Name Vaishnavi kedar

367 c4 batch

202201040045

TITLE –

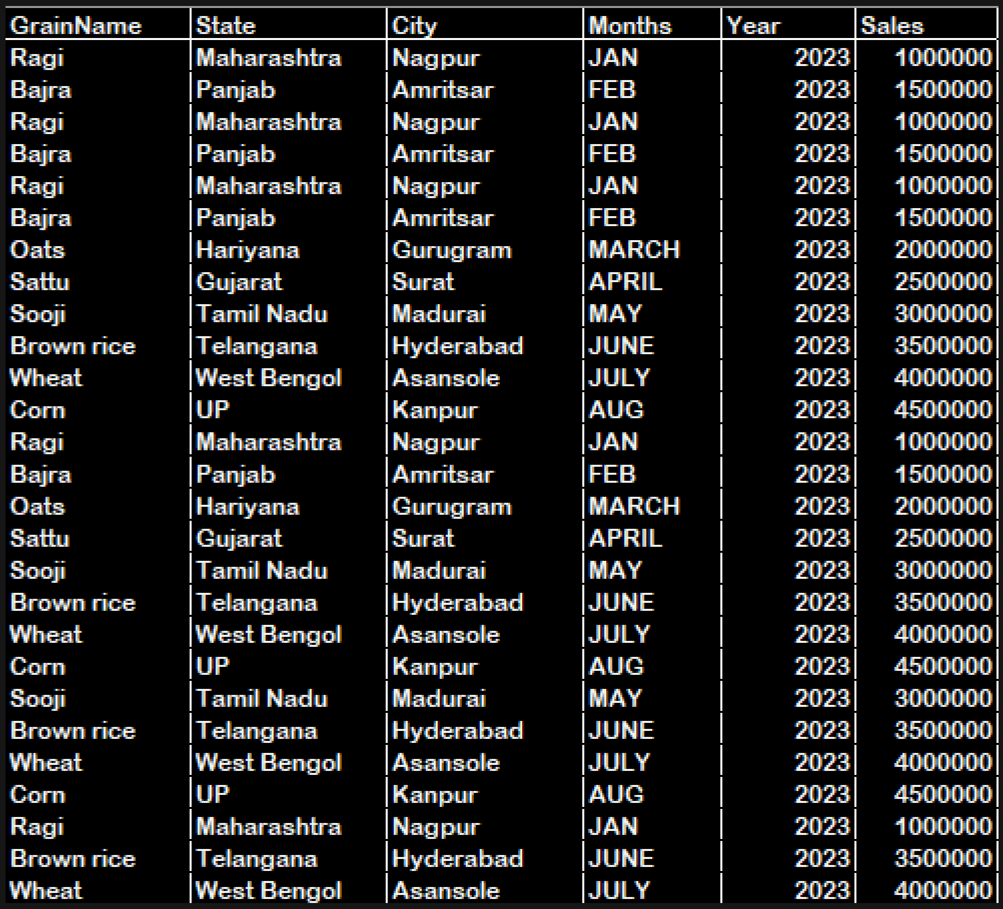
# EDS Practical Assignment no. 4

AIM-

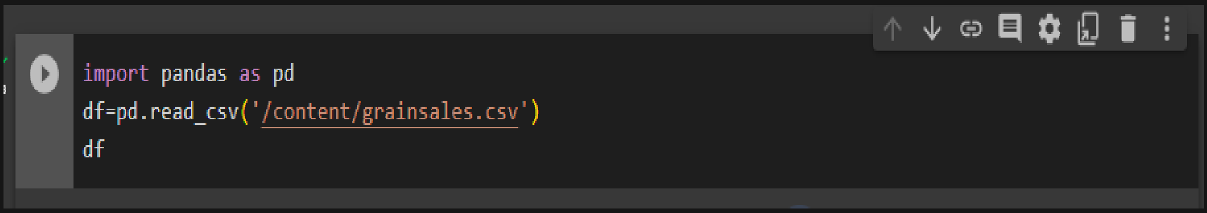
Read any real-life dataset. Store the data in Data Frames. Identify 10 grains for the given dataset.

