

Practical 4 Submission

Name: Shreyash Padase

Roll No: 839

PRN No: 202201070071

Problem Statement: Read any real-life **dataset**. Store the data in Data Frames. Identify 10 grains for the given **dataset**.

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

1. Which was the best month for sales? How much was earned that month?
2. Which product sold the most? Why do you think it did?
3. Which city sold the most products?
4. What Products are most often sold together?

Code:

```
import pandas as pd
```

```
df=pd.read_csv("grainsales.csv",delimiter=",")
```

```
#Maximum sales in Month
```

```
month=df.groupby('Months')['Sales'].sum().idxmax()
```

```
print("\nMaximum sales are done in the month of: ",month)
```

```
sales=df.groupby('Months')['Sales'].sum().max() print("\nThe  
earning made in that month is: ",sales)
```

#Product which sold the most and by whom

```
max_sales=df.groupby('GrainName')['Sales'].sum().idxmax()
```

```
print("\nThe product who sold the most is: ",max_sales)
```

#City which sold the most products

```
city=df.groupby('City')['Sales'].sum().idxmax() print("\nThe
```

```
city which sold the most products: ",city)
```

#Minimum sales done in the month

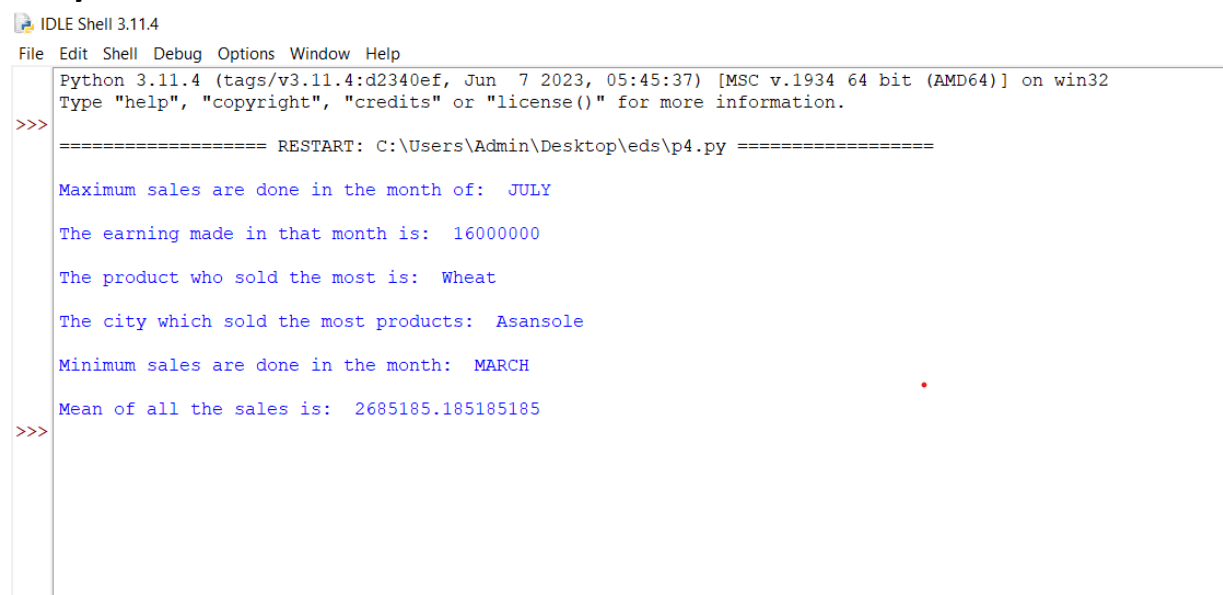
```
min_sales=df.groupby('Months')['Sales'].sum().idxmin()
```

```
print("\nMinimum sales are done in the month: ",min_sales)
```

#Mean of all the sales mean=df['Sales'].mean()

```
print("\nMean of all the sales is: ",mean)
```

Output:



```
IDLE Shell 3.11.4
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Admin\Desktop\eds\p4.py =====
Maximum sales are done in the month of: JULY
The earning made in that month is: 16000000
The product who sold the most is: Wheat
The city which sold the most products: Asansole
Minimum sales are done in the month: MARCH
Mean of all the sales is: 2685185.185185185
>>>
```

