)

# Show the plot
plt.tight_layout()
plt.show()
```



The line plot illustrates that total revenue throughout the 2024 season fluctuates, with noticeable peaks in certain weeks, particularly between weeks 22 and 30. The revenue reaches its highest point around week 26, suggesting a period of increased visitor activity, likely corresponding with the peak tourist season or favorable weather conditions. However, after week 30, there is a sharp decline in revenue, particularly after week 34, signaling a drop-off in activity towards the end of the season. This trend highlights the seasonality of the business, with revenue rising during peak times, such as summer and holidays, and falling off significantly during the off-season as visitor numbers decrease.

```
In [ ]:
```