

NAME:- ANSHUM MANKAR

ROLL NO. :-CS3-66

PRN:-202401040078

DIVISION:- CS3

BATCH:- C33

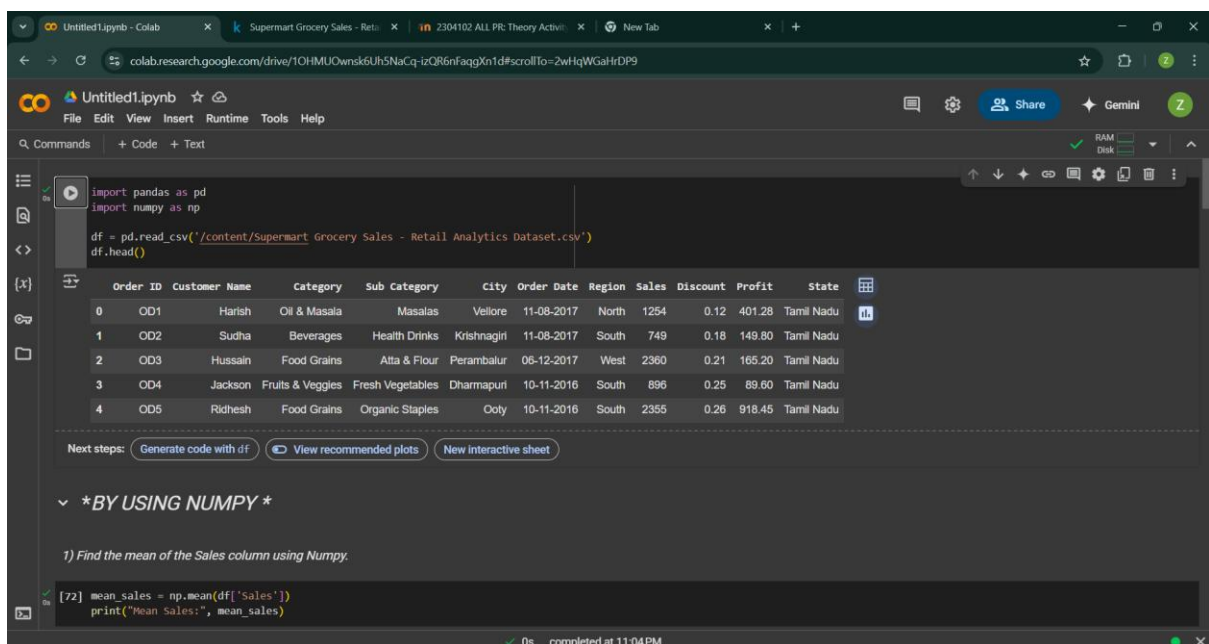
LINK OF THE COLAB NOTE-BOOK:-

<https://colab.research.google.com/drive/1OHMUOwnsk6Uh5NaCq-izQR6nFaggXn1d?usp=sharing>

LINK OF THE DATA-SET:- <https://www.kaggle.com/datasets/mohamedharris/supermart-grocery-sales-retail-analytics-dataset>

TOPIC:- Grocery

First importing Numpy and Pandas



```
import pandas as pd
import numpy as np

df = pd.read_csv('/content/Supermart Grocery Sales - Retail Analytics Dataset.csv')
df.head()
```

	Order ID	Customer Name	Category	Sub Category	City	Order Date	Region	Sales	Discount	Profit	State
0	OD1	Harish	Oil & Masala	Masalas	Vellore	11-08-2017	North	1254	0.12	401.28	Tamil Nadu
1	OD2	Sudha	Beverages	Health Drinks	Krishnagiri	11-08-2017	South	749	0.18	149.80	Tamil Nadu
2	OD3	Hussain	Food Grains	Atta & Flour	Perambalur	06-12-2017	West	2360	0.21	165.20	Tamil Nadu
3	OD4	Jackson	Fruits & Veggies	Fresh Vegetables	Dharmapuri	10-11-2016	South	896	0.25	89.60	Tamil Nadu
4	OD5	Ridhesh	Food Grains	Organic Staples	Ooty	10-11-2016	South	2355	0.26	918.45	Tamil Nadu

