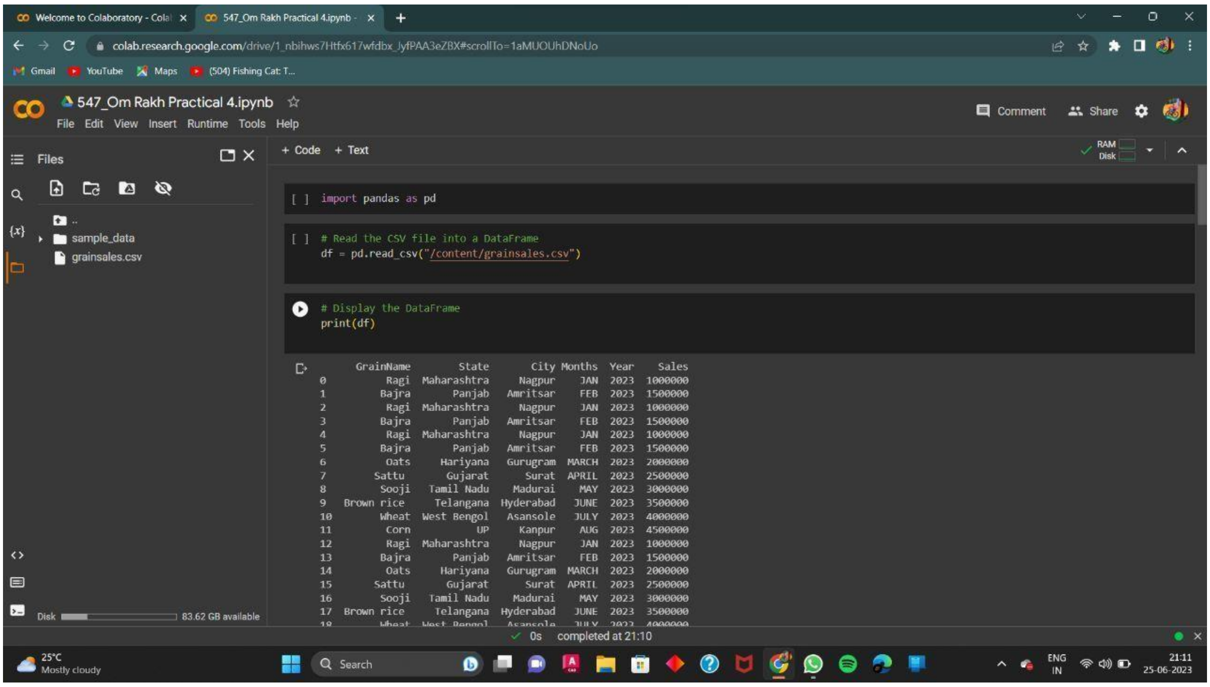
Name : Om Rakh Roll no. :547 Div : E3 Sub : EDS

PRACTICAL 4

```
import pandas as pd


# Read the CSV file into a DataFrame df =
pd.read_csv("Downloads\\grainsales.csv")


# Display the DataFrame print(df)
# Group by month and calculate total sales monthly_sales
= df.groupby('Months')['Sales'].sum()


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


# Calculate the earnings for the best month earnings
= monthly_sales.max()


# Print the result
print("The best month for sales was", best_month, "with earnings of", earnings)
# Group by grain and calculate total sales
grain_sales = df.groupby('GrainName')['Sales'].sum()


# Find the grain with the highest sales best_selling_grain
= grain_sales.idxmax()
```

```
# Print the result
print("The product that sold the most was", best_selling_grain)

# Group by city and calculate total number of products sold city_sales
= df.groupby('City').size()

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

# Print the result
print("The city that sold the most products was", best_selling_city)

# 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)

#least sell product
least_selling_grain = grain_sales.idxmin()
print("The product that sold the most was", least_selling_grain) #earning of
each states
sales_state=df.groupby('State')['Sales'].sum() print(sales_state)

#max income of state  earning=sales_state.max()
state_earnmore=sales_state.idxmax()
print ("state have max  sales earning",state_earnmore,"with earning",earning)
```

```
earning=sales_state.min() state_earnless=sales_state.idxmin()
print ("state have max  sales earning",state_earnless,"with earning",earning)
#Calculate the average sales per city: average_sales_per_city =
df.groupby('City')['Sales'].mean() print(average_sales_per_city)
#Find the total sales for each state and sort in descending order:
state_sales = df.groupby('State')['Sales'].sum().sort_values(ascending=False)
print(state_sales)

#Filter the data for the month of January: january_data
= df[df['Months'] == 'JAN'] print(january_data)
```