Implement all 20 grains using Pandas methods

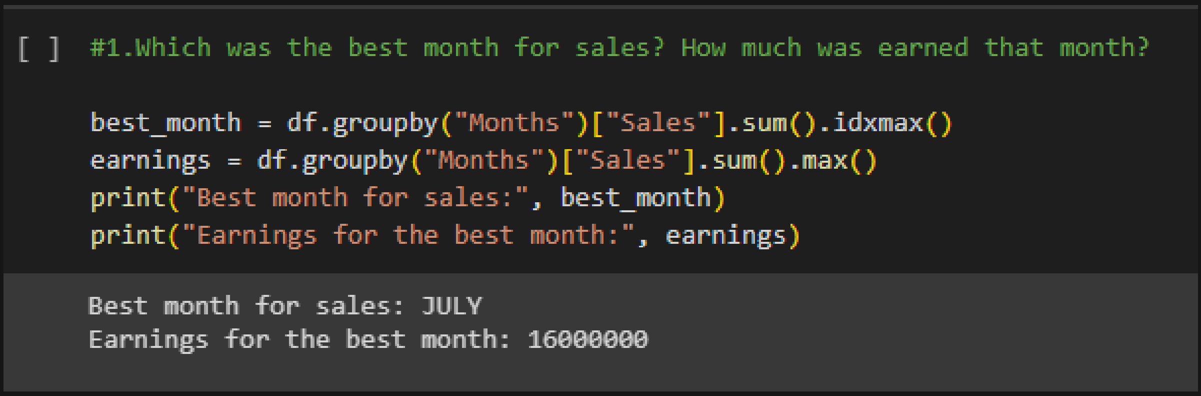
# DATA SHEET



| GrainName | State | City | Months | Year | Sales |
|---|---|---|---|---|---|
| Ragi | Maharashtra | Nagpur | JAN | 2023 | 1000000 |
| Bajra | Panjab | Amritsar | FEB | 2023 | 1500000 |
| Ragi | Maharashtra | Nagpur | JAN | 2023 | 1000000 |
| Bajra | Panjab | Amritsar | FEB | 2023 | 1500000 |
| Ragi | Maharashtra | Nagpur | JAN | 2023 | 1000000 |
| Bajra | Panjab | Amritsar | FEB | 2023 | 1500000 |
| Oats | Hariyana | Gurugram | MARCH | 2023 | 2000000 |
| Sattu | Gujarat | Surat | APRIL | 2023 | 2500000 |
| Sooji | Tamil Nadu | Madurai | MAY | 2023 | 3000000 |
| Brown rice | Telangana | Hyderabad | JUNE | 2023 | 3500000 |
| Wheat | West Bengol | Asansole | JULY | 2023 | 4000000 |
| Corn | UP | Kanpur | AUG | 2023 | 4500000 |
| Ragi | Maharashtra | Nagpur | JAN | 2023 | 1000000 |
| Bajra | Panjab | Amritsar | FEB | 2023 | 1500000 |
| Oats | Hariyana | Gurugram | MARCH | 2023 | 2000000 |
| Sattu | Gujarat | Surat | APRIL | 2023 | 2500000 |
| Sooji | Tamil Nadu | Madurai | MAY | 2023 | 3000000 |
| Brown rice | Telangana | Hyderabad | JUNE | 2023 | 3500000 |
| Wheat | West Bengol | Asansole | JULY | 2023 | 4000000 |
| Corn | UP | Kanpur | AUG | 2023 | 4500000 |
| Sooji | Tamil Nadu | Madurai | MAY | 2023 | 3000000 |
| Brown rice | Telangana | Hyderabad | JUNE | 2023 | 3500000 |
| Wheat | West Bengol | Asansole | JULY | 2023 | 4000000 |
| Corn | UP | Kanpur | AUG | 2023 | 4500000 |
| Ragi | Maharashtra | Nagpur | JAN | 2023 | 1000000 |
| Brown rice | Telangana | Hyderabad | JUNE | 2023 | 3500000 |
| Wheat | West Bengol | Asansole | JULY | 2023 | 4000000 |

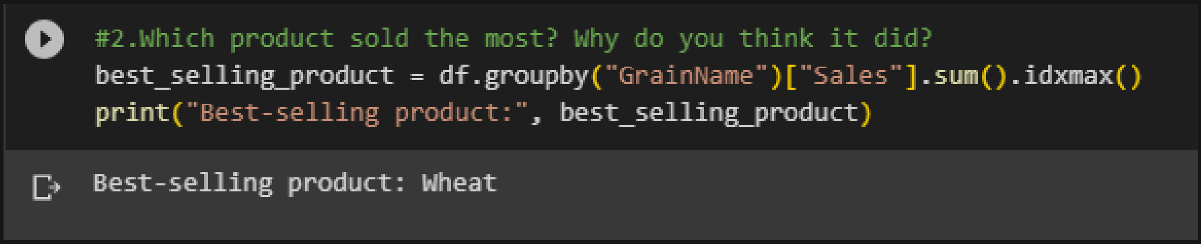


```
import pandas as pd
df=pd.read_csv('/content/grainsales.csv')
df
```



