

Object Oriented Programming (Assignment-1)

Name: Md Fardin Islam Ali

ID: 0152420018

Section: BD

Ans: to the question no-1:

a) class Check Empty (Exception):

pass

class Check Burns (Exception):

pass

class Check Threshold (Exception):

pass

class Check Range (Exception):

pass

def process_Sensor_data (temperatures, threshold):

if len(temperatures) == 0:

raise Check Empty ("list can't be empty")

elif not isinstance(threshold, int):

raise Check Threshold ("Threshold must be an integer")

elif

else:

for i in temperatures:

if not isinstance(i, (int, float)):

raise Check Burns (Temperature must be integer or float)

elif i < -100 or i > 100:

raise Check Range (Temperature out of range)

filtered = []

for i in temperatures:

if i > threshold:

filtered.append(i)

else:

print ("Temperature is below threshold")

return filtered

1b)

f(1)

Start

Caught Value Error: Invalid value

Executing finally

f(2)

Start

Executing finally

f(3)

Start

Caught Type Error: Wrong Type

Executing finally

f(4)

Start

Executing finally

Ans: to the question no 2:

```
b) def even-value-extractor(n):  
    for i in range(0, n*2):  
        if i%2 == 0:  
            yield i
```

Ans: to the question no- 3:

a) df["Product-Category"].unique()

b) df["Sales"] = df["Price"] * df["Quantity"]
df.groupby("city")["Sales"].sum().idxmax()

c) df.groupby("Customer-Name")["Quantity"].sum().idxmax()

d) ~~df["Payment-Method"]~~

d) new = df[df["Product-Category"] == "Electronics"]
new["Payment-Method"].mode()

e) ~~df.groupby~~

df["Month"] = df["Purchase-Date"].dt.strftime("%B")

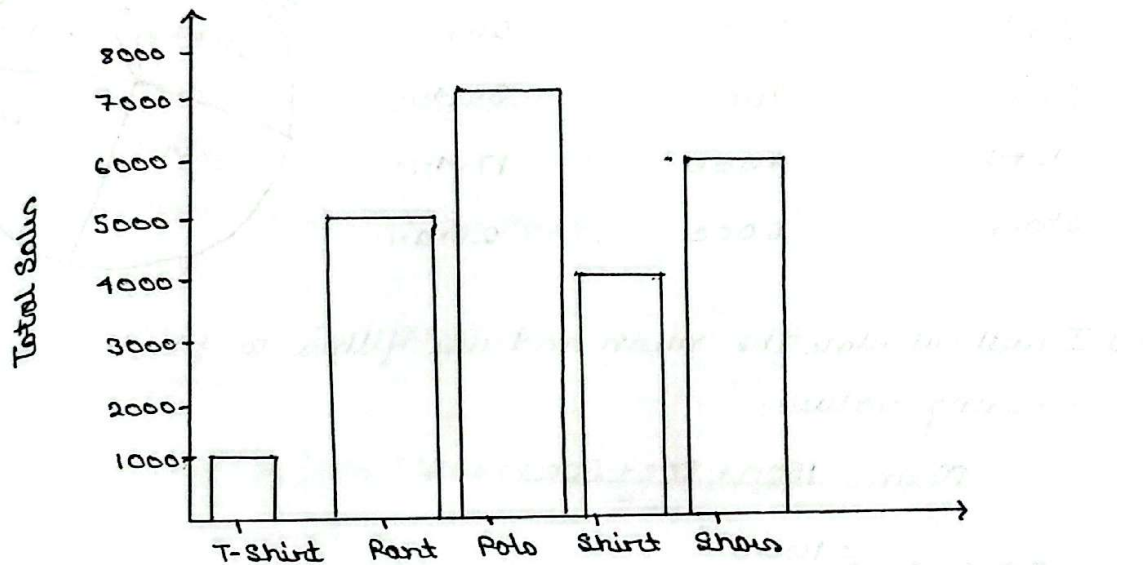
df.groupby("Month")["Sales"].sum().idxmax()

