

Name → Garvit Joshi
Subject → INT 350
Teacher → Aman Kumar
Registration Number → 12106692

Roll No → 211

Page No → 01

PAGE No.	
DATE	

SET → 09

Ques-1

Suppose you're given the following Table Called 'orders' and 'order-info'. The table 'order' show revenue value for unique order along with its associated Channel ('online' or 'in-store') while the table 'order-info' shows the order ID along with its location.

Table: Orders

Table: Order-info

Using these Table, write a SQL Query to return the top 3 'online' orders and their associated location based on revenue generated.

Solution →
SELECT o.order_id, o.revenue, oi.location
FROM Orders o
JOIN order_info ON order_id = order_info.
order_id
WHERE channel = 'Online'
ORDER BY revenue DESC
LIMIT 3;

Ques-2

Consider the following table, Annual Sale, shown below:

year	total - sale
2015	23000
2016	25000
2017	34000
2018	32000
2019	33000

Use lag() & lead() function to compare annual sale amount across years.

Solution → CREATE TABLE annual-sale (
year INT,
total-sale INT
);

INSERT INTO annual-sale (year, total-sale)
VALUES

(2015, 23000),

(2016, 25000),

(2017, 34000),

(2018, 32000),

(2019, 33000);

SELECT

year,

total-sale,

Lag (total sale) OVER (ORDER BY year) AS
previous-year-sale,

LEAD (total-sale) OVER (ORDER BY year) AS
next-year-sale

FROM

annual-sale;

Ques-3 What is the difference between stored procedure and UDFs?

Name → Yashvi Joshi
Subject → INT 350
Teacher → Mr. Aman Kumar
Registration Number → 12106692

Roll No → 21
Page No → 03

PAGE No.	
DATE	

SET → 09

Stored procedure	User-defined function
Perform a set of operation, may or may not return a value.	Computes values and return a single value or a table.
May or may not be return a value. Can return multiple results set.	Must return a value. Can return a single value or a table.
Can be Called Stand-alone or within other SQL statements	Used as a expression within a set SQL statement, primarily (SELECT).
Can include DML (Data manipulation language) Statement.	Generally does not perform data modification.
Can include explicit transaction Control Statement ('COMMIT', 'ROLLBACK', etc.)	Can't include transaction control Statement.
Can have input, output, input-output parametre.	Accept input parametre only. Read only, Can't modify parametre.

Ques-4 Take any dataset of your choice & perform outlier analysis using boxplot. Write interpretation of graph.

Solution → Dataset = Scores of students

Name → Garvit Joshi

Subject → INT 350

Teacher → Mr. Aman Kumar

Registration Number → 12106682

Roll No → 21

Page No → 04

SET → 09

PAGE No.

DATE

Box plot :

```
import matplotlib.pyplot as plt
import seaborn as sns
scores = [70, 75, 80, 85, 90, 95, 100, 110]
sns.boxplot(x=scores)
plt.show()
```

Interpretation :

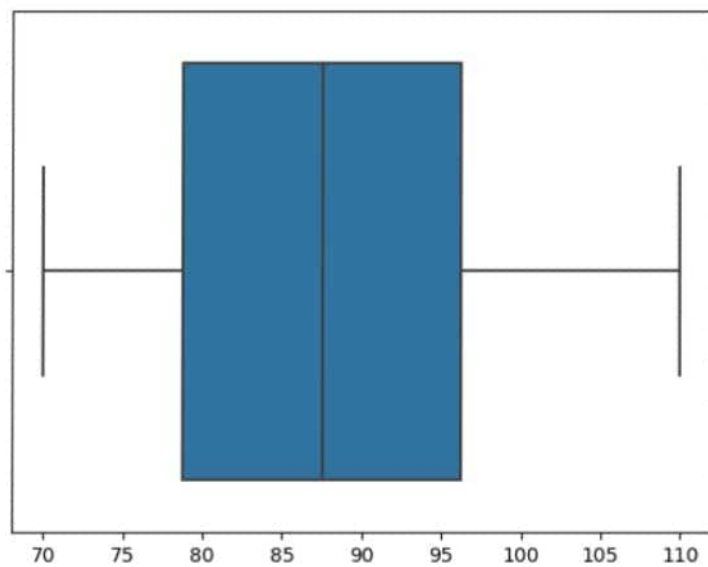
- Rectangular box represent interquartile range (IQR), which contain middle 50% of data.
- The 'whiskers' extending from box indicate the range of data within a certain multiple of IQR.
- The value '110' is significantly higher than rest of the score.

```
In [1]: import matplotlib.pyplot as plt
import seaborn as sns

# Define the dataset
scores = [70, 75, 80, 85, 90, 95, 100, 110]

# Create the box plot
sns.boxplot(x=scores)

# Show the plot
plt.show()
```



Page → 05

Ques-5 Using any dataset of your choice perform bivariate analysis & interpret each graph -