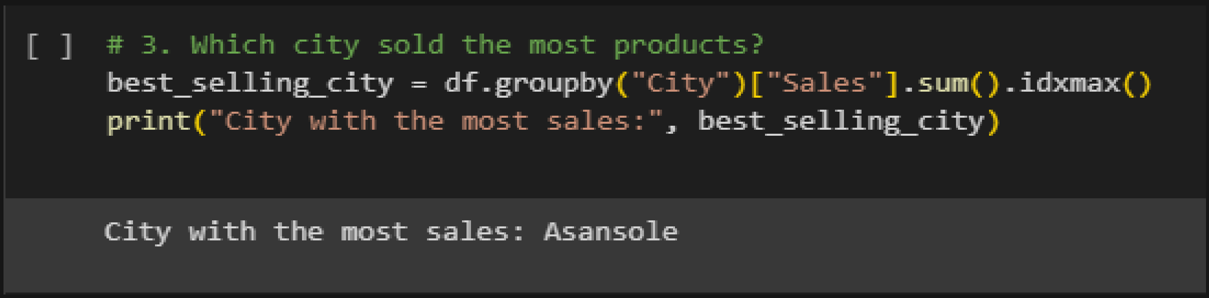
```
[ ]  #1.Which was the best month for sales? How much was earned that month?

     best_month = df.groupby("Months")["Sales"].sum().idxmax()
     earnings = df.groupby("Months")["Sales"].sum().max()
     print("Best month for sales:", best_month)
     print("Earnings for the best month:", earnings)
```

```
Best month for sales: JULY
Earnings for the best month: 16000000
```



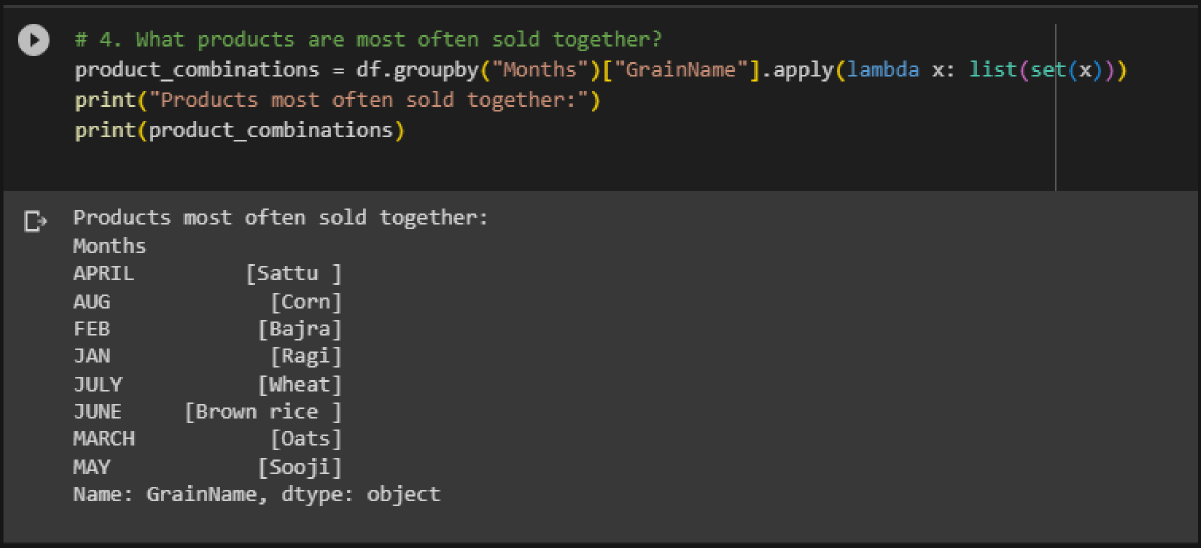
```
#2.Which product sold the most? Why do you think it did?
best_selling_product = df.groupby("GrainName")["Sales"].sum().idxmax()
print("Best-selling product:", best_selling_product)
```

```
Best-selling product: Wheat
```



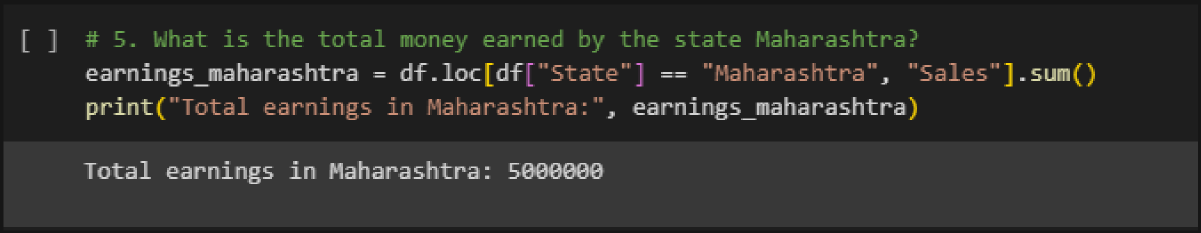
```
[ ]  # 3. Which city sold the most products?
     best_selling_city = df.groupby("City")["Sales"].sum().idxmax()
     print("City with the most sales:", best_selling_city)
```

```
City with the most sales: Asansole
```