Ans: to the question no-4:-

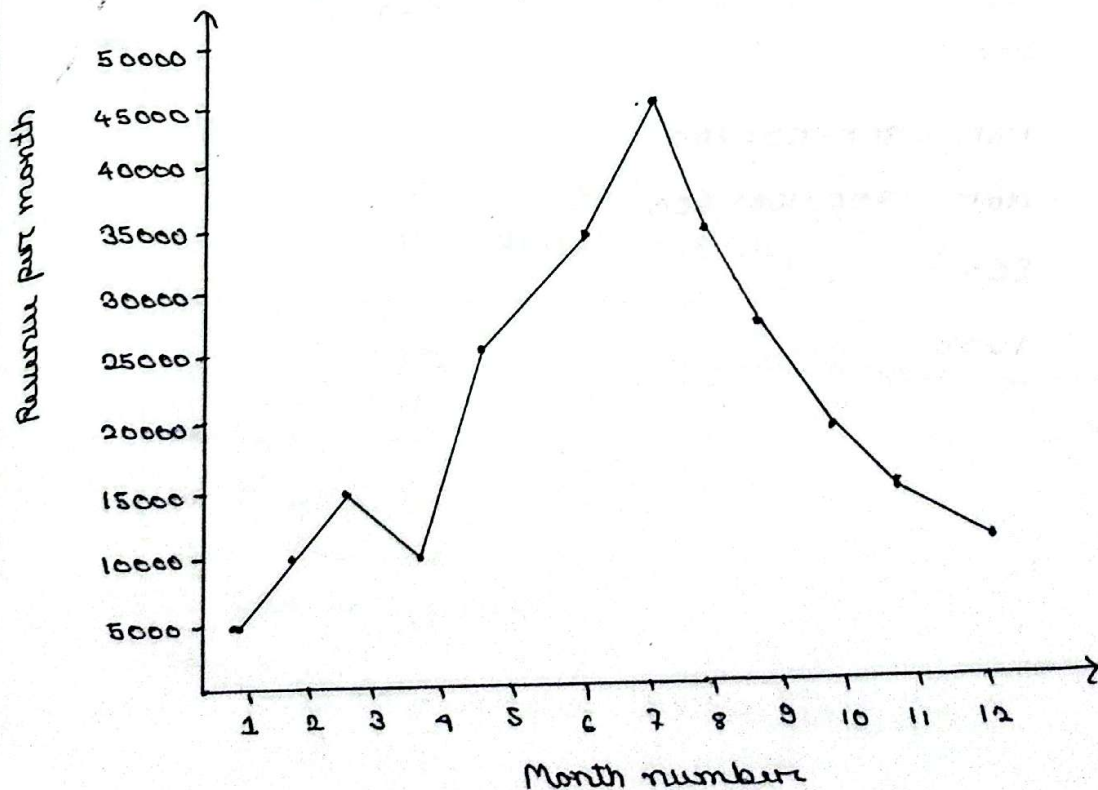
ai) Bar graph. Because it helps to visualize a single numerical column on different categorical & value.

x-data = ["T-Shirt", "Pant", "Polo", "Shirt", "Shoes"]

y-data = [1000, 5000, 7000, 4000, 6000]

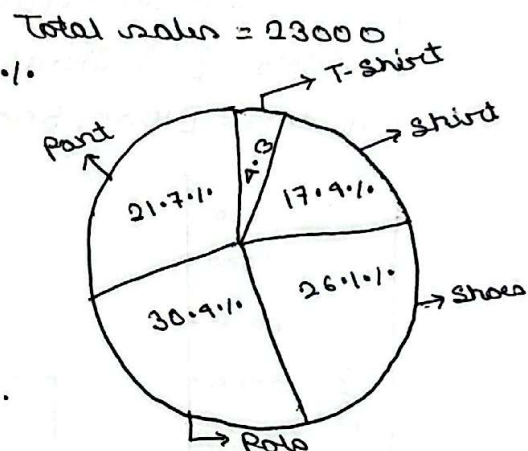


ii) Line graph. To visualize or observe the trend of something over time, line graph is a good choice.



iii) Pie Chart. It helps us to understand which product is contributing the most by taking most of the part of the circle.

Products	Sales	Sales in %
T-shirt	1000	4.3%
Pant	5000	21.7%
Polo	7000	30.4%
Shirt	4000	17.4%
Shoes	6000	26.2%



bi) I will calculate the mean and use fillna to fill the missing values.

$$\text{Mean} = \frac{1500 + 800 + 500 + 1200}{4}$$

$$= 1000$$

ii) Total Sales

15000

800

$$\text{NaN} = 800 - 100 = 700$$

$$\text{NaN} = 700 - 100 = 600$$

500

1200

$$\text{interpolate} = \frac{500 - 800}{2 + 1}$$

$$= -100$$