Name: Shivshankar J Ghyar

Roll No: 836H2

PRN: 202201040031

PRACTICAL NO 4

Implement all 20 grains using Pandas methods. The Sample Grains for the Sales Dataset are as:

- Which was the best month for sales? How much was earned that month?
- Which product sold the most? Why do you think it did?
- Which city sold the most products?
- What Products are most often sold together?
- How many unique grain types are present in the dataset?
- What is the average sales for each grain type?
- Calculate the average sales per month for each grain type, considering only the data from the last 2 years
- Calculate Aggregate Sum, Mean & Count of each city.
- Print all city's names in upper case
- Print all basic information about datset

#CODE:

```
import pandas as pd

df=pd.read_csv('grainsales (1).csv')

print(df)
```

	GrainName	State	City	Months	Year	Sales
0	Ragi	Maharashtra	Nagpur	JAN	2023	1000000
1	Bajra	Panjab	Amritsar	FEB	2023	1500000
2	Ragi	Maharashtra	Nagpur	JAN	2020	1000000
3	Bajra	Panjab	Amritsar	FEB	2023	1500000
4	Ragi	Maharashtra	Nagpur	JAN	2022	1000000
5	Bajra	Panjab	Amritsar	FEB	2022	1500000
6	Oats	Hariyana	Gurugram	MARCH	2023	2000000
7	Sattu	Gujarat	Surat	APRIL	2023	2500000
8	Sooji	Tamil Nadu	Madurai	MAY	2023	3000000
9	Brown rice	Telangana	Hyderabad	JUNE	2023	3500000
10	Wheat	West Bengol	Asansole	JULY	2022	4000000
11	Corn	UP	Kanpur	AUG	2023	4500000
12	Ragi	Maharashtra	Nagpur	JAN	2023	1000000
13	Bajra	Panjab	Amritsar	FEB	2022	1500000
14	Oats	Hariyana	Gurugram	MARCH	2023	2000000
15	Sattu	Gujarat	Surat	APRIL	2023	2500000
16	Sooji	Tamil Nadu	Madurai	MAY	2022	3000000
17	Brown rice	Telangana	Hyderabad	JUNE	2023	3500000
18	Wheat	West Bengol	Asansole	JULY	2023	4000000
19	Corn	UP	Kanpur	AUG	2023	4500000
20	Sooji	Tamil Nadu	Madurai	MAY	2022	3000000
21	Brown rice	Telangana	Hyderabad	JUNE	2023	3500000
22	Wheat	West Bengol	Asansole	JULY	2023	4000000

23 Corn UP Kanpur AUG 2023 4500000

24 Ragi Maharashtra Nagpur JAN 2022 1000000

25 Brown rice Telangana Hyderabad JUNE 2023 3500000

26 Wheat West Bengol Asansole JULY 2019 4000000

#Best MOnth for the Sale

```
import pandas as pd
df=pd.read_csv('grainsales (1).csv')
mm=df.groupby('Months')['Sales'].sum().idxmax()
tm=df.groupby('Months')['Sales'].sum().max()
print("The best month for the sale is:",mm)
print("Total earning of ",mm,"is:",tm)
```

```
The best month for the sale is: JULY
Total earning of  JULY is: 16000000
```

#Product which sold most

```
psm=df.GrainName.value_counts()
print("The product which sold most is:",psm)
print("Because total sales of it is: ",psm['Ragi'])
```

```
The product which sold most is: Ragi          5
Bajra         4
```

```
Brown rice     4
Wheat          4
Sooji          3
Corn           3
Oats           2
Sattu          2
Name: GrainName, dtype: int64
Because total sales of it is:  5
```

```
# City which sold the most products
cmp = df['City'].value_counts().idxmax()
cmn = df['City'].value_counts().max()
print("The city which sold the most product is:",cmp)
print("Number:",cmn)
```

```
The city which sold the most product is: Nagpur
Number: 5
```

Q4 What products are most often sold together?

```
pc = df.groupby('Year')['GrainName'].unique().reset_index()
print("Products most often sold together:")
print(pc)
```

```
Products most often sold together:
```

	Year	GrainName
0	2019	[Wheat]
1	2020	[Ragi]
2	2022	[Ragi, Bajra, Wheat, Sooji]
3	2023	[Ragi, Bajra, Oats, Sattu , Sooji, Brown rice ...

#Q5 Unique grain types in Grainsales

```
unique_grain = df['GrainName'].nunique()
print("Number of unique grain types in Grainsales: ", unique_grain)
```

```
Number of unique grain types in Grainsales:  8
```

#Q6 Average sale of each Grain

```
average_sales = df.groupby('GrainName')['Sales'].mean()
print(average_sales)
```

```
GrainName
Bajra         1500000.0
Brown rice    3500000.0
Corn          4500000.0
Oats          2000000.0
Ragi          1000000.0
Sattu         2500000.0
Sooji         3000000.0
Wheat         4000000.0
```

Name: Sales, dtype: float64

#Q7 average sales per month for each grain type

```
last_two_years = df[df['Year'] >= df['Year'].max() - 1]
average_sales_per_month =
last_two_years.groupby(['GrainName','Months'])['Sales'].mean()
print(average_sales_per_month)
```

GrainName Months

Bajra FEB 1500000.0

Brown rice JUNE 3500000.0

Corn AUG 4500000.0

Oats MARCH 2000000.0

Ragi JAN 1000000.0

Sattu APRIL 2500000.0

Sooji MAY 3000000.0

Wheat JULY 4000000.0

Name: Sales, dtype: float64

#Q8 Aggregate sum ,Mean & Count of each city

```
grouped_df = df.groupby('City').agg({'Sales': ['sum', 'mean', 'count']})
print(grouped_df)
```

	Sales		
	sum	mean	count
City			

Amritsar 6000000 1500000.0 4

Asansole 16000000 4000000.0 4

Gurugram 4000000 2000000.0 2

Hyderabad 14000000 3500000.0 4

Kanpur 13500000 4500000.0 3

Madurai 9000000 3000000.0 3

Nagpur 5000000 1000000.0 5

Surat 5000000 2500000.0 2

#Q9 All city's name in upper case

print(df['City'].str.upper())

```
0      NAGPUR
1    AMRITSAR
2      NAGPUR
3    AMRITSAR
4      NAGPUR
5    AMRITSAR
6    GURUGRAM
7       SURAT
8     MADURAI
9   HYDERABAD
10   ASANSOLE
11     KANPUR
12     NAGPUR
13   AMRITSAR
14   GURUGRAM
15      SURAT
16    MADURAI
17  HYDERABAD
18   ASANSOLE
19     KANPUR
20    MADURAI
21  HYDERABAD
22   ASANSOLE
23     KANPUR
24     NAGPUR
25  HYDERABAD
26   ASANSOLE
Name: City, dtype: object
```

#Q10 Basic information about the dataset:

print(df.info())

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 27 entries, 0 to 26

Data columns (total 6 columns):

#	Column	Non-Null Count	Dtype
0	GrainName	27 non-null	object
1	State	27 non-null	object
2	City	27 non-null	object
3	Months	27 non-null	object
4	Year	27 non-null	int64
5	Sales	27 non-null	int64

dtypes: int64(2), object(4)

memory usage: 1.4+ KB

None