6. Grocery Store Billing System (Discount + Tax)

Program:-

```
prices=[50,30,20]

quantities=[2,3,1]

discount_rate=10

tax_rate=5

subtotal=sum(p*q for p,q in zip(prices,quantities))

discount_amount=subtotal*(discount_rate/100)

subtotal_after_discount=subtotal-discount_amount

tax_amount=subtotal_after_discount*(tax_rate/100)

total_cost=subtotal_after_discount+tax_amount

print("subtotal:",subtotal)

print("total cost:",total_cost)
```

Output:-

```
subtotal: 210

total cost: 198.45
```

7. Order Data Analysis using Pandas

Program:-

```
import pandas as pd
order_data=pd.DataFrame({'customer_id':[1,2,1,3,2],
                    'order_date':pd.to_datetime(["2024-01-01","2024-01-05","2024-01-10","2024-01-03","2024-01-08"]),
                    'product_name':["apple","banana","apple","milk","banana"],
                    'order_quantity':[3,5,2,4,6]})
orders_per_customer=order_data.groupby('customer_id').size()
avg_qty_per_product=order_data.groupby('product_name')['order_quantity'].mean()
earliest_date=order_data['order_date'].min()
latest_date=order_data['order_date'].max()
print(orders_per_customer)
print(avg_qty_per_product)
print("earlieast:",earliest_date)
print("latest:",latest_date)
```

Output:-

```
customer_id
1    2
2    2
3    1
dtype: int64
product_name
apple     2.5
banana    5.5
milk      4.0
Name: order_quantity, dtype: float64
earlieast: 2024-01-01 00:00:00
latest: 2024-01-10 00:00:00
```

8. Top 5 Best-Selling Products (Pandas)

Program:-

```
import pandas as pd
sales_data=pd.DataFrame({
   'product':["A","B","C","A","B","D","E","A"],
   'quantity':[10,5,7,3,6,9,2,4]
})
top_5_products=(sales_data.groupby('product')['quantity']
             .sum()
             .sort_values(ascending=False)
             .head(5))
print(top_5_products)
```

Output:-

```
product
A    17
B    11
D     9
C     7
E     2
Name: quantity, dtype: int64
```

9. Real Estate Property Data Analysis

Program:-

```
import pandas as pd
property_data=pd.DataFrame({
   'property_id':[101,102,103,104],
   'location':["cityA","cityB","cityA","cityC"],
   'bedrooms':[3,5,4,6],
   'area':[1400,2000,1800,2500],
   'price':[300000,450000,380000,550000] })
avg_price_location=property_data.groupby('location')['price'].mean()
count_bedrooms=property_data[property_data['bedrooms']>4].shape[0]
largest_property=property_data.loc[property_data['area'].idxmax()]
print(avg_price_location)
print("properties with >4 bedrooms:",count_bedrooms)
print("largest property:\n",largest_property)
```

Output:-

```
location
cityA    340000.0
cityB    450000.0
cityC    550000.0
Name: price, dtype: float64
properties with >4 bedrooms: 2
largest property:
 property_id      104
location       cityC
bedrooms           6
area            2500
price         550000
Name: 3, dtype: object
```

