

Name:-Sagar Patil

Div:-E/E3

Roll no:-544

PRN:-202201090070

jupyter 544\_Sagar Patil Last Checkpoint: an hour ago (autosaved)  Logou

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [25]: `import pandas as pd`  
`df=pd.read_csv("C:/Users/Sagar/Desktop/grainsales.csv")`

In [26]: `#Find the month with the highest sales`  
`best_month = monthly_sales.idxmax()`  
`earnings = monthly_sales.max()`  
`print("The best month for sales was", best_month, "with earnings of",`  
`earnings)`

The best month for sales was JULY with earnings of 16000000

In [27]: `#Group by grain and calculate total sales`  
`grain_sales = df.groupby('GrainName')['Sales'].sum()`  
`best_selling_grain = grain_sales.idxmax()`  
`print("The product that sold the most was", best_selling_grain)`

The product that sold the most was Wheat

In [28]: `#Group by city and calculate total number of products sold`  
`city_sales = df.groupby('City').size()`  
`best_selling_city = city_sales.idxmax()`  
`print("The city that sold the most products was", best_selling_city)`

The city that sold the most products was Nagpur

In [29]: `# min income in which month`  
`earnings = monthly_sales.min()`  
`least_month = monthly_sales.idxmin()`  
`print("The least month for sales was", least_month, "with earnings of",`  
`earnings)`

The least month for sales was MARCH with earnings of 4000000

`#Retrieve the total sales for each grain.`

`total_sales_per_grain = df.groupby('GrainName')['Sales'].sum()`  
`print(total_sales_per_grain)`

GrainName  
Bajra 6000000  
Brown rice 14000000  
Corn 13500000  
Oats 4000000  
Ragi 5000000  
Sattu 5000000  
Sooji 9000000  
Wheat 16000000  
Name: Sales, dtype: int64

`#Find the total sales in Maharashtra`

```
#Find the total sales in Maharashtra.
```

```
sales_in_maharashtra = df[df['State'] == 'Maharashtra']
total_sales_in_maharashtra = sales_in_maharashtra['Sales'].sum()
print("The total sales in Maharashtra:", total_sales_in_maharashtra)
```

The total sales in Maharashtra: 5000000

```
sales_in_may_2023 = df[(df['Months'] == 'MAY') & (df['Year'] == 2023)]
print("Sales data for May 2023:")
print(sales_in_may_2023)
```

Sales data for May 2023:

	GrainName	State	City	Months	Year	Sales
8	Sooji	Tamil Nadu	Madurai	MAY	2023	3000000
16	Sooji	Tamil Nadu	Madurai	MAY	2023	3000000
20	Sooji	Tamil Nadu	Madurai	MAY	2023	3000000

```
#Find the top-selling grain in terms of sales.
```

```
total_sales_per_grain = df.groupby('GrainName')['Sales'].sum()
top_selling_grain = total_sales_per_grain.idxmax()
print("The top-selling grain is:", top_selling_grain)
```

The top-selling grain is: Wheat

```
#Find the average sales for each grain.
```

```
average_sales = df.groupby('GrainName')['Sales'].mean()
print("Average sales for each grain:")
print(average_sales)
```

Average sales for each grain:

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

```
#Retrieve the sales data for oats.
```

```
oats_sales = df[df['GrainName'] == 'Oats']
print("Sales data for oats:")
print(oats_sales)
```

Sales data for oats:

	GrainName	State	City	Months	Year	Sales
6	Oats	Hariyana	Gurugram	MARCH	2023	2000000
14	Oats	Hariyana	Gurugram	MARCH	2023	2000000

```
#Get the sales data for the top three months with the highest sales.
```

```
monthly_sales = df.groupby('Months')['Sales'].sum()
sorted_sales = monthly_sales.sort_values(ascending=False)
top_three_months = sorted_sales.head(3)
```

```
top_three_months = sorted_sales.head(3)
top_three_sales_data = df[df['Months'].isin(top_three_months.index)]
print("Sales data for the top three months with the highest sales:")
print(top_three_sales_data)
```

Sales data for the top three months with the highest sales:

	GrainName	State	City	Months	Year	Sales
9	Brown rice	Telangana	Hyderabad	JUNE	2023	3500000
10	Wheat	West Bengal	Asansole	JULY	2023	4000000
11	Corn	UP	Kanpur	AUG	2023	4500000
17	Brown rice	Telangana	Hyderabad	JUNE	2023	3500000
18	Wheat	West Bengal	Asansole	JULY	2023	4000000
19	Corn	UP	Kanpur	AUG	2023	4500000
21	Brown rice	Telangana	Hyderabad	JUNE	2023	3500000
22	Wheat	West Bengal	Asansole	JULY	2023	4000000
23	Corn	UP	Kanpur	AUG	2023	4500000
25	Brown rice	Telangana	Hyderabad	JUNE	2023	3500000
26	Wheat	West Bengal	Asansole	JULY	2023	4000000

```
#Identify the city with the highest sales.
city_sales = df.groupby('City')['Sales'].sum()
city_with_highest_sales = city_sales.idxmax()
print("City with the highest sales:", city_with_highest_sales)
```

City with the highest sales: Asansole

```
#Get the sales data for the state of Punjab.
sales_in_punjab = df[df['State'] == 'Punjab']
print("Sales data for Punjab:")
print(sales_in_punjab)
```

Sales data for Punjab:  
 Empty DataFrame  
 Columns: [GrainName, State, City, Months, Year, Sales]  
 Index: []

```
#Find the total sales for the year 2023.
sales_2023 = df[df['Year'] == 2023]
total_sales_2023 = sales_2023['Sales'].sum()
print("Total sales for the year 2023:", total_sales_2023)
```

Total sales for the year 2023: 72500000