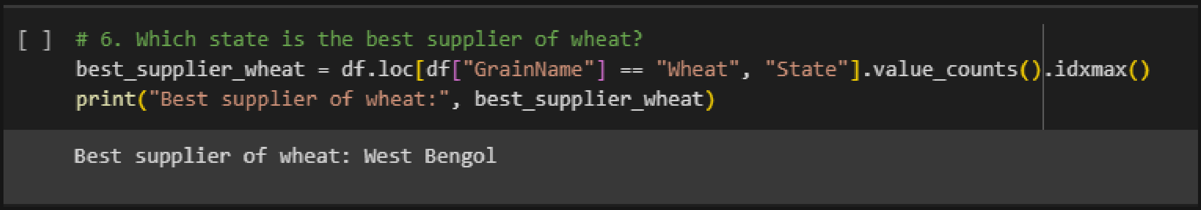
```python
# 4. What products are most often sold together?
product_combinations = df.groupby("Months")["GrainName"].apply(lambda x: list(set(x)))
print("Products most often sold together:")
print(product_combinations)
```

```
Products most often sold together:
Months
APRIL         [Sattu ]
AUG             [Corn]
FEB            [Bajra]
JAN             [Ragi]
JULY           [Wheat]
JUNE     [Brown rice ]
MARCH           [Oats]
MAY            [Sooji]
Name: GrainName, dtype: object
```



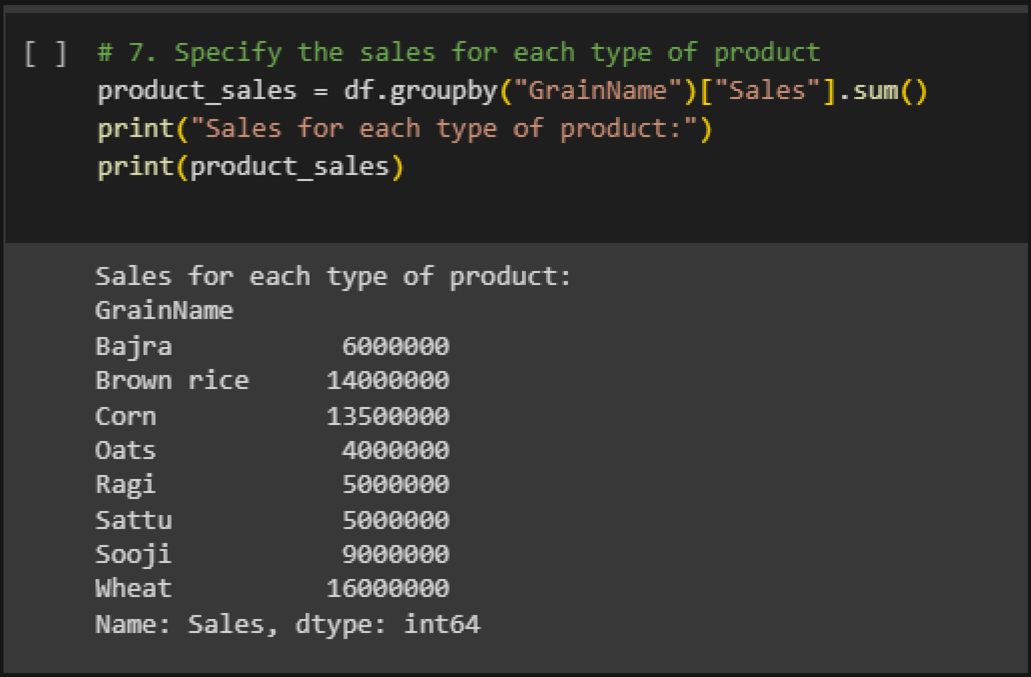
```python
# 5. What is the total money earned by the state Maharashtra?
earnings_maharashtra = df.loc[df["State"] == "Maharashtra", "Sales"].sum()
print("Total earnings in Maharashtra:", earnings_maharashtra)
```

```
Total earnings in Maharashtra: 5000000
```



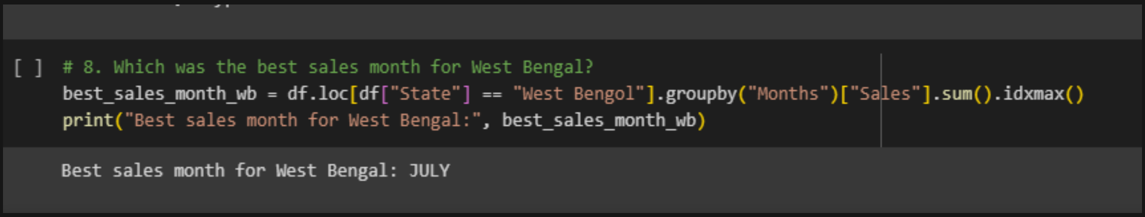
```python
# 6. Which state is the best supplier of wheat?
best_supplier_wheat = df.loc[df["GrainName"] == "Wheat", "State"].value_counts().idxmax()
print("Best supplier of wheat:", best_supplier_wheat)
```

```
Best supplier of wheat: West Bengol
```