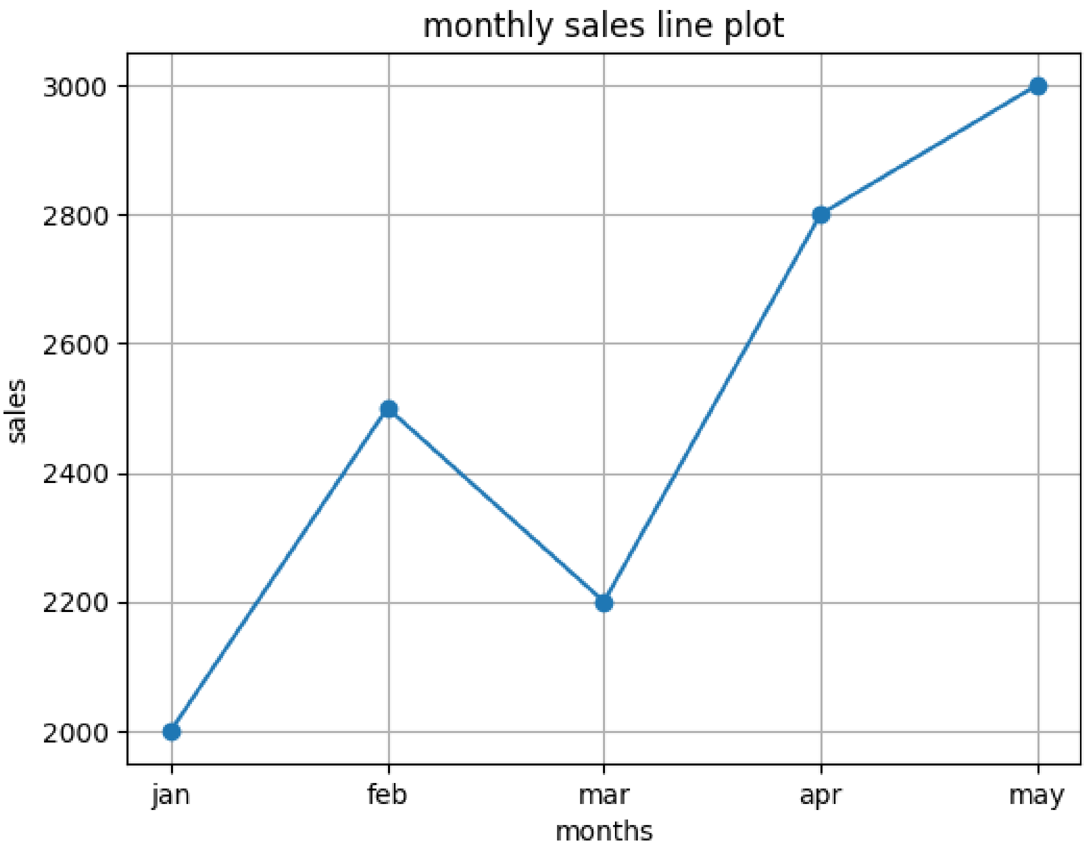
10. Data Visualization Using Matplotlib

(1) Line Plot of Monthly Sales

Program:-

```
import matplotlib.pyplot as plt

months=["jan","feb","mar","apr","may"]

sales=[2000,2500,2200,2800,3000]

plt.plot(months,sales,marker='o')

plt.xlabel("months")

plt.ylabel("sales")

plt.title("monthly sales line plot")

plt.grid(True)

plt.show()
```

Output:-

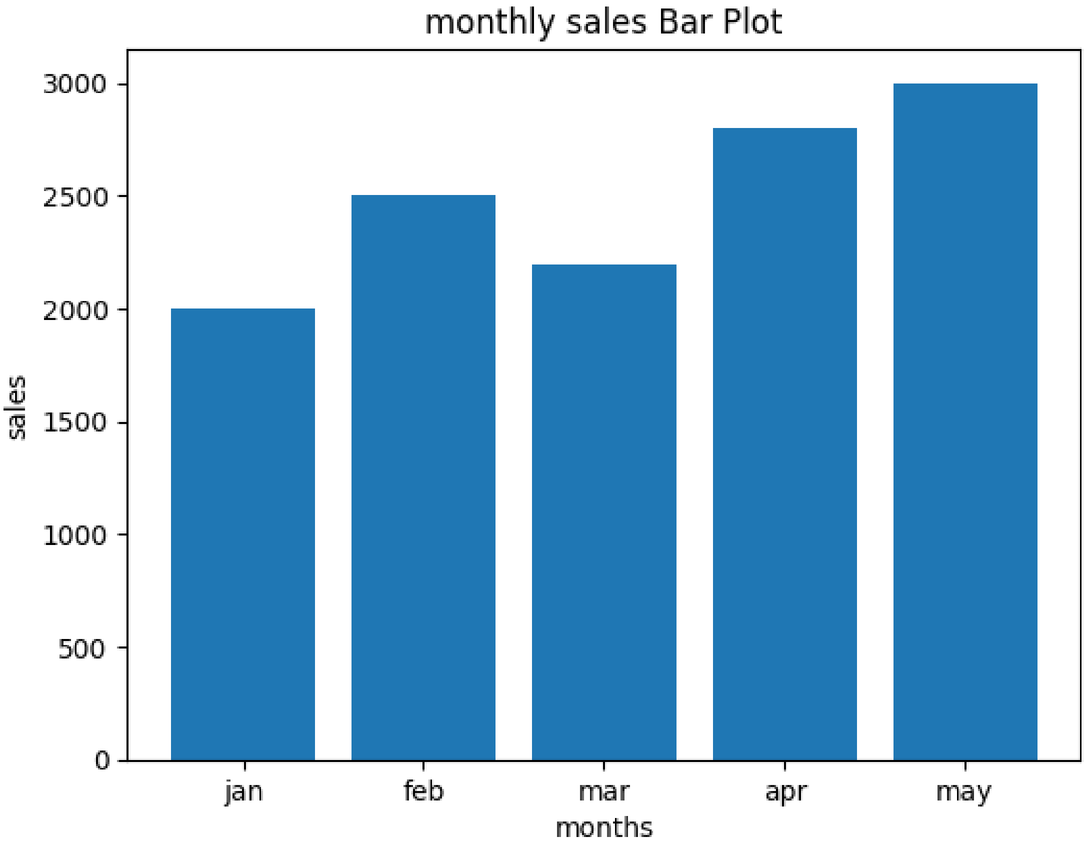


(2) Bar Plot of Monthly Sales

Program:-

```
import matplotlib.pyplot as plt

months=["jan","feb","mar","apr","may"]

sales=[2000,2500,2200,2800,3000]

plt.bar(months, sales)

plt.xlabel("months")

plt.ylabel("sales")

plt.title("monthly sales Bar Plot")

plt.show()
```

Output:-


monthly sales Bar Plot
sales
jan
feb
mar
apr
may
months