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

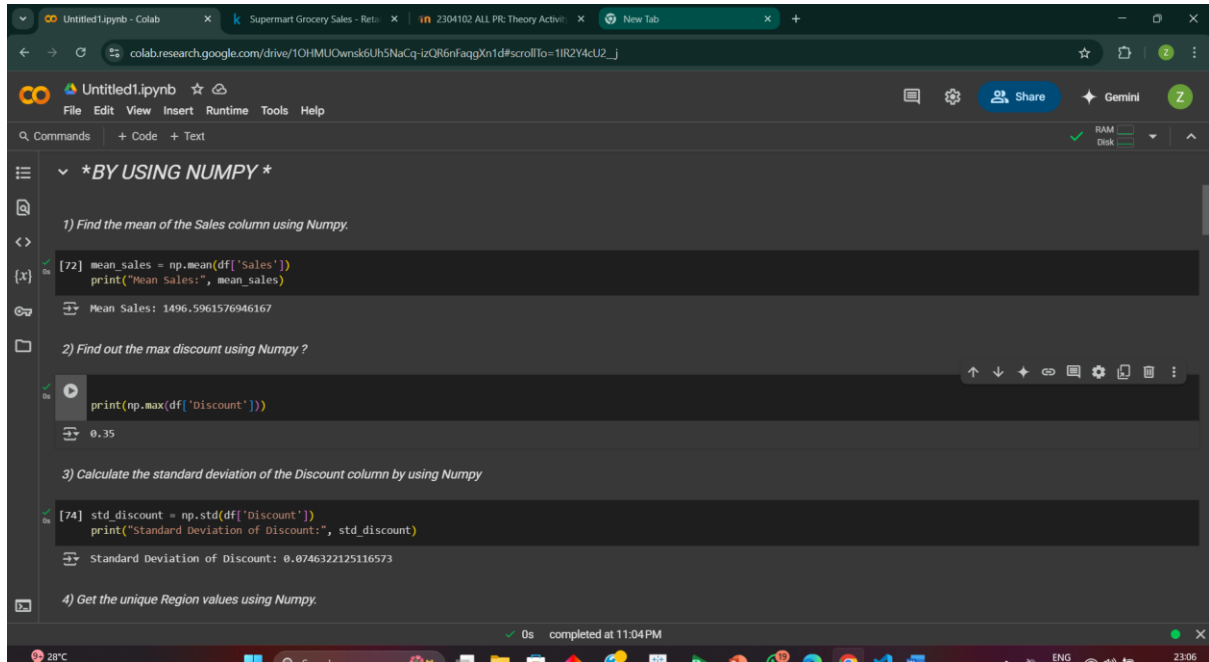
▼ ***BY USING NUMPY***

1) Find the mean of the Sales column using Numpy.

```
[72] mean_sales = np.mean(df['Sales'])
print("Mean Sales:", mean_sales)
```

