

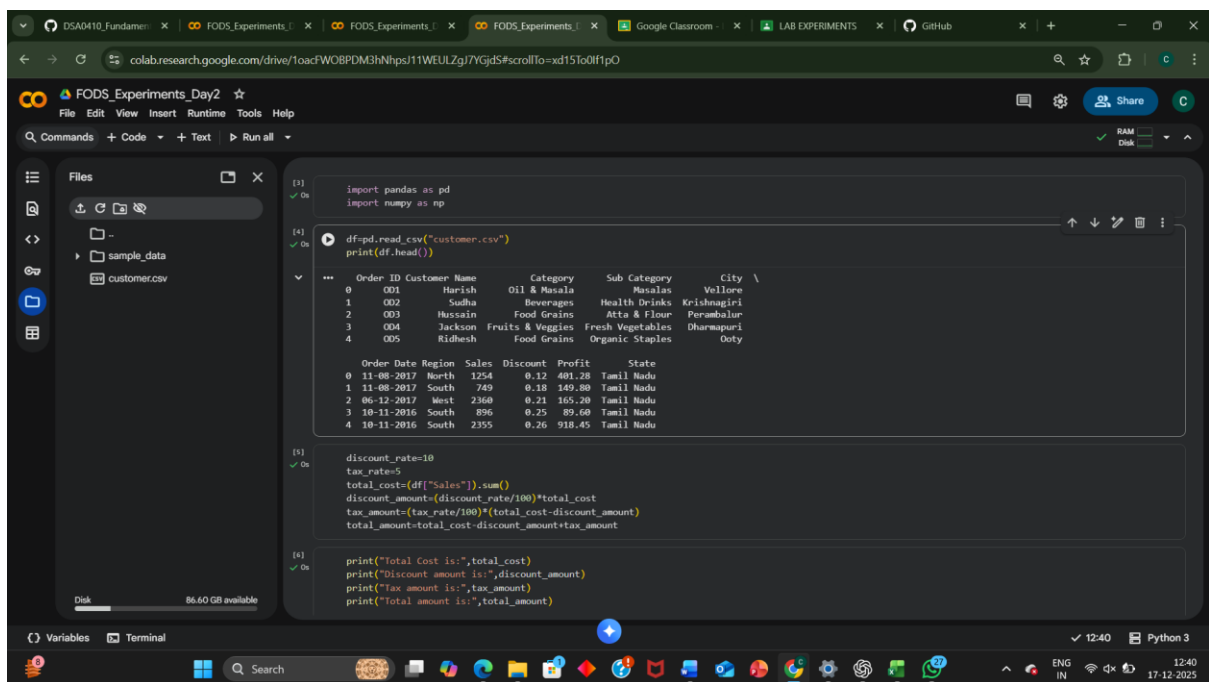
# DAY 2 LAB EXPERIMENTS

NAME : GUTTI CHAITANYA

Reg No : 192424185

Date : 17/12/2025

## EXP\_6 To Calculate the total Cost of Customer Purchase including Discount and Taxes



The screenshot shows a Google Colab notebook titled "FODS\_Experiments\_Day2". The code in the notebook performs the following steps:

- Imports pandas as pd and numpy as np.
- Reads a CSV file named "customer.csv" and prints the first 5 rows.
- Prints a summary of the data, showing columns: Order ID, Customer Name, Category, Sub Category, City, Order Date, Region, Sales, Discount, Profit, and State.
- Calculates the total cost, discount amount, tax amount, and total amount.
- Prints the final results.

```
[3]: import pandas as pd
import numpy as np

[4]: df=pd.read_csv("customer.csv")
print(df.head())

...
Order ID Customer Name Category Sub Category City \
0 001 Harish Oil & Masala Masalas Vellore
1 002 Sudha Beverages Health Drinks Krishnagiri
2 003 Hussain Food Grains Atta & Flour Perambalur
3 004 Jackson Fruits & Veggies Fresh Vegetables Dharmapuri
4 005 Ridhesh Food Grains Organic Staples Ooty

Order Date Region Sales Discount Profit State
0 11-08-2017 North 1254 0.12 401.28 Tamil Nadu
1 11-08-2017 South 749 0.18 149.80 Tamil Nadu
2 06-12-2017 West 2360 0.21 165.20 Tamil Nadu
3 10-11-2016 South 896 0.25 89.60 Tamil Nadu
4 10-11-2016 South 2355 0.26 918.45 Tamil Nadu

[5]: discount_rate=10
tax_rate=5
total_cost=(df["Sales"]).sum()
discount_amount=(discount_rate/100)*total_cost
tax_amount=(tax_rate/100)*(total_cost-discount_amount)
total_amount=total_cost-discount_amount+tax_amount

[6]: print("Total Cost is:",total_cost)
print("Discount amount is:",discount_amount)
print("Tax amount is:",tax_amount)
print("Total amount is:",total_amount)
```