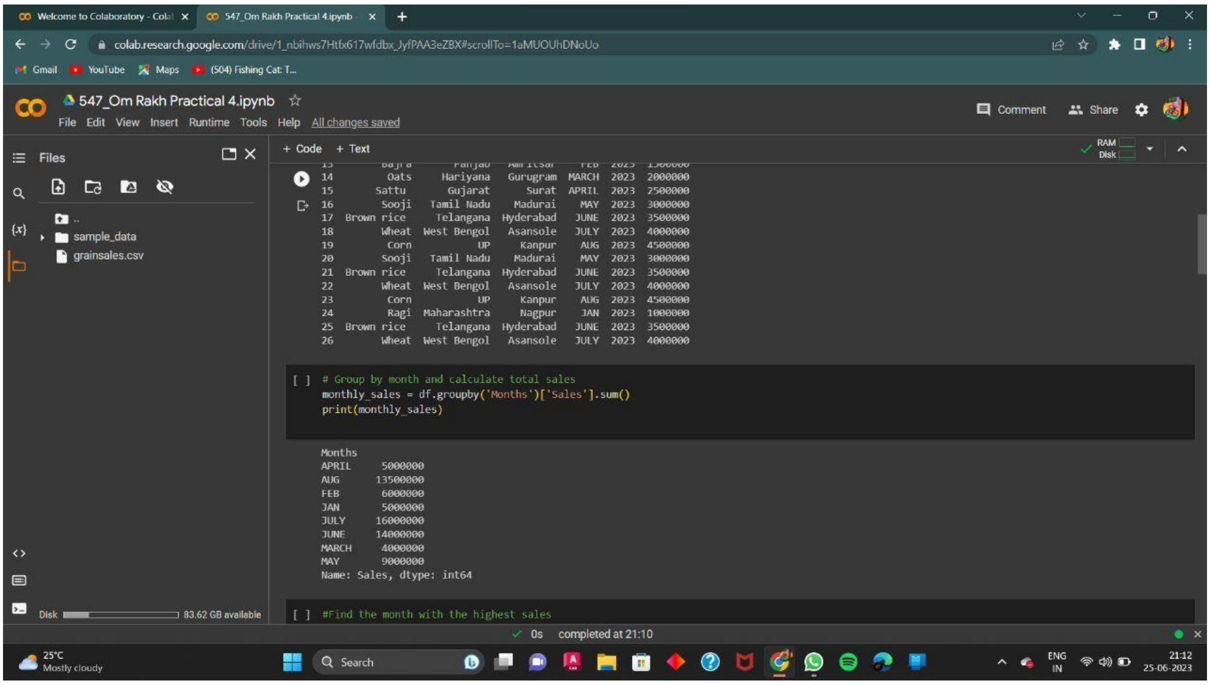

Welcome to Colaboratory - Colal
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_JyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
Files
+ Code
+ Text
RAM
Disk
sample_data
grainsales.csv
Disk
83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:11
25-06-2023

```
[ ] import pandas as pd
```

```
[ ] # Read the CSV file into a DataFrame
    df = pd.read_csv("/content/grainsales.csv")
```

```
# Display the DataFrame
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  2023  1000000
3         Bajra       Panjab   Amritsar    FEB  2023  1500000
4          Ragi  Maharashtra     Nagpur    JAN  2023  1000000
5         Bajra       Panjab   Amritsar    FEB  2023  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  2023  4000000
11         Corn           UP     Kanpur    AUG  2023  4500000
12         Ragi  Maharashtra     Nagpur    JAN  2023  1000000
13        Bajra       Panjab   Amritsar    FEB  2023  1500000
14         Oats     Hariyana   Gurugram  MARCH  2023  2000000
15       Sattu       Gujarat      Surat  APRIL  2023  2500000
16        Sooji   Tamil Nadu    Madurai    MAY  2023  3000000
17   Brown rice    Telangana  Hyderabad   JUNE  2023  3500000
18        Wheat  West Bengol   Asansole   JULY  2023  4000000
```