0s completed at 11:04 PM

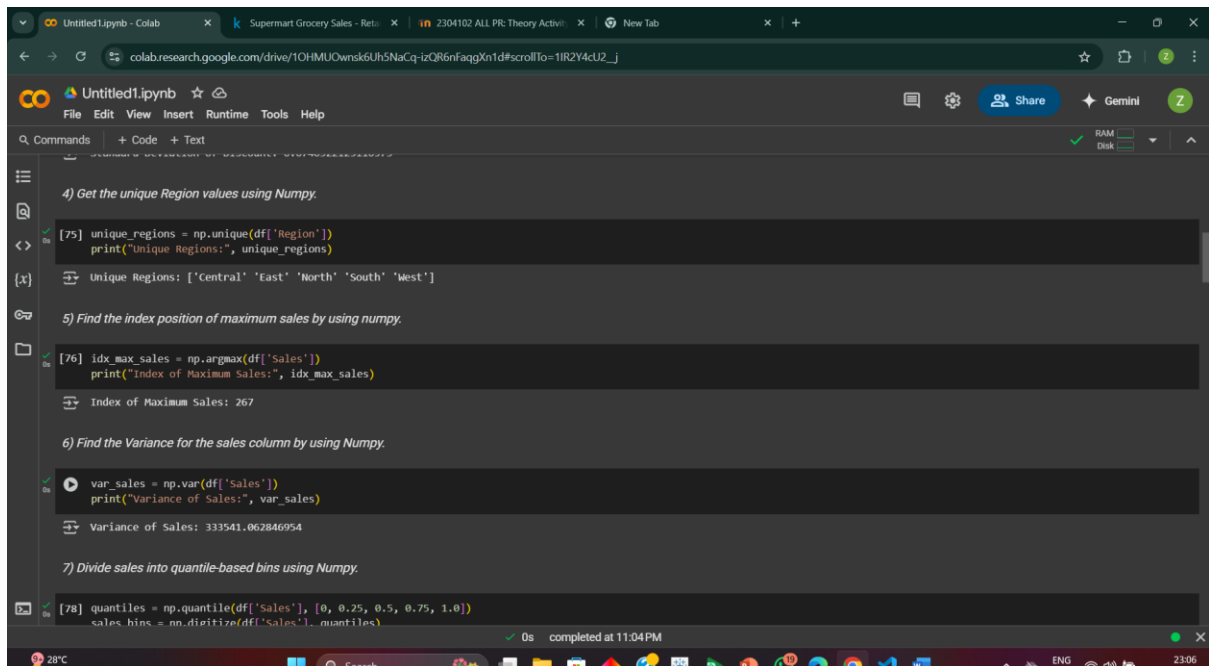
A) By using Numpy



The screenshot shows a Google Colab notebook titled 'Untitled1.ipynb'. The notebook contains four tasks, each with a corresponding code cell and output. The tasks are:

- 1) Find the mean of the Sales column using Numpy.
Code: `mean_sales = np.mean(df['Sales'])`
Output: `Mean Sales: 1496.5961576946167`
- 2) Find out the max discount using Numpy ?
Code: `print(np.max(df['Discount']))`
Output: `0.35`
- 3) Calculate the standard deviation of the Discount column by using Numpy
Code: `std_discount = np.std(df['Discount'])`
Output: `Standard Deviation of Discount: 0.0746322125116573`
- 4) Get the unique Region values using Numpy.

The notebook interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help), a toolbar with icons for file operations, and a status bar at the bottom showing '0s completed at 11:04 PM'.



The screenshot shows the continuation of the Google Colab notebook. It contains tasks 4 through 7, each with a corresponding code cell and output. The tasks are:

- 4) Get the unique Region values using Numpy.
Code: `unique_regions = np.unique(df['Region'])`
Output: `Unique Regions: ['Central' 'East' 'North' 'South' 'West']`
- 5) Find the index position of maximum sales by using numpy.
Code: `idx_max_sales = np.argmax(df['Sales'])`
Output: `Index of Maximum Sales: 267`
- 6) Find the Variance for the sales column by using Numpy.
Code: `var_sales = np.var(df['Sales'])`
Output: `Variance of Sales: 333541.062846954`
- 7) Divide sales into quantile-based bins using Numpy.
Code: `quantiles = np.quantile(df['Sales'], [0, 0.25, 0.5, 0.75, 1.0])`
Output: `sales.bins = np.digitize(df['Sales'], quantiles)`

The notebook interface is consistent with the previous screenshot, showing the same menu bar, toolbar, and status bar.

Untitled1.ipynb - Colab

colab.research.google.com/drive/1OHMUOwnsk6UHSNaCq-izQR6nfaggXn1d#scrollTo=11R2Y4cU2_j

File Edit View Insert Runtime Tools Help

Commands + Code + Text

7) Divide sales into quantile-based bins using Numpy.

```
[78]: quantiles = np.quantile(df['Sales'], [0, 0.25, 0.5, 0.75, 1.0])
      sales_bins = np.digitize(df['Sales'], quantiles)
      print("Sales Quantile Bins:", sales_bins)
```

Sales Quantile Bins: [2 1 4 ... 3 3 2]

8) Count the number of unique State values by using Numpy.

```
[9]: unique_states_count = np.unique(df['State']).size
      print("Number of Unique States:", unique_states_count)
```

Number of Unique States: 1

9) Find the 75th percentile of Profit by using Numpy.

```
[59]: percentile_75_profit = np.percentile(df['Profit'], 75)
      print("75th Percentile of Profit:", percentile_75_profit)
```