## EXP\_7 Data Analyst for e commerce company and perform some operations like calculate total number of orders, average

The screenshot shows a Google Colab notebook titled "EXP\_7 Data Analyst for e commerce company and perform some operations like calculate total number of orders, average". The notebook is open to a cell containing the following code:

```
[66] import pandas as pd
[67] import numpy as np

[67] df=pd.read_csv("e commerce.csv",encoding="latin1")
[67] print(df.head())
```

The output of the code is a preview of the first 5 rows of the "e commerce.csv" file:

	InvoiceNo	StockCode	Description	Quantity
0	536365	851228A	WHITE HANGING HEART T-LIGHT HOLDER	6
1	536365	71053	WHITE METAL LANTERN	6
2	536365	844068	CREAM CUPID HEARTS COAT HANGER	8
3	536365	848029G	KNITTED UNION FLAG HOT WATER BOTTLE	6
4	536365	848029E	RED WOOLLY HOTTIE WHITE HEART.	6

Below the preview, the code continues with:

```
[68] df["InvoiceDate"] = pd.to_datetime(df["InvoiceDate"])

[68] total_orders=df.groupby("CustomerID").size()
[68] print("Total number of orders by each customer is :")
[68] print(total_orders)
```

The output of the code is a table showing the total number of orders by each customer:

CustomerID	size
12346.0	2
12347.0	182

The notebook interface shows the "Files" panel on the left with the "sample\_data" folder containing "customer.csv" and "e commerce.csv". The "Variables" panel at the bottom shows the current variables in the notebook.

The screenshot shows the same Google Colab notebook, now with the following code in a new cell:

```
[69] average_quantity=df.groupby("StockCode")["Quantity"].mean()
[69] print("Average order quantity for each product is :")
[69] print(average_quantity)
```

The output of the code is a table showing the average order quantity for each product:

StockCode	mean
10002	14.205479
10090	20.625900
10120	6.433333
10123C	-3.250000
10123G	-38.000000
...	...
gift_0001_20	2.000000
gift_0001_30	4.625000
gift_0001_40	1.000000
gift_0001_50	1.000000
...	...
...	...

The code continues with:

```
[70] earliest_date = df["InvoiceDate"].min()
```

The output of the code is the earliest date in the dataset:

```
12346.0    2009-01-01 00:00:00
12347.0    2009-01-01 00:00:00
12348.0    2009-01-01 00:00:00
12349.0    2009-01-01 00:00:00
12350.0    2009-01-01 00:00:00
...
```

```
1 11-08-2017 South 749 0.18 149.80 Tamil Nadu
2 06-12-2017 West 2360 0.21 165.20 Tamil Nadu
3 10-11-2016 South 896 0.25 89.60 Tamil Nadu
4 10-11-2016 South 2355 0.26 918.45 Tamil Nadu
```

```
[1] ✓ On
discount_rate=10
tax_rate=5
total_cost=(df["Sales"].sum())
discount_amount=(discount_rate/100)*total_cost
tax_amount=(tax_rate/100)*(total_cost-discount_amount)
total_amount=total_cost-discount_amount+tax_amount

[4] ✓ On
print("Total Cost is:",total_cost)
print("Discount amount is:",discount_amount)
print("Tax amount is:",tax_amount)
print("Total amount is:",total_amount)

Total Cost is: 14956982
Discount amount is: 1495698.2000000002
Tax amount is: 673064.1900000001
Total amount is: 14134347.99
```

