

```

import pandas as pd

# Load the CSV files
dim_date = pd.read_csv('dim_date.csv')
dim_hotels = pd.read_csv('dim_hotels.csv')
dim_rooms = pd.read_csv('dim_rooms.csv')

# Display the first few rows of each dataset
print("dim_date:")
print(dim_date.head())
print("\ndim_hotels:")
print(dim_hotels.head())
print("\ndim_rooms:")
print(dim_rooms.head())
import matplotlib.pyplot as plt
import seaborn as sns

# Example filters (you can adjust these as needed)
city_filter = "New York"
status_filter = "Active"
platform_filter = "Booking.com"
month_filter = "2023-08"
week_filter = "2023-W33"

# Apply filters
filtered_rooms = dim_rooms[
    (dim_rooms['City'] == city_filter) &
    (dim_rooms['Status'] == status_filter) &
    (dim_rooms['Platform'] == platform_filter)
]

# Calculate key metrics
revenue = filtered_rooms['Revenue'].sum()
occupancy_rate = filtered_rooms['Occupied'].mean()
average_rating = filtered_rooms['Rating'].mean()

# Print key metrics
print(f"Revenue: ${revenue}")
print(f"Occupancy Rate: {occupancy_rate * 100:.2f}%")
print(f"Average Rating: {average_rating}")

# Aggregations by City and Property
city_metrics = filtered_rooms.groupby('City').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

property_metrics = filtered_rooms.groupby('Property').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

# Trends by Week
weekly_trends = filtered_rooms.groupby('Week').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',

```

```

        'Rating': 'mean'
    }).reset_index()

# Occupancy by Day Type
occupancy_day_type =
filtered_rooms.groupby('DayType')['Occupied'].mean().reset_index()

# Booking % by Platform
booking_platform =
filtered_rooms.groupby('Platform')['Occupied'].mean().reset_index()

# Visualizations
plt.figure(figsize=(12, 6))
sns.lineplot(data=weekly_trends, x='Week', y='Revenue', label='Revenue')
sns.lineplot(data=weekly_trends, x='Week', y='Occupied', label='Occupancy Rate')
sns.lineplot(data=weekly_trends, x='Week', y='Rating', label='Average Rating')
plt.title('Trends by Week')
plt.xlabel('Week')
plt.ylabel('Metrics')
plt.legend()
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=occupancy_day_type, x='DayType', y='Occupied')
plt.title('Occupancy by Day Type')
plt.xlabel('Day Type')
plt.ylabel('Occupancy Rate')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=booking_platform, x='Platform', y='Occupied')
plt.title('Booking % by Platform')
plt.xlabel('Platform')
plt.ylabel('Booking %')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=city_metrics, x='City', y='Revenue')
plt.title('Revenue by City')
plt.xlabel('City')
plt.ylabel('Revenue')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=property_metrics, x='Property', y='Revenue')
plt.title('Revenue by Property')
plt.xlabel('Property')
plt.ylabel('Revenue')
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Create synthetic booking data
np.random.seed(42)
num_entries = 1000
cities = ['New York', 'Los Angeles', 'Chicago', 'Houston', 'Phoenix']

```

```

statuses = ['Active', 'Inactive']
platforms = ['Booking.com', 'Airbnb', 'Expedia']
properties = ['Property_A', 'Property_B', 'Property_C', 'Property_D']

data = {
    'City': np.random.choice(cities, num_entries),
    'Status': np.random.choice(statuses, num_entries),
    'Platform': np.random.choice(platforms, num_entries),
    'Revenue': np.random.rand(num_entries) * 1000,
    'Occupied': np.random.rand(num_entries),
    'Rating': np.random.rand(num_entries) * 5,
    'Week': np.random.choice(pd.date_range('2023-01-01', '2023-12-31',
freq='W').strftime('%Y-W%U'), num_entries),
    'DayType': np.random.choice(['weekday', 'weekend'], num_entries),
    'Property': np.random.choice(properties, num_entries)
}

booking_data = pd.DataFrame(data)

# Example filters (you can adjust these as needed)
city_filter = "New York"
status_filter = "Active"
platform_filter = "Booking.com"

# Apply filters
filtered_data = booking_data[
    (booking_data['City'] == city_filter) &
    (booking_data['Status'] == status_filter) &
    (booking_data['Platform'] == platform_filter)
]

# Calculate key metrics
revenue = filtered_data['Revenue'].sum()
occupancy_rate = filtered_data['Occupied'].mean()
average_rating = filtered_data['Rating'].mean()

# Print key metrics
print(f"Revenue: ${revenue:.2f}")
print(f"Occupancy Rate: {occupancy_rate * 100:.2f}%")
print(f"Average Rating: {average_rating:.2f}")

# Aggregations by City and Property
city_metrics = filtered_data.groupby('City').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

property_metrics = filtered_data.groupby('Property').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

# Trends by Week
weekly_trends = filtered_data.groupby('Week').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',

