Practical 4

Name-Devansh Tyagi

Class-E 3

Roll No-551

PRN-202201040173

Store the information in a DataFrame and identifiy the top 10 grains,

```
import pandas as pd

# Create a dictionary with the given data
data = {
    'GrainName': ['Ragi', 'Bajra', 'Ragi', 'Bajra', 'Ragi', 'Bajra', 'Oats', 'Sattu', 'Sooji', 'Brown rice',
                  'Wheat', 'Corn', 'Ragi', 'Bajra', 'Oats', 'Sattu', 'Sooji', 'Brown rice', 'Wheat', 'Corn',
                  'Sooji', 'Brown rice', 'Wheat', 'Corn', 'Ragi', 'Brown rice', 'Wheat', 'Corn'],
    'State': ['Maharashtra', 'Panjab', 'Maharashtra', 'Panjab', 'Maharashtra', 'Panjab', 'Hariyana', 'Gujarat',
              'Tamil Nadu', 'Telangana', 'West Bengol', 'UP', 'Maharashtra', 'Panjab', 'Hariyana', 'Gujarat',
              'Tamil Nadu', 'Telangana', 'West Bengol', 'UP', 'Tamil Nadu', 'Telangana', 'West Bengol', 'UP',
              'Maharashtra', 'Telangana', 'West Bengol', 'UP'],
    'City': ['Nagpur', 'Amritsar', 'Nagpur', 'Amritsar', 'Nagpur', 'Amritsar', 'Gurugram', 'Surat', 'Madurai',
             'Hyderabad', 'Asansole', 'Kanpur', 'Nagpur', 'Amritsar', 'Gurugram', 'Surat', 'Madurai', 'Hyderabad',
             'Asansole', 'Kanpur', 'Madurai', 'Hyderabad', 'Asansole', 'Kanpur', 'Nagpur', 'Hyderabad', 'Asansole',
             'Kanpur'],
    'Months': ['JAN', 'FEB', 'JAN', 'FEB', 'JAN', 'FEB', 'MARCH', 'APRIL', 'MAY', 'JUNE', 'JULY', 'AUG', 'JAN', 'FEB',
               'MARCH', 'APRIL', 'MAY', 'JUNE', 'JULY', 'AUG', 'MAY', 'JUNE', 'JULY', 'AUG', 'JAN', 'JUNE', 'JULY', 'AUG'],
    'Year': [2023] * 28,
    'Sales': [1000000, 1500000, 1000000, 1500000, 1000000, 1500000, 2000000, 2500000, 3000000, 3500000, 4000000,
              4500000, 1000000, 1500000, 2000000, 2500000, 3000000, 3500000, 4000000, 4500000, 3000000, 3500000,
              4000000, 4500000, 1000000, 3500000, 4000000, 4500000]
}
# Create a DataFrame from the dictionary
df = pd.DataFrame(data)

# Identify the 10 grains
top_10_grains = df['GrainName'].unique()[:10]

print("Top 10 grains:")
print(top_10_grains)
```

```
Top 10 grains:
['Ragi' 'Bajra' 'Oats' 'Sattu' 'Sooji' 'Brown rice' 'Wheat' 'Corn']
```

1. **Which was the best month for sales? How much was earned that month?**

```
[2] # Group the data by month and calculate the total sales for each month
    monthly_sales = df.groupby('Months')['Sales'].sum()

    # Find the month with the highest sales
    best_month = monthly_sales.idxmax()
    highest_earnings = monthly_sales.max()

    print("Best month for sales:", best_month)
    print("Earnings in the best month:", highest_earnings)
```

```
Best month for sales: AUG
Earnings in the best month: 18000000
```

2. **Which product sold the most? Why do you think it did?**

```
[3] # Group the data by product and calculate the total sales for each product
    product_sales = df.groupby('GrainName')['Sales'].sum()

    # Find the product with the highest sales
    best_product = product_sales.idxmax()

    print("Product with the highest sales:", best_product)

    Product with the highest sales: Corn
```

3. Which city sold the most products?

```
# Group the data by city and calculate the total sales for each city
city_sales = df.groupby('City')['Sales'].sum()

# Find the city with the highest sales
best_city = city_sales.idxmax()

print("City with the highest sales:", best_city)

City with the highest sales: Kanpur
```

4. What Products are most often sold together?

```
from itertools import combinations

# Get all combinations of two products
product_combinations = list(combinations(df['GrainName'].unique(), 2))

# Count the occurrences of each product combination
combination_counts = {}
for combination in product_combinations:
    count = len(df[(df['GrainName'] == combination[0]) & (df['GrainName'].shift(-1) == combination[1])])
    combination_counts[combination] = count

# Find the combinations with the highest counts
most_common_combinations = sorted(combination_counts.items(), key=lambda x: x[1], reverse=True)[:5]

print("Products most often sold together:")
for combination, count in most_common_combinations:
    print(combination[0], "and", combination[1], "(", count, "times )")

Products most often sold together:
Ragi and Bajra ( 4 times )
Brown rice and Wheat ( 4 times )
Wheat and Corn ( 4 times )
Sooji and Brown rice ( 3 times )
Bajra and Oats ( 2 times )
```