**EXP\_8 A Company that sells products online and the Data Scientist been tasked with Analyzing the Sales data for the past month.**

```
EXP_8 A Company that sells products online and the Data Scientist been tasked with Analyzing the Sales data for the past month.
```

```
[114] ✓ On
import pandas as pd

[115] ✓ On
df=pd.read_csv("online_retail.csv",encoding="latin1")
print(df.head())
```

```
index InvoiceNo StockCode Description Quantity \
0 0 536365 85123A WHITE HANGING HEART T-LIGHT HOLDER 6
1 1 536365 71053 WHITE METAL LANTERN 6
2 2 536365 844068 CREAM CUPID HEARTS COAT HANGER 8
3 3 536365 848296 KNITTED UNION FLAG HOT WATER BOTTLE 6
4 4 536365 848296 RED WOOLLY HOTTIE WHITE HEART 6

InvoiceDate UnitPrice CustomerID Country
0 12/1/2010 8:26 2.55 17850.0 United Kingdom
1 12/1/2010 8:26 3.39 17850.0 United Kingdom
2 12/1/2010 8:26 2.75 17850.0 United Kingdom
3 12/1/2010 8:26 3.39 17850.0 United Kingdom
4 12/1/2010 8:26 3.39 17850.0 United Kingdom
```

```
[116] ✓ On
df["InvoiceDate"]=pd.to_datetime(df["InvoiceDate"])
latest_date=df["InvoiceDate"].max()
last_month_date=latest_date-pd.Timedelta(days=30)
last_month_data=df[df["InvoiceDate"]>=last_month_date]

[117] ✓ On
product_sales = last_month_data.groupby("Description")["Quantity"].sum()

[118] ✓ On
top_5_products = product_sales.sort_values(ascending=False).head(5)
print("Top 5 products sold in the past month:")
print(top_5_products)
```

The screenshot shows a Google Colab notebook titled 'FODS\_Experiments\_Day2'. The left sidebar displays a file explorer with a folder named 'sample\_data' containing files: 'customer.csv', 'e commerce.csv', and 'online\_retail.csv'. The main code area contains the following Python code:

```
2 2 536365 848296 LUCKY LUCKY HOTTIE WHITE HEART 6
3 3 536365 848296 KNITTED UNION FLAG HOT WATER BOTTLE 6
4 4 536365 848296 RED WOOLLY HOTTIE WHITE HEART. 6

InvoiceDate UnitPrice CustomerID Country
0 12/1/2010 8:26 2.55 17850.0 United Kingdom
1 12/1/2010 8:26 3.39 17850.0 United Kingdom
2 12/1/2010 8:26 2.75 17850.0 United Kingdom
3 12/1/2010 8:26 3.39 17850.0 United Kingdom
4 12/1/2010 8:26 3.39 17850.0 United Kingdom

[116] ✓ On
df["InvoiceDate"] = pd.to_datetime(df["InvoiceDate"])
latest_date = df["InvoiceDate"].max()
last_month_date = latest_date - pd.Timedelta(days=30)
last_month_data = df[df["InvoiceDate"] >= last_month_date]

[117] ✓ On
product_sales = last_month_data.groupby("Description")["Quantity"].sum()

[118] ✓ On
top_5_products = product_sales.sort_values(ascending=False).head(5)
print("Top 5 products sold in the past month:")
print(top_5_products)

*** Top 5 products sold in the past month:
Description
WORLD WAR 2 GLIDERS ASSD DESIGNS 8169
POPCORN HOLDER 6682
RABBIT NIGHT LIGHT 5719
JUMBO BAG RED RETROSPOT 4693
60 CAKE CASES VINTAGE CHRISTMAS 3460
Name: Quantity, dtype: int64
```

**EXP\_9 Real Estate Agency and have been given a Dataset containing Information about Properties for sale.**

The screenshot shows a Google Colab notebook titled 'FODS\_Experiments\_Day2'. The left sidebar displays a file explorer with a folder named 'sample\_data' containing files: 'customer.csv', 'e commerce.csv', 'online\_retail.csv', and 'properties.csv'. The main code area contains the following Python code:

```
[71] ✓ On
import pandas as pd

[72] ✓ On
property_data = pd.read_csv("properties.csv", low_memory=False)
print(property_data.head())
print(property_data.columns)

area_type availability location size \
0 Super built-up Area 15-Dec Electronic City Phase II 2 BHK
1 Plot Area Ready To Move Chikka Tirupathi 4 Bedroom
2 Built-up Area Ready To Move Uttarahalli 3 BHK
3 Super built-up Area Ready To Move Lingadheeranahalli 3 BHK
4 Super built-up Area Ready To Move Kothanur 2 BHK

society total_sqft bath balcony price
0 Coome 1056 2.0 1.0 39.07
1 Theanmp 2600 5.0 3.0 120.00
2 Nall 1440 2.0 3.0 62.00
3 Soieure 1521 3.0 1.0 95.00
4 Nall 1200 2.0 1.0 51.00
Index(['area_type', 'availability', 'location', 'size', 'society',
      'total_sqft', 'bath', 'balcony', 'price'],
      dtype='object')

[84] ✓ On
property_data["bedrooms"] = (
    property_data["size"]
    .str.extract(r'(\d+)')
    .astype(float)
)

[79] ✓ On
avg_price_per_location = property_data.groupby("location")["price"].mean()
print(avg_price_per_location)
```