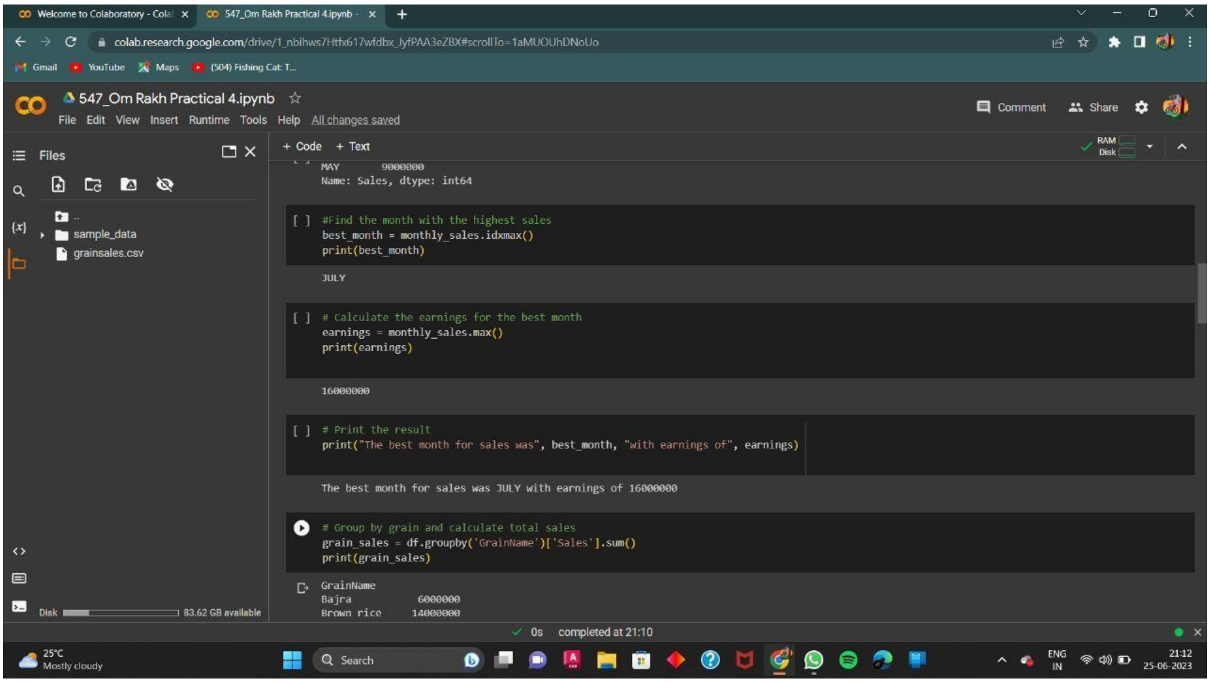

Welcome to Colaboratory - Colal
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_JyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code
+ Text
RAM
Disk
sample_data
grainsales.csv
Disk
83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:12
25-06-2023

```
13        Bajra       Panjab   Amritsar    FEB  2023  1500000
14         Oats     Hariyana   Gurugram  MARCH  2023  2000000
15       Sattu       Gujarat      Surat  APRIL  2023  2500000
16        Sooji   Tamil Nadu    Madurai    MAY  2023  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  2023  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  2023  1000000
25   Brown rice    Telangana  Hyderabad   JUNE  2023  3500000
26        Wheat  West Bengol   Asansole   JULY  2023  4000000
```

```
[ ] # Group by month and calculate total sales
    monthly_sales = df.groupby('Months')['Sales'].sum()
    print(monthly_sales)
```

```
Months
APRIL     5000000
AUG      13500000
FEB       6000000
JAN       5000000
JULY     16000000
JUNE     14000000
MARCH     4000000
MAY       9000000
Name: Sales, dtype: int64
```

```
[ ] #Find the month with the highest sales
```



Welcome to Colaboratory - Cola
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_JyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
sample_data
grainsales.csv
Disk 83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:12
25-06-2023

```
MAY         9000000
Name: Sales, dtype: int64
```

```
#Find the month with the highest sales
best_month = monthly_sales.idxmax()
print(best_month)
```

```
JULY
```

```
# Calculate the earnings for the best month
earnings = monthly_sales.max()
print(earnings)
```

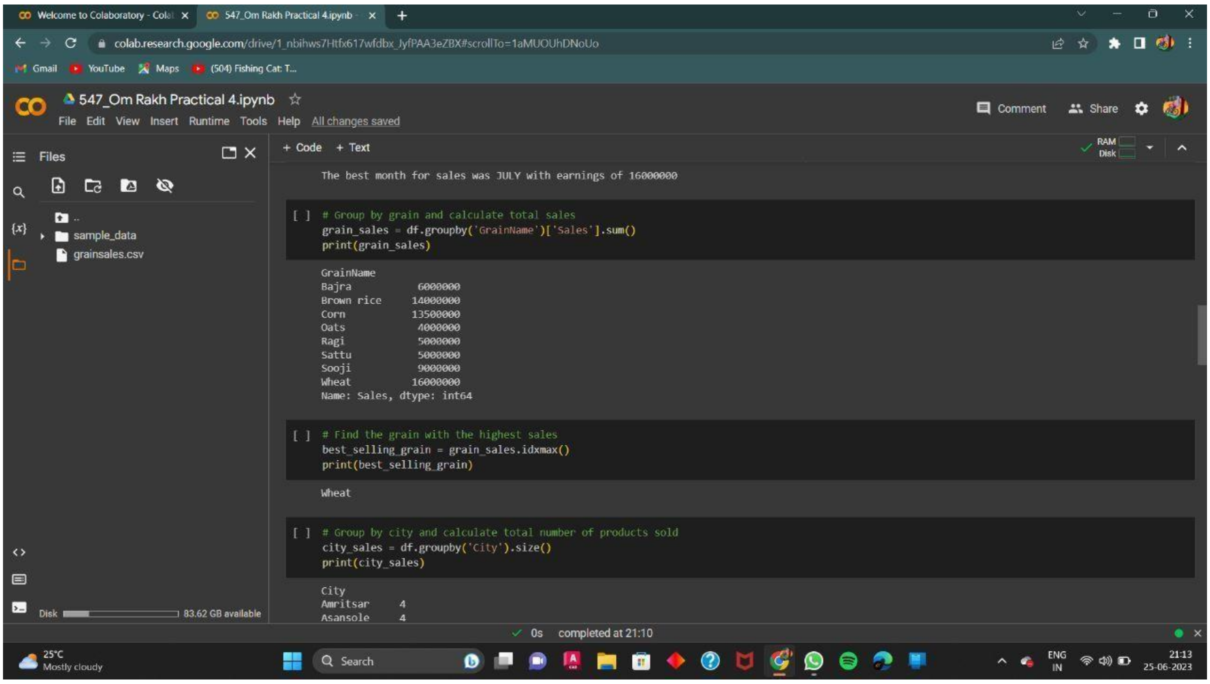
```
16000000
```

```
# Print the result
print("The best month for sales was", best_month, "with earnings of", earnings)
```

```
The best month for sales was JULY with earnings of 16000000
```

```
# Group by grain and calculate total sales
grain_sales = df.groupby('GrainName')['Sales'].sum()
print(grain_sales)
```

```
GrainName
Bajra           6000000
Brown rice     14000000
```