75th Percentile of Profit: 525.6275

10) Apply natural logarithm on the Sales column by using Numpy

0s completed at 11:04 PM

28°C Clear

Untitled1.ipynb - Colab

colab.research.google.com/drive/1OHMUOwnsk6UHSNaCq-izQR6nfaggXn1d#scrollTo=11R2Y4cU2_j

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Number of Unique States: 1

9) Find the 75th percentile of Profit by using Numpy.

```
[59]: percentile_75_profit = np.percentile(df['Profit'], 75)
      print("75th Percentile of Profit:", percentile_75_profit)
```

75th Percentile of Profit: 525.6275

10) Apply natural logarithm on the Sales column by using Numpy

```
[9]: log_sales = np.log(df['Sales'].replace(0, np.nan)).dropna()
      print("Logarithm of Sales (non-zero):", log_sales)
```

Logarithm of Sales (non-zero): 0 7.134094

1	6.618739
2	7.766417
3	6.797940
4	7.764296
...	
9989	6.851185
9990	7.085901
9991	7.356918
9992	7.413970
9993	6.941190

Name: Sales, Length: 9994, dtype: float64

BY USING THE PANDAS

0s completed at 11:04 PM

B) BY USING PANDAS

The screenshot shows a Google Colab notebook titled 'Untitled1.ipynb'. The first task is to find the total sales for each region. The code uses `df.groupby('Region')['Sales'].sum()` to calculate the total sales for each region. The output is a table with the following data:

Region	Sales
Central	3468156
East	4248368
North	1254
South	2440461
West	4798743

The second task is to find the top 5 cities with the highest total profit. The code uses `df.groupby('City')['Profit'].sum().sort_values(ascending=False).head(5)` to calculate the total profit for each city and then sort them in descending order. The output is a table with the following data:

City	Profit
Vellore	174073.01
Bodi	173655.13
Kanyakumari	172217.74
Perambalur	171132.19
Karur	169305.94

The screenshot shows the continuation of the Google Colab notebook. The third task is to identify the number of unique Order IDs per State. The code uses `df.groupby('State')['Order ID'].nunique()` to calculate the number of unique order IDs for each state. The output is a table with the following data:

State	Order ID
Tamil Nadu	9994

The fourth task is to find the average profit per sub-category. The code uses `df.groupby('Sub Category')['Profit'].mean()` to calculate the average profit for each sub-category. The output is a table with the following data:

Sub Category	Profit
Atta & Flour	362.212748
Biscuits	368.970050
Breads & Buns	380.009920
Cakes	372.562965
Chicken	356.465201
Chocolates	368.435551
Cookies	366.622500
Dals & Pulses	379.685977
Edible Oil & Ghee	373.821685
Eggs	381.714828
Fish	399.046098
Fresh Fruits	364.954878
Fresh Vegetables	370.828616
Health Drinks	372.002490

Untitled1.ipynb · Colab

colab.research.google.com/drive/1OHMUOwnsk6Uh5NaCq-izQR6nfaggXn1d#scrollTo=1IR2Y4CU2_

File Edit View Insert Runtime Tools Help

Commands + Code + Text

4) Find the average Profit per Sub-Category byt using pandas.

```
avg_profit_subcategory = df.groupby('Sub Category')['Profit'].mean()
print(avg_profit_subcategory)
```

Sub Category	Profit
Atta & Flour	362.212748
Biscuits	368.978850
Breads & Buns	388.009920
Cakes	372.562965
Chicken	356.465201
Chocolates	368.435551
Cookies	366.622500
Dals & Pulses	379.685977
Edible Oil & Ghee	373.821685
Eggs	381.714828
Fish	399.046098
Fresh Fruits	364.954878
Fresh Vegetables	370.828616
Health Drinks	372.002490
Masalas	365.008877
Mutton	384.233956
Noodles	391.284465
Organic Fruits	376.041178
Organic Staples	387.464758
Organic Vegetables	385.003948
Rice	384.643515
Soft Drinks	379.054288
Spices	358.618792

Name: Profit, dtype: float64

5) Find the Customer Name with maximum total Sales by usinn pandas.