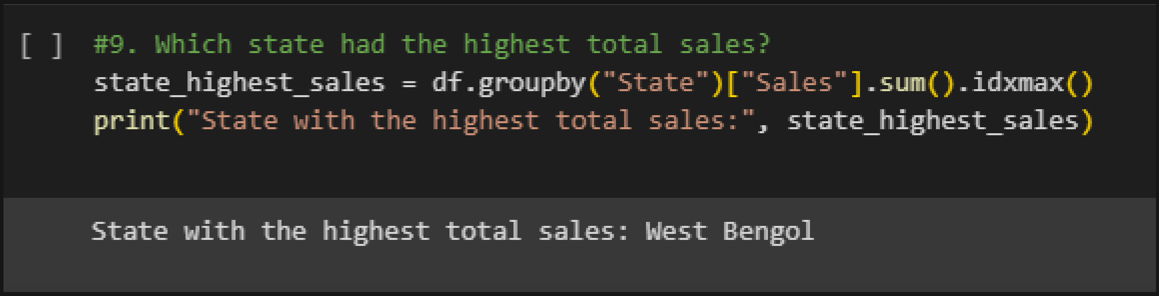
```
[ ] # 7. Specify the sales for each type of product
    product_sales = df.groupby("GrainName")["Sales"].sum()
    print("Sales for each type of product:")
    print(product_sales)
```

```
Sales for each type of product:
GrainName
Bajra           6000000
Brown rice     14000000
Corn           13500000
Oats            4000000
Ragi            5000000
Sattu           5000000
Sooji           9000000
Wheat          16000000
Name: Sales, dtype: int64
```



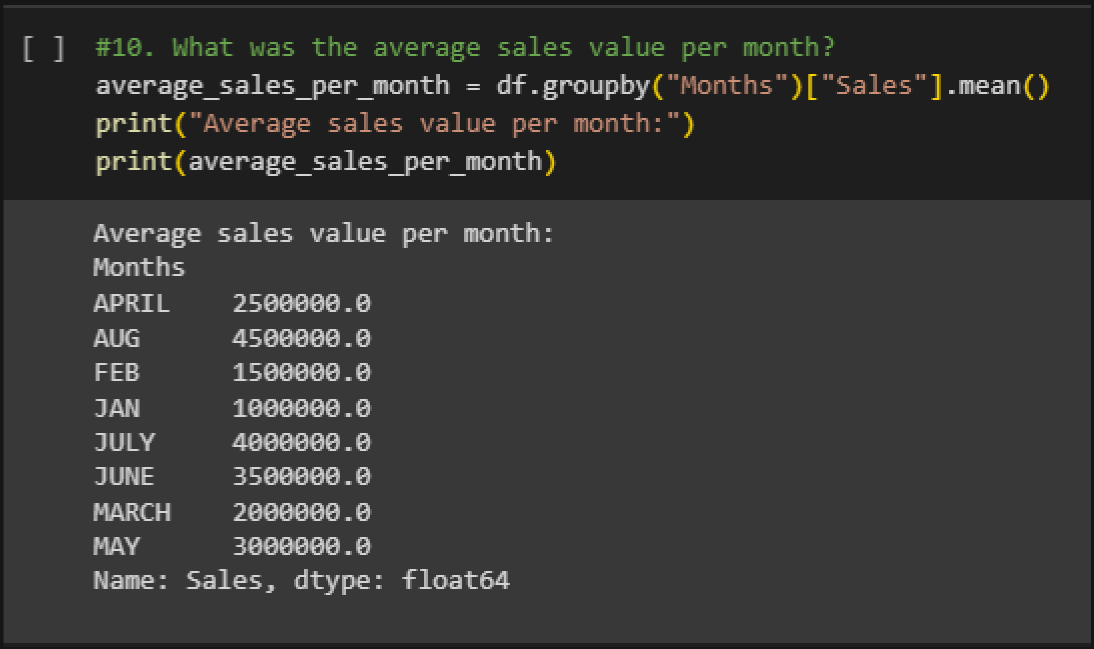
```
[ ] # 8. Which was the best sales month for West Bengal?
    best_sales_month_wb = df.loc[df["State"] == "West Bengol"].groupby("Months")["Sales"].sum().idxmax()
    print("Best sales month for West Bengal:", best_sales_month_wb)
```

```
Best sales month for West Bengal: JULY
```



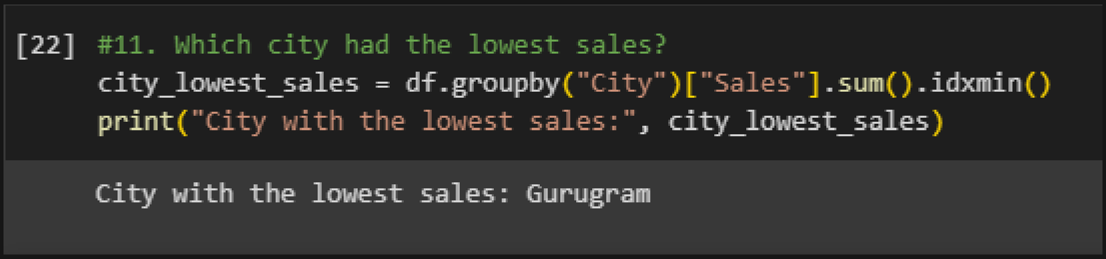
```
[ ] #9. Which state had the highest total sales?
    state_highest_sales = df.groupby("State")["Sales"].sum().idxmax()
    print("State with the highest total sales:", state_highest_sales)
```

```
State with the highest total sales: West Bengol
```