Welcome to Colaboratory - Cola
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_JyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
sample_data
grainsales.csv
Disk 83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:13
25-06-2023

```
The best month for sales was JULY with earnings of 16000000
```

```
# Group by grain and calculate total sales
grain_sales = df.groupby('GrainName')['Sales'].sum()
print(grain_sales)
```

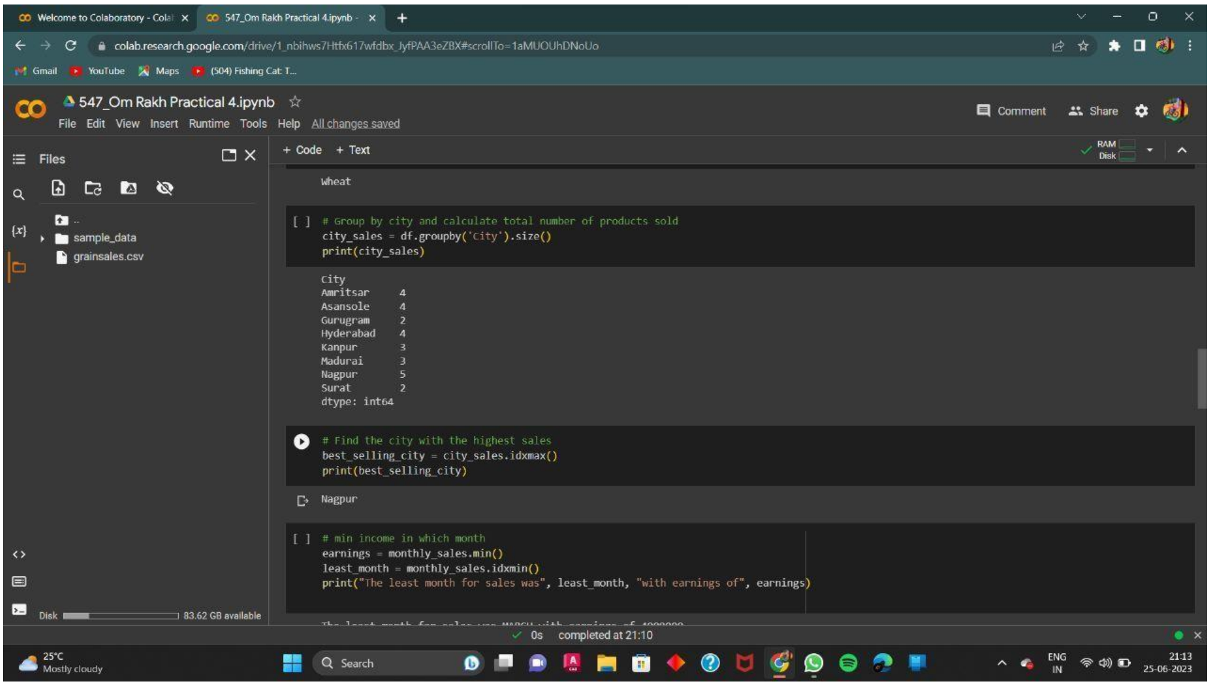
```
GrainName
Bajra           6000000
Brown rice     14000000
Corn           13500000
Oats            4000000
Ragi            5000000
Sattu           5000000
Sooji           9000000
Wheat          16000000
Name: Sales, dtype: int64
```

```
# Find the grain with the highest sales
best_selling_grain = grain_sales.idxmax()
print(best_selling_grain)
```

```
Wheat
```

```
# Group by city and calculate total number of products sold
city_sales = df.groupby('City').size()
print(city_sales)
```

```
City
Amritsar    4
Asansole    4
```


Welcome to Colaboratory - Colal
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_lyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
sample_data
grainsales.csv
Disk 83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:13
25-06-2023

```
Wheat
```

```
# Group by city and calculate total number of products sold
city_sales = df.groupby('City').size()
print(city_sales)
```