```

```

        'Rating': 'mean'
    }).reset_index()

# Occupancy by Day Type
occupancy_day_type =
filtered_data.groupby('DayType')['Occupied'].mean().reset_index()

# Booking % by Platform
booking_platform =
filtered_data.groupby('Platform')['Occupied'].mean().reset_index()

# Visualizations
plt.figure(figsize=(12, 6))
sns.lineplot(data=weekly_trends, x='Week', y='Revenue', label='Revenue')
sns.lineplot(data=weekly_trends, x='Week', y='Occupied', label='Occupancy Rate')
sns.lineplot(data=weekly_trends, x='Week', y='Rating', label='Average Rating')
plt.title('Trends by Week')
plt.xlabel('Week')
plt.ylabel('Metrics')
plt.legend()
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=occupancy_day_type, x='DayType', y='Occupied')
plt.title('Occupancy by Day Type')
plt.xlabel('Day Type')
plt.ylabel('Occupancy Rate')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=booking_platform, x='Platform', y='Occupied')
plt.title('Booking % by Platform')
plt.xlabel('Platform')
plt.ylabel('Booking %')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=city_metrics, x='City', y='Revenue')
plt.title('Revenue by City')
plt.xlabel('City')
plt.ylabel('Revenue')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=property_metrics, x='Property', y='Revenue')
plt.title('Revenue by Property')
plt.xlabel('Property')
plt.ylabel('Revenue')
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Create synthetic booking data
np.random.seed(42)
num_entries = 1000

```

```

cities = ['New York', 'Los Angeles', 'Chicago', 'Houston', 'Phoenix']
statuses = ['Active', 'Inactive']
platforms = ['Booking.com', 'Airbnb', 'Expedia']
properties = ['Property_A', 'Property_B', 'Property_C', 'Property_D']

data = {
    'City': np.random.choice(cities, num_entries),
    'Status': np.random.choice(statuses, num_entries),
    'Platform': np.random.choice(platforms, num_entries),
    'Revenue': np.random.rand(num_entries) * 1000,
    'Occupied': np.random.rand(num_entries),
    'Rating': np.random.rand(num_entries) * 5,
    'Week': np.random.choice(pd.date_range('2023-01-01', '2023-12-31',
freq='W').strftime('%Y-W%U'), num_entries),
    'DayType': np.random.choice(['weekday', 'weekend'], num_entries),
    'Property': np.random.choice(properties, num_entries)
}

booking_data = pd.DataFrame(data)

# Example filters (you can adjust these as needed)
city_filter = "New York"
status_filter = "Active"
platform_filter = "Booking.com"

# Apply filters
filtered_data = booking_data[
    (booking_data['City'] == city_filter) &
    (booking_data['Status'] == status_filter) &
    (booking_data['Platform'] == platform_filter)
]

# Calculate key metrics
revenue = filtered_data['Revenue'].sum()
occupancy_rate = filtered_data['Occupied'].mean()
average_rating = filtered_data['Rating'].mean()

# Print key metrics
print(f"Revenue: ${revenue:.2f}")
print(f"Occupancy Rate: {occupancy_rate * 100:.2f}%")
print(f"Average Rating: {average_rating:.2f}")

# Aggregations by City and Property
city_metrics = filtered_data.groupby('City').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

property_metrics = filtered_data.groupby('Property').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

# Trends by Week
weekly_trends = filtered_data.groupby('Week').agg({
    'Revenue': 'sum',