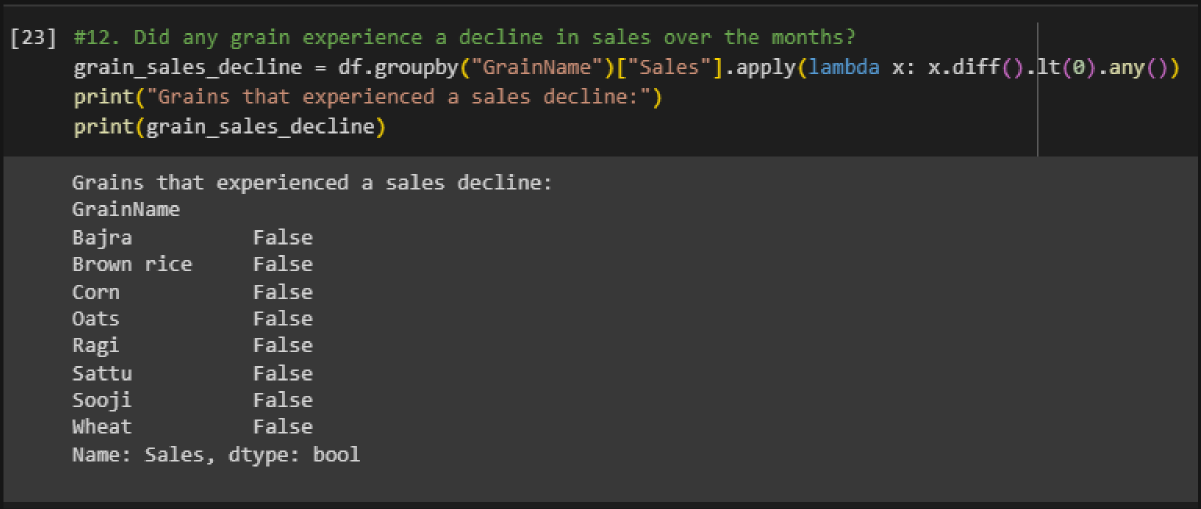
```
[ ]  #10. What was the average sales value per month?
     average_sales_per_month = df.groupby("Months")["Sales"].mean()
     print("Average sales value per month:")
     print(average_sales_per_month)
```

```
Average sales value per month:
Months
APRIL    2500000.0
AUG      4500000.0
FEB      1500000.0
JAN      1000000.0
JULY     4000000.0
JUNE     3500000.0
MARCH    2000000.0
MAY      3000000.0
Name: Sales, dtype: float64
```



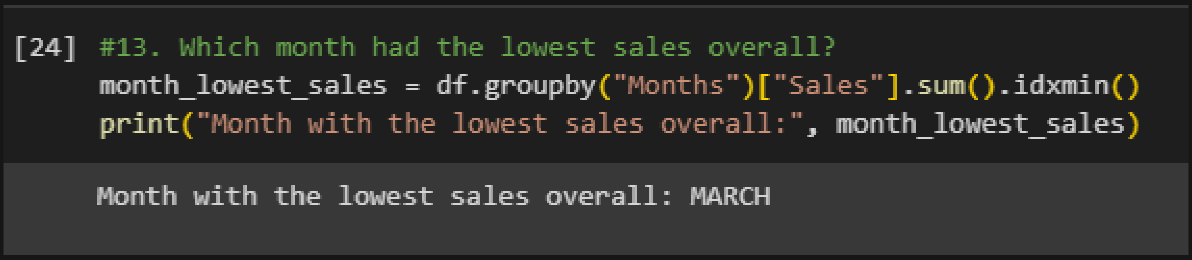
```
[22] #11. Which city had the lowest sales?
     city_lowest_sales = df.groupby("City")["Sales"].sum().idxmin()
     print("City with the lowest sales:", city_lowest_sales)
```

```
City with the lowest sales: Gurugram
```



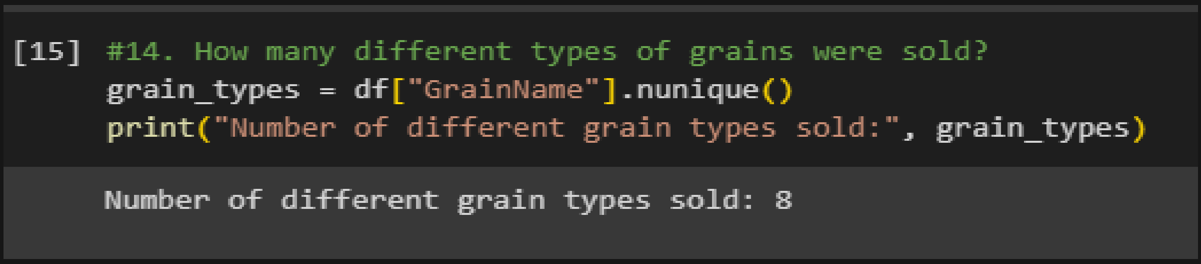
```
[23] #12. Did any grain experience a decline in sales over the months?
     grain_sales_decline = df.groupby("GrainName")["Sales"].apply(lambda x: x.diff().lt(0).any())
     print("Grains that experienced a sales decline:")
     print(grain_sales_decline)
```

```
Grains that experienced a sales decline:
GrainName
Bajra         False
Brown rice    False
Corn          False
Oats          False
Ragi          False
Sattu         False
Sooji         False
Wheat         False
Name: Sales, dtype: bool
```