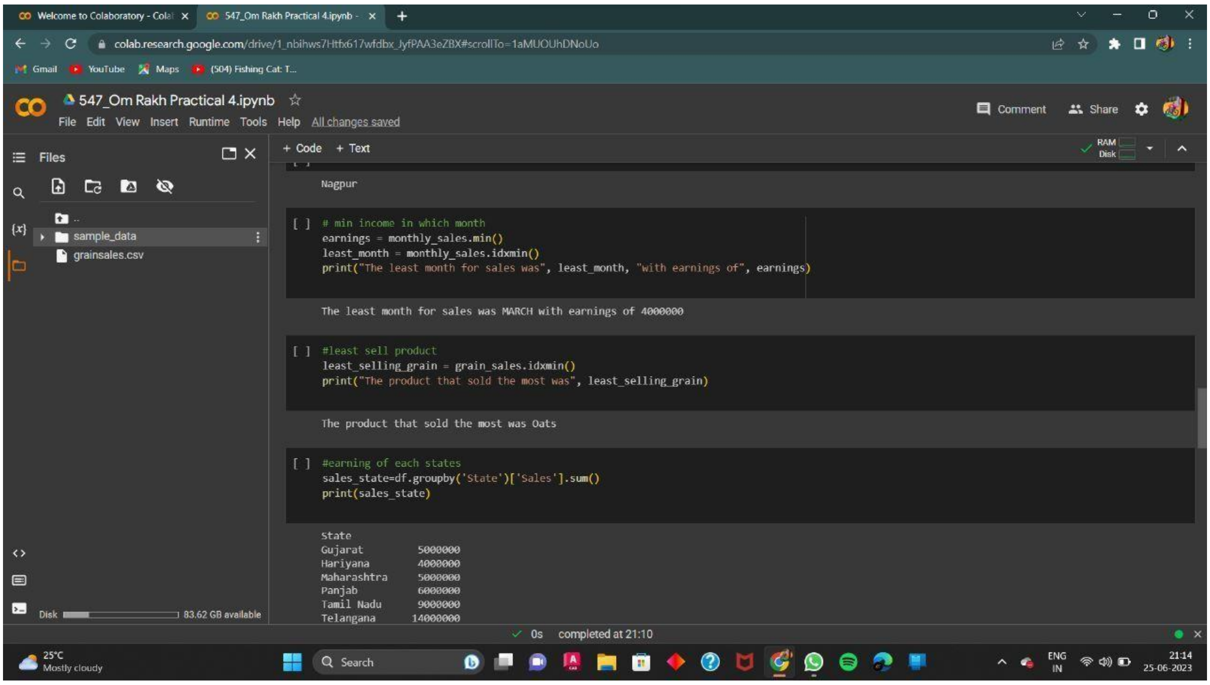
```
City
Amritsar     4
Asansole     4
Gurugram     2
Hyderabad    4
Kanpur       3
Madurai      3
Nagpur       5
Surat        2
dtype: int64
```

```
# Find the city with the highest sales
best_selling_city = city_sales.idxmax()
print(best_selling_city)
```

```
Nagpur
```

```
# 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
```


Welcome to Colaboratory - Colal
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_lyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
sample_data
grainsales.csv
Disk 83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:14
25-06-2023

```
Nagpur
```

```
# 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)
```

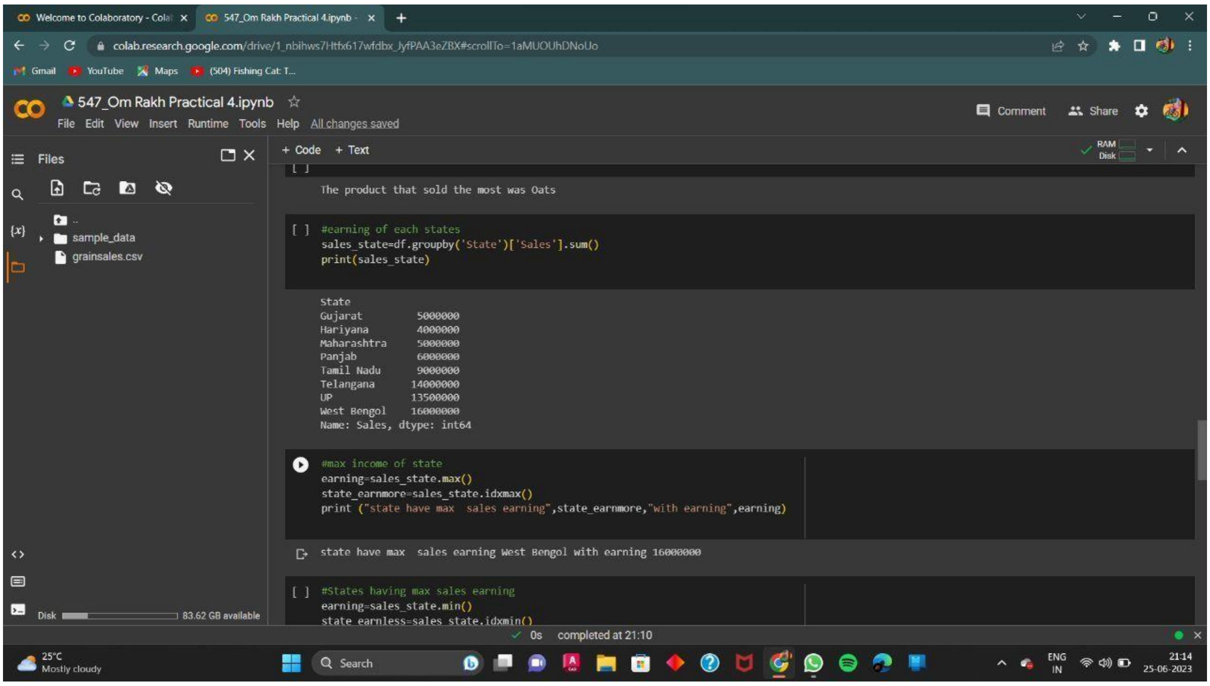
```
The least month for sales was MARCH with earnings of 4000000
```

```
#least sell product
least_selling_grain = grain_sales.idxmin()
print("The product that sold the most was", least_selling_grain)
```

```
The product that sold the most was Oats
```

```
#earning of each states
sales_state=df.groupby('State')['Sales'].sum()
print(sales_state)
```

```
State
Gujarat         5000000
Hariyana        4000000
Maharashtra     5000000
Panjab          6000000
Tamil Nadu      9000000
Telangana      14000000
```