```

```

        'Occupied': 'mean',
        'Rating': 'mean'
    }).reset_index()

# Occupancy by Day Type
occupancy_day_type =
filtered_data.groupby('DayType')['Occupied'].mean().reset_index()

# Booking % by Platform
booking_platform =
filtered_data.groupby('Platform')['Occupied'].mean().reset_index()

# Visualizations
plt.figure(figsize=(12, 6))
sns.lineplot(data=weekly_trends, x='Week', y='Revenue', label='Revenue')
sns.lineplot(data=weekly_trends, x='Week', y='Occupied', label='Occupancy Rate')
sns.lineplot(data=weekly_trends, x='Week', y='Rating', label='Average Rating')
plt.title('Trends by Week')
plt.xlabel('Week')
plt.ylabel('Metrics')
plt.legend()
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=occupancy_day_type, x='DayType', y='Occupied')
plt.title('Occupancy by Day Type')
plt.xlabel('Day Type')
plt.ylabel('Occupancy Rate')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=booking_platform, x='Platform', y='Occupied')
plt.title('Booking % by Platform')
plt.xlabel('Platform')
plt.ylabel('Booking %')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=city_metrics, x='City', y='Revenue')
plt.title('Revenue by City')
plt.xlabel('City')
plt.ylabel('Revenue')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=property_metrics, x='Property', y='Revenue')
plt.title('Revenue by Property')
plt.xlabel('Property')
plt.ylabel('Revenue')
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Create synthetic booking data
np.random.seed(42)

```

```

num_entries = 1000
cities = ['New York', 'Los Angeles', 'Chicago', 'Houston', 'Phoenix']
statuses = ['Active', 'Inactive']
platforms = ['Booking.com', 'Airbnb', 'Expedia']
properties = ['Property_A', 'Property_B', 'Property_C', 'Property_D']

data = {
    'City': np.random.choice(cities, num_entries),
    'Status': np.random.choice(statuses, num_entries),
    'Platform': np.random.choice(platforms, num_entries),
    'Revenue': np.random.rand(num_entries) * 1000,
    'Occupied': np.random.rand(num_entries),
    'Rating': np.random.rand(num_entries) * 5,
    'Week': np.random.choice(pd.date_range('2023-01-01', '2023-12-31',
freq='W').strftime('%Y-W%U'), num_entries),
    'DayType': np.random.choice(['weekday', 'weekend'], num_entries),
    'Property': np.random.choice(properties, num_entries)
}

booking_data = pd.DataFrame(data)

# Example filters (you can adjust these as needed)
city_filter = "New York"
status_filter = "Active"
platform_filter = "Booking.com"

# Apply filters
filtered_data = booking_data[
    (booking_data['City'] == city_filter) &
    (booking_data['Status'] == status_filter) &
    (booking_data['Platform'] == platform_filter)
]

# Calculate key metrics
revenue = filtered_data['Revenue'].sum()
occupancy_rate = filtered_data['Occupied'].mean()
average_rating = filtered_data['Rating'].mean()

# Print key metrics
print(f"Revenue: ${revenue:.2f}")
print(f"Occupancy Rate: {occupancy_rate * 100:.2f}%")
print(f"Average Rating: {average_rating:.2f}")

# Aggregations by City and Property
city_metrics = filtered_data.groupby('City').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

property_metrics = filtered_data.groupby('Property').agg({
    'Revenue': 'sum',
    'Occupied': 'mean',
    'Rating': 'mean'
}).reset_index()

# Trends by Week
weekly_trends = filtered_data.groupby('Week').agg({

```

```

        'Revenue': 'sum',
        'Occupied': 'mean',
        'Rating': 'mean'
    }).reset_index()

# Occupancy by Day Type
occupancy_day_type =
filtered_data.groupby('DayType')['Occupied'].mean().reset_index()

# Booking % by Platform
booking_platform =
filtered_data.groupby('Platform')['Occupied'].mean().reset_index()

# Visualizations
plt.figure(figsize=(12, 6))
sns.lineplot(data=weekly_trends, x='Week', y='Revenue', label='Revenue')
sns.lineplot(data=weekly_trends, x='Week', y='Occupied', label='Occupancy Rate')
sns.lineplot(data=weekly_trends, x='Week', y='Rating', label='Average Rating')
plt.title('Trends by Week')
plt.xlabel('Week')
plt.ylabel('Metrics')
plt.legend()
plt.xticks(rotation=45)
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=occupancy_day_type, x='DayType', y='Occupied')
plt.title('Occupancy by Day Type')
plt.xlabel('Day Type')
plt.ylabel('Occupancy Rate')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=booking_platform, x='Platform', y='Occupied')
plt.title('Booking % by Platform')
plt.xlabel('Platform')
plt.ylabel('Booking %')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=city_metrics, x='City', y='Revenue')
plt.title('Revenue by City')
plt.xlabel('City')
plt.ylabel('Revenue')
plt.show()

plt.figure(figsize=(12, 6))
sns.barplot(data=property_metrics, x='Property', y='Revenue')
plt.title('Revenue by Property')
plt.xlabel('Property')
plt.ylabel('Revenue')
plt.show()

```