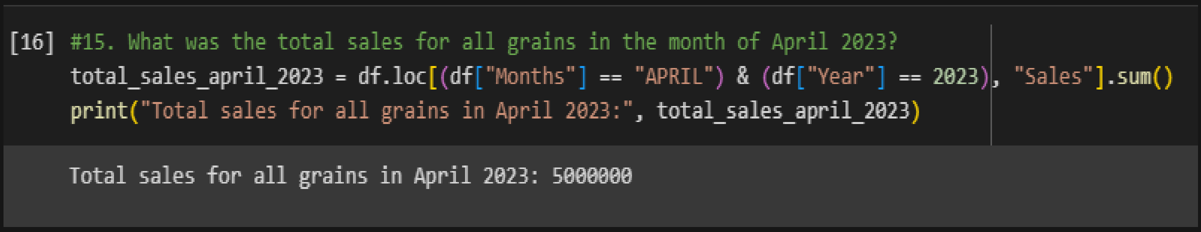
```
[24] #13. Which month had the lowest sales overall?
     month_lowest_sales = df.groupby("Months")["Sales"].sum().idxmin()
     print("Month with the lowest sales overall:", month_lowest_sales)
```

```
Month with the lowest sales overall: MARCH
```



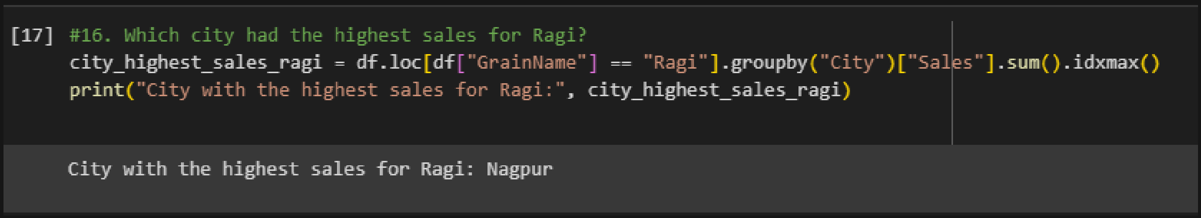
```
[15] #14. How many different types of grains were sold?
     grain_types = df["GrainName"].nunique()
     print("Number of different grain types sold:", grain_types)
```

```
Number of different grain types sold: 8
```



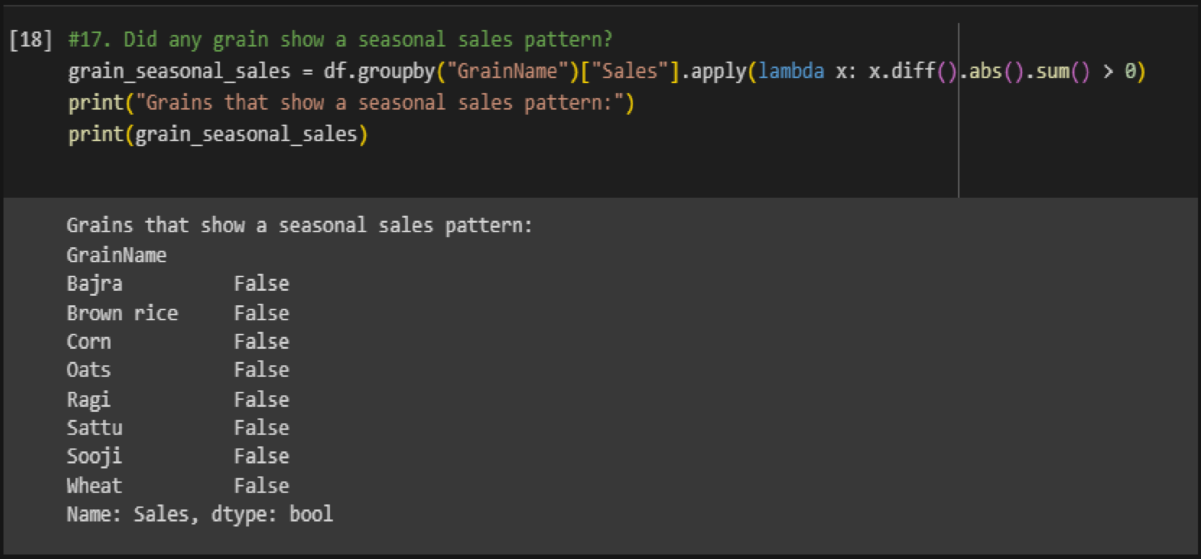
```
[16] #15. What was the total sales for all grains in the month of April 2023?
     total_sales_april_2023 = df.loc[(df["Months"] == "APRIL") & (df["Year"] == 2023), "Sales"].sum()
     print("Total sales for all grains in April 2023:", total_sales_april_2023)
```

```
Total sales for all grains in April 2023: 5000000
```