FODS\_Experiments\_Day2

colab.research.google.com/drive/1oacFWOBPDM3hNhpsJ11WEULZgI7YGjds#scrollTo=g-OuiewjRjP

Files

- sample\_data
  - customer.csv
  - e commerce.csv
  - online\_retail.csv
  - properties.csv

```
[79] ✓ Gs
avg_price_per_location = property_data.groupby("location")["price"].mean()
print("Average listing price of properties in each location:")
print(avg_price_per_location)

Average listing price of properties in each location:
location
Anekal          16.000000
Banaswadi       35.000000
Basavangudi     50.000000
Bhoganhalli     22.890000
Devarabeesana Halli  124.833333
...
t.c.palya       160.000000
tc.palya        60.750000
vinayakanagar   200.000000
white field,kadugodi  275.000000
whitefield      32.730000
Name: price, Length: 1305, dtype: float64

[80] ✓ Gs
properties_more_than_4_bedrooms = property_data[property_data["bedrooms"] > 4].shape[0]
print("Number of properties with more than four bedrooms:")
print(properties_more_than_4_bedrooms)

...
Number of properties with more than four bedrooms:
846

[81] ✓ Gs
largest_area_property = property_data.loc[
    property_data["total_sqft"].idxmax()
]
print("\nProperty with the largest area:")
print(largest_area_property)

...
Property with the largest area:
```

Variables Terminal

Air: Moderate Friday

22:35 Python 3

17-12-2023

FODS\_Experiments\_Day2

colab.research.google.com/drive/1oacFWOBPDM3hNhpsJ11WEULZgI7YGjds#scrollTo=g-OuiewjRjP

Files

- sample\_data
  - customer.csv
  - e commerce.csv
  - online\_retail.csv
  - properties.csv

```
[80] ✓ Gs
properties_more_than_4_bedrooms = property_data[property_data["bedrooms"] > 4].shape[0]
print("Number of properties with more than four bedrooms:")
print(properties_more_than_4_bedrooms)

Number of properties with more than four bedrooms:
846

[81] ✓ Gs
largest_area_property = property_data.loc[
    property_data["total_sqft"].idxmax()
]
print("\nProperty with the largest area:")
print(largest_area_property)

...
Property with the largest area:
area_type      Built-up Area
availability    Ready To Move
location        BEML Layout
size            2 BHK
society         HMT
total_sqft      999
bath            2.0
balcony         3.0
price           45.0
bedrooms        2.0
Name: 1369, dtype: object
```

Variables Terminal

Air: Moderate Friday

22:35 Python 3

17-12-2023

## EXP\_10 Data visualization project and need to create basic plots using Matplotlib.

Browser tabs: FODS\_Experiments\_Day2 - Colab, Pandas analysis code, (26) WhatsApp, LAB EXPERIMENTS

Address bar: colab.research.google.com/drive/1oacFWOBPDM3hNhpsJ11WEULZgJ7YGjdS#scrollTo=MkgVL2Lor1uY

Set Google Chrome as your default browser and pin it to your taskbar. [Set as default](#)

FODS\_Experiments\_Day2

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

- ..
- sample\_data
- data\_visualization\_cars.csv

Disk 86.60 GB available

```
[4] ✓ 0s
import pandas as pd
import matplotlib.pyplot as plt

[5] ✓ 0s
data = pd.read_csv("data_visualization_cars.csv")

[6] ✓ 0s
plt.figure()
plt.plot(data["Month"], data["Sales"])
plt.xlabel("Month")
plt.ylabel("Sales")
plt.title("Monthly Sales Line Plot")
plt.show()
```

Monthly Sales Line Plot

Activate Windows  
Go to Settings to activate Windows.

11:01 AM Python 3

24°C Mostly clear

Search

Taskbar icons: File Explorer, Edge, Word, etc.