Welcome to Colaboratory - Cola
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_lyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
sample_data
grainsales.csv
Disk 83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:14
25-06-2023

```
The product that sold the most was Oats
```

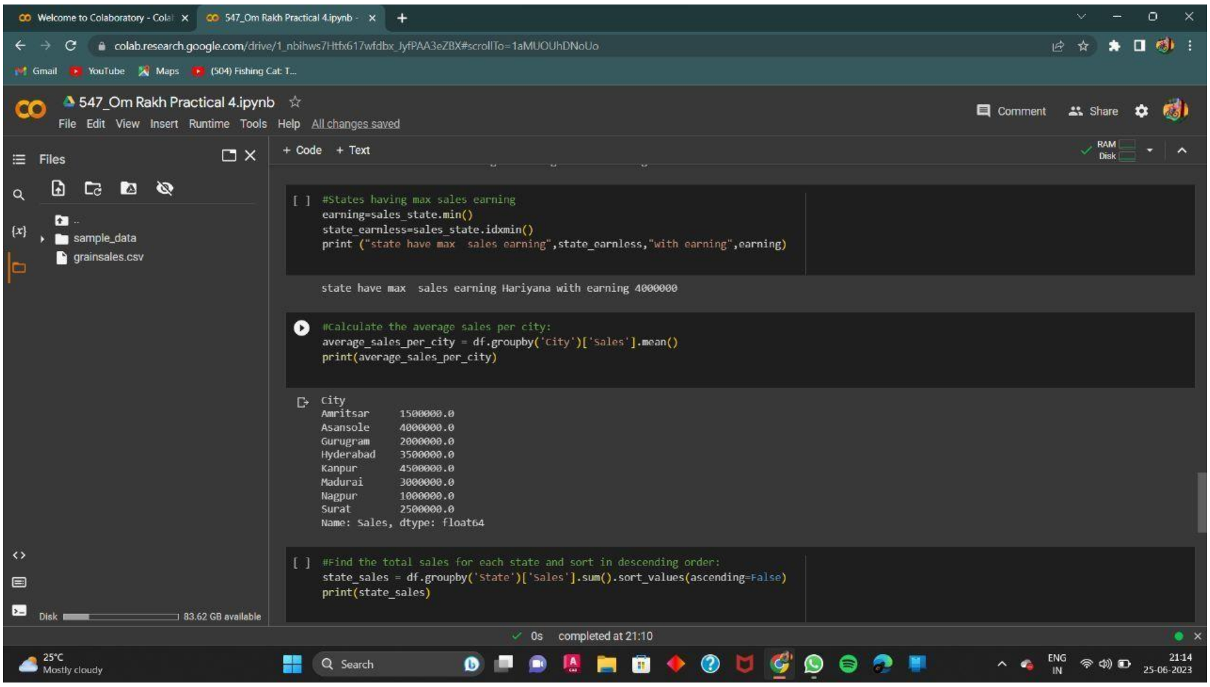
```
#earning of each states
sales_state=df.groupby('State')['Sales'].sum()
print(sales_state)
```

```
State
Gujarat         5000000
Hariyana        4000000
Maharashtra     5000000
Panjab          6000000
Tamil Nadu      9000000
Telangana      14000000
UP             13500000
West Bengol    16000000
Name: Sales, dtype: int64
```

```
#max income of state
earning=sales_state.max()
state_earnmore=sales_state.idxmax()
print ("state have max  sales earning",state_earnmore,"with earning",earning)
```

```
state have max  sales earning West Bengol with earning 16000000
```

```
#States having max sales earning
earning=sales_state.min()
state_earnless=sales_state.idxmin()
```


Welcome to Colaboratory - Cola
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_lyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
+ Code + Text
RAM
Disk
sample_data
grainsales.csv
Disk 83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:14
25-06-2023