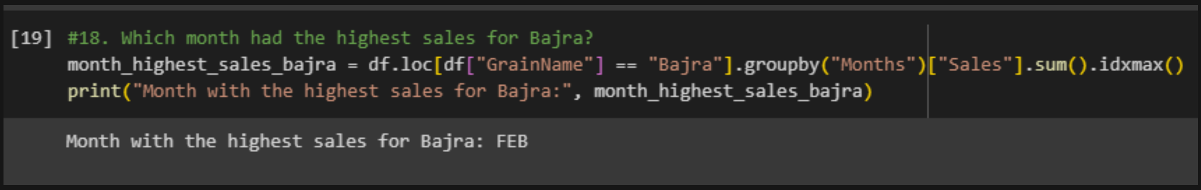
```
[17] #16. Which city had the highest sales for Ragi?
     city_highest_sales_ragi = df.loc[df["GrainName"] == "Ragi"].groupby("City")["Sales"].sum().idxmax()
     print("City with the highest sales for Ragi:", city_highest_sales_ragi)
```

```
City with the highest sales for Ragi: Nagpur
```

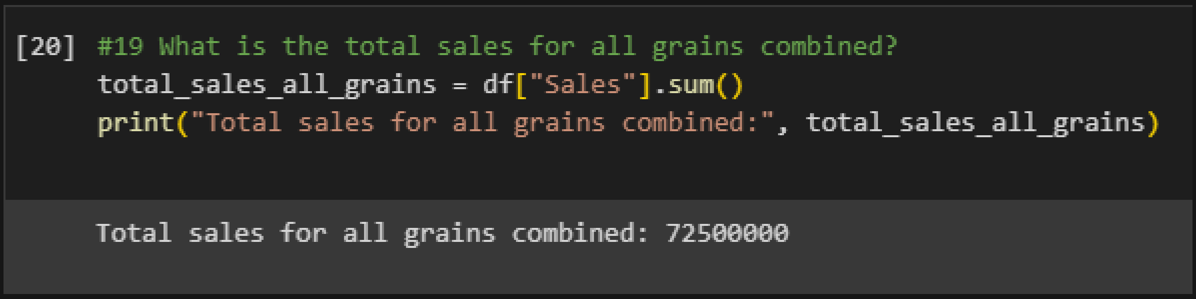


```
[18] #17. Did any grain show a seasonal sales pattern?
     grain_seasonal_sales = df.groupby("GrainName")["Sales"].apply(lambda x: x.diff().abs().sum() > 0)
     print("Grains that show a seasonal sales pattern:")
     print(grain_seasonal_sales)
```

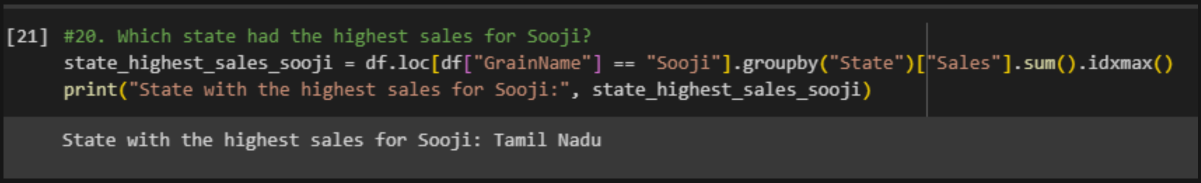
```
Grains that show a seasonal sales pattern:
GrainName
Bajra         False
Brown rice    False
Corn          False
Oats          False
Ragi          False
Sattu         False
Sooji         False
Wheat         False
Name: Sales, dtype: bool
```



```
[19] #18. Which month had the highest sales for Bajra?
     month_highest_sales_bajra = df.loc[df["GrainName"] == "Bajra"].groupby("Months")["Sales"].sum().idxmax()
     print("Month with the highest sales for Bajra:", month_highest_sales_bajra)

     Month with the highest sales for Bajra: FEB
```



```
[20] #19 What is the total sales for all grains combined?
     total_sales_all_grains = df["Sales"].sum()
     print("Total sales for all grains combined:", total_sales_all_grains)

     Total sales for all grains combined: 72500000
```



```
[21] #20. Which state had the highest sales for Sooji?
     state_highest_sales_sooji = df.loc[df["GrainName"] == "Sooji"].groupby("State")["Sales"].sum().idxmax()
     print("State with the highest sales for Sooji:", state_highest_sales_sooji)

     State with the highest sales for Sooji: Tamil Nadu
```