0s completed at 11:04 PM

28°C Clear

Untitled1.ipynb · Colab

colab.research.google.com/drive/1OHMUOwnsk6Uh5NaCq-izQR6nfaggXn1d#scrollTo=1IR2Y4CU2_

File Edit View Insert Runtime Tools Help

Commands + Code + Text

5) Find the Customer Name with maximum total Sales by using pandas.

```
customer_max_sales = df.groupby('Customer Name')['Sales'].sum().idxmax()
print("Customer with maximum sales:", customer_max_sales)
```

Customer with maximum sales: Krithika

6) Find the Region and Category combination with highest total Profit by using pandas.

```
highest_profit_combo = df.groupby(['Region', 'Category'])['Profit'].sum().idxmax()
print("Highest profit combo:", highest_profit_combo)
```

Highest profit combo: ('West', 'Eggs, Meat & Fish')

7) Find the state having the maximum average Discount by using pandas.

```
state_max_avg_discount = df.groupby('State')['Discount'].mean().idxmax()
print("State with max avg discount:", state_max_avg_discount)
```

State with max avg discount: Tamil Nadu

8) Find the top 5 Sub-Categories that contribute the most to overall Profit by using pandas.

```
top5_profit_subcat = df.groupby('Sub Category')['Profit'].sum().sort_values(ascending=False).head(5)
print(top5_profit_subcat)
```

0s completed at 11:04 PM

Untitled1.ipynb - Colab

colab.research.google.com/drive/1OHMUOwnsk6Uh5NaCq-izQR6nfaggXn1d#scrollTo=1IR2Y4cU2_

File Edit View Insert Runtime Tools Help

Commands + Code + Text

8) Find the top 5 Sub-Categories that contribute the most to overall Profit by using pandas.

```
[96] top5_profit_subcat = df.groupby('Sub Category')['Profit'].sum().sort_values(ascending=False).head(5)
print(top5_profit_subcat)
```

Sub Category	Profit
Health Drinks	267469.79
Soft Drinks	258135.97
Noodles	193685.81
Breads & Buns	190764.98
Cookies	190643.70

Name: Profit, dtype: float64

9) Find the average number of orders per Customer by using pandas.

```
[97] orders_per_customer = df.groupby('Customer Name')['order ID'].nunique().mean()
print("Average orders per customer:", orders_per_customer)
```

Average orders per customer: 199.88

10) Find out the discount percentage range (min, max) for each Sub-Category by using pandas.

```
[98] discount_range_subcat = df.groupby('Sub Category')['Discount'].agg(['min', 'max'])
print(discount_range_subcat)
```

Sub Category	min	max
Atta & Flour	0.1	0.35

0s completed at 11:04 PM

Untitled1.ipynb - Colab

colab.research.google.com/drive/1OHMUOwnsk6Uh5NaCq-izQR6nfaggXn1d#scrollTo=1IR2Y4cU2_

File Edit View Insert Runtime Tools Help

Commands + Code + Text

Average orders per customer: 199.88

10) Find out the discount percentage range (min, max) for each Sub-Category by using pandas.

```
discount_range_subcat = df.groupby('Sub Category')['Discount'].agg(['min', 'max'])
print(discount_range_subcat)
```

Sub Category	min	max
Atta & Flour	0.1	0.35
Biscuits	0.1	0.35
Breads & Buns	0.1	0.35
Cakes	0.1	0.35
Chicken	0.1	0.35
Chocolates	0.1	0.35
Cookies	0.1	0.35
Dals & Pulses	0.1	0.35
Edible Oil & Ghee	0.1	0.35
Eggs	0.1	0.35
Fish	0.1	0.35
Fresh Fruits	0.1	0.35
Fresh Vegetables	0.1	0.35
Health Drinks	0.1	0.35
Masalas	0.1	0.35
Mutton	0.1	0.35
Noodles	0.1	0.35
Organic Fruits	0.1	0.35
Organic Staples	0.1	0.35
Organic Vegetables	0.1	0.35
Rice	0.1	0.35
Soft Drinks	0.1	0.35
Spices	0.1	0.35

0s completed at 11:04 PM