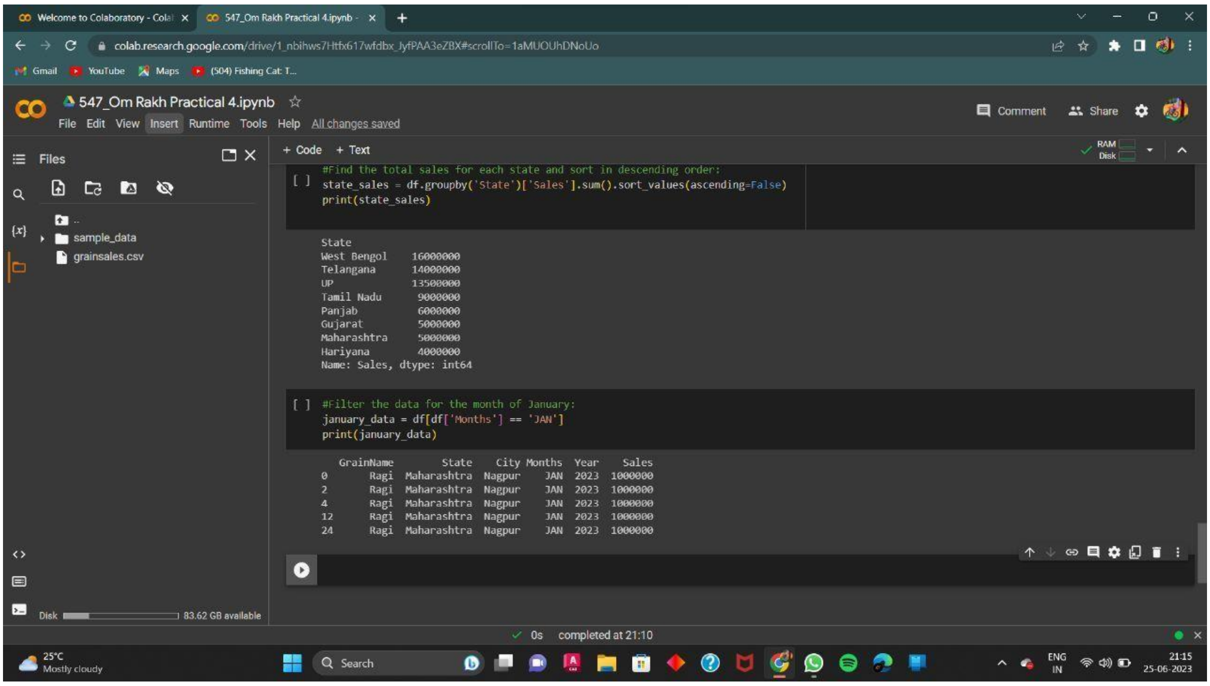
```
#States having max sales earning
earning=sales_state.min()
state_earnless=sales_state.idxmin()
print ("state have max  sales earning",state_earnless,"with earning",earning)
```

```
state have max  sales earning Hariyana with earning 4000000
```

```
#Calculate the average sales per city:
average_sales_per_city = df.groupby('City')['Sales'].mean()
print(average_sales_per_city)
```

```
City
Amritsar     1500000.0
Asansole     4000000.0
Gurugram     2000000.0
Hyderabad    3500000.0
Kanpur       4500000.0
Madurai      3000000.0
Nagpur       1000000.0
Surat        2500000.0
Name: Sales, dtype: float64
```

```
#Find the total sales for each state and sort in descending order:
state_sales = df.groupby('State')['Sales'].sum().sort_values(ascending=False)
print(state_sales)
```

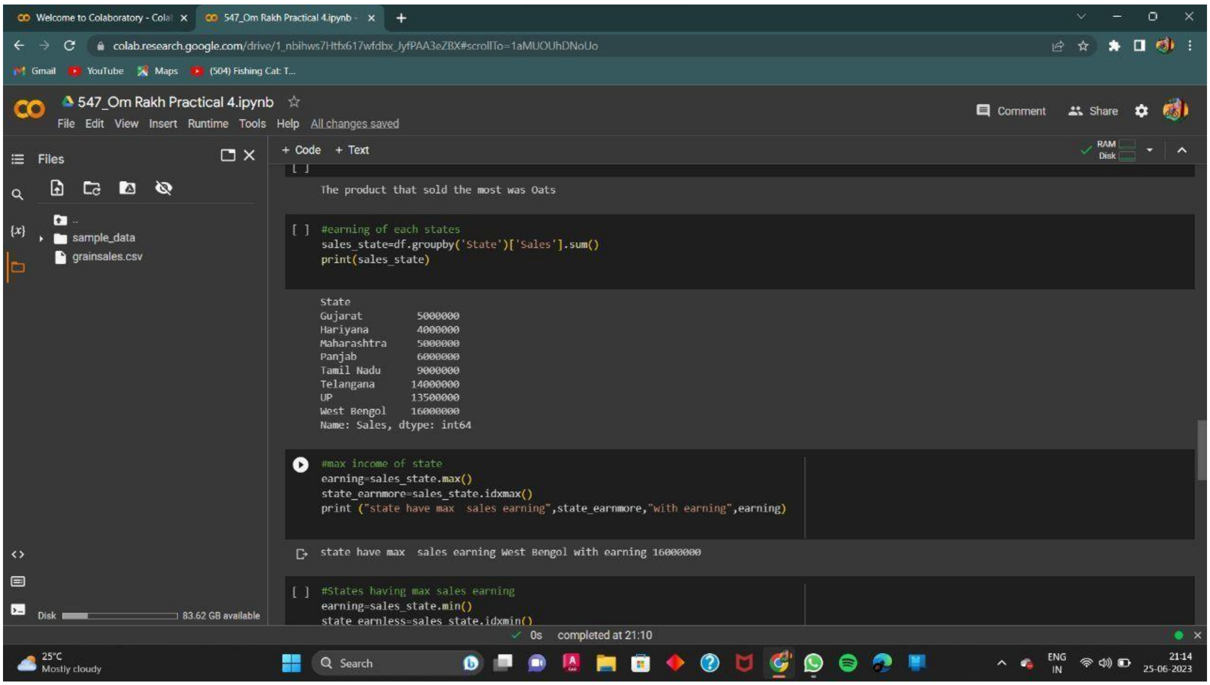


```
#Find the total sales for each state and sort in descending order:
state_sales = df.groupby('State')['Sales'].sum().sort_values(ascending=False)
print(state_sales)
```

```
State
West Bengol    16000000
Telangana      14000000
UP             13500000
Tamil Nadu      9000000
Panjab          6000000
Gujarat         5000000
Maharashtra     5000000
Hariyana        4000000
Name: Sales, dtype: int64
```

```
#Filter the data for the month of January:
january_data = df[df['Months'] == 'JAN']
print(january_data)
```

```
   GrainName        State    City Months  Year    Sales
0       Ragi  Maharashtra  Nagpur    JAN  2023  1000000
2       Ragi  Maharashtra  Nagpur    JAN  2023  1000000
4       Ragi  Maharashtra  Nagpur    JAN  2023  1000000
12      Ragi  Maharashtra  Nagpur    JAN  2023  1000000
24      Ragi  Maharashtra  Nagpur    JAN  2023  1000000
```



Welcome to Colaboratory - Cola
547_Om Rakh Practical 4.ipynb
colab.research.google.com/drive/1_nbihws7Htfx617wfdbx_JyfPAA3eZBX#scrollTo=1aMUOUhDNoUo
Gmail
YouTube
Maps
(504) Fishing Cat: T...
547_Om Rakh Practical 4.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
Files
sample_data
grainsales.csv
+ Code + Text
RAM
Disk
Disk 83.62 GB available
0s completed at 21:10
25°C
Mostly cloudy
Search
ENG
IN
21:14
25-06-2023

```
The product that sold the most was Oats
```

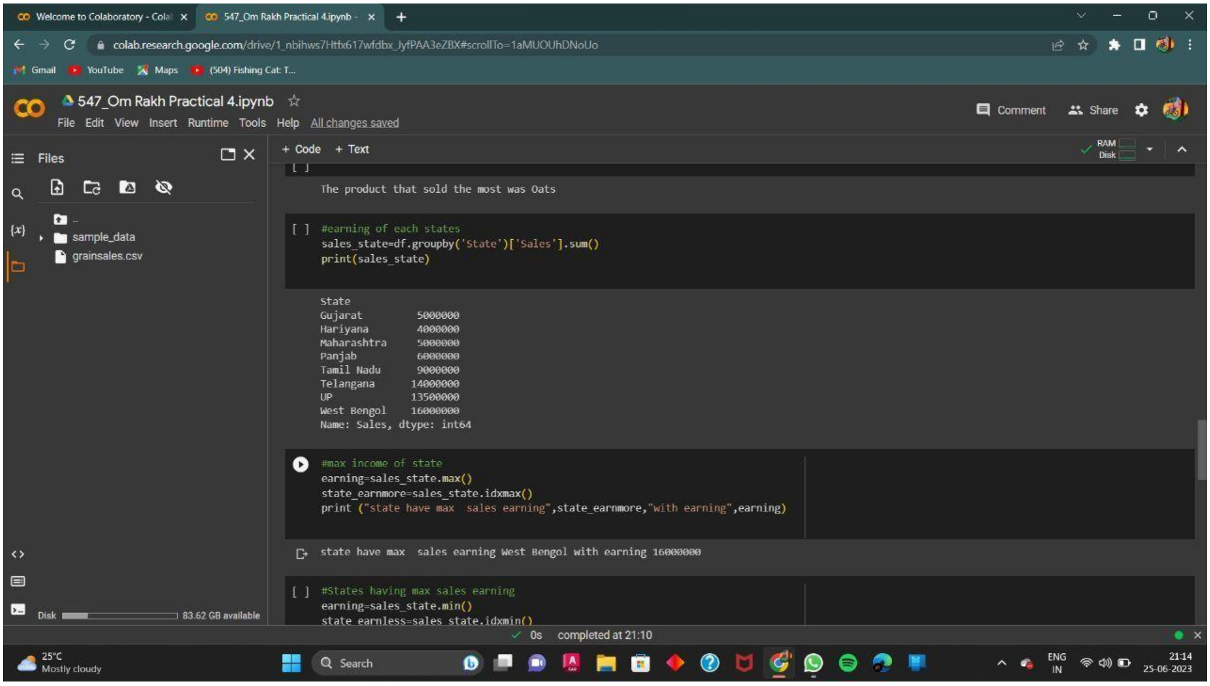
```
[ ] #earning of each states
    sales_state=df.groupby('State')['Sales'].sum()
    print(sales_state)
```

```
State
Gujarat         5000000
Hariyana        4000000
Maharashtra     5000000
Panjab          6000000
Tamil Nadu      9000000
Telangana      14000000
UP             13500000
West Bengol    16000000
Name: Sales, dtype: int64
```

```
#max income of state
earning=sales_state.max()
state_earnmore=sales_state.idxmax()
print ("state have max  sales earning",state_earnmore,"with earning",earning)
```

```
state have max  sales earning West Bengol with earning 16000000
```

```
[ ] #States having max sales earning
    earning=sales_state.min()
    state_earnless=sales_state.idxmin()
```



```
The product that sold the most was Oats
```

```
[ ] #earning of each states
    sales_state=df.groupby('State')['Sales'].sum()
    print(sales_state)
```

```
State
Gujarat         5000000
Hariyana        4000000
Maharashtra     5000000
Panjab          6000000
Tamil Nadu      9000000
Telangana      14000000
UP             13500000
West Bengol    16000000
Name: Sales, dtype: int64
```

```
#max income of state
earning=sales_state.max()
state_earnmore=sales_state.idxmax()
print ("state have max  sales earning",state_earnmore,"with earning",earning)
```

```
state have max  sales earning West Bengol with earning 16000000
```

```
[ ] #States having max sales earning
    earning=sales_state.min()
    state_earnless=sales_state.idxmin()
```