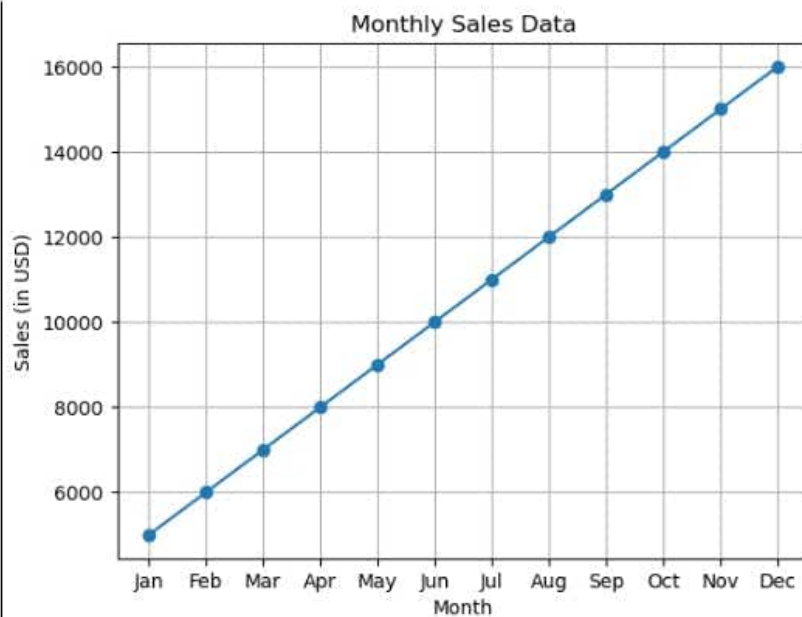
- a) Line Chart
- b) Bar graph
- c) Ben plot

Line Chart:

```
In [2]: import matplotlib.pyplot as plt

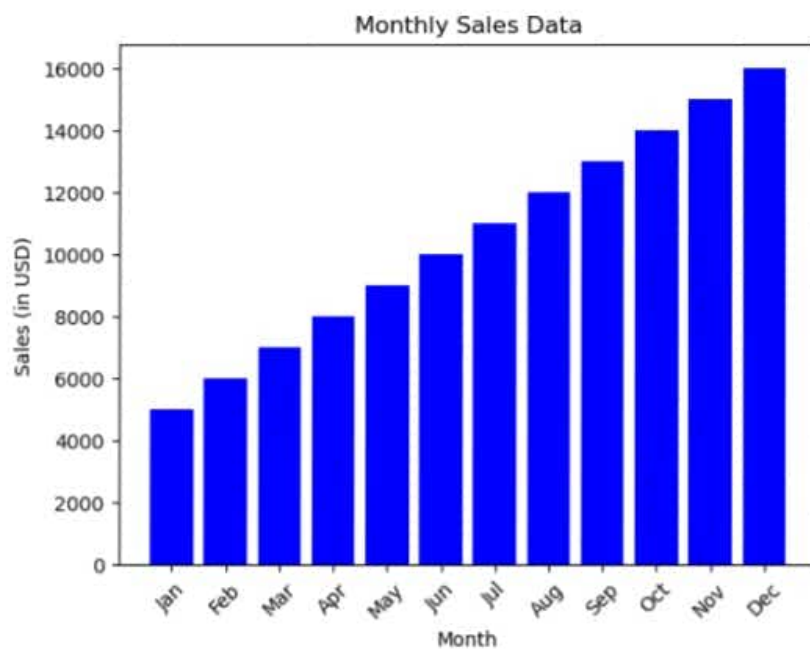
months = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"]
sales = [5000, 6000, 7000, 8000, 9000, 10000, 11000, 12000, 13000, 14000, 15000, 16000]

plt.plot(months, sales, marker='o')
plt.xlabel('Month')
plt.ylabel('Sales (in USD)')
plt.title('Monthly Sales Data')
plt.grid(True)
plt.show()
```



Bar Graph:

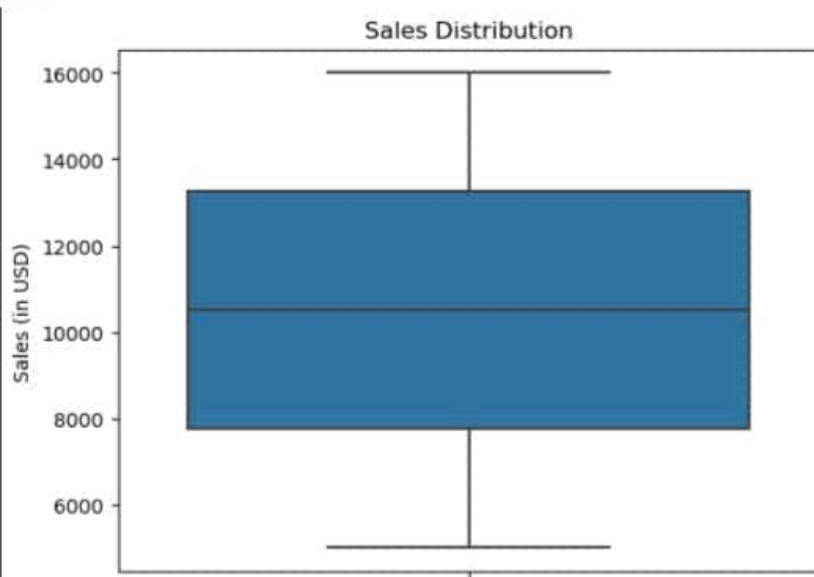
```
In [3]: plt.bar(months, sales, color='blue')
plt.xlabel('Month')
plt.ylabel('Sales (in USD)')
plt.title('Monthly Sales Data')
plt.xticks(rotation=45)
plt.show()
```



Box Plot:

```
In [4]: import seaborn as sns

sns.boxplot(y=sales)
plt.ylabel('Sales (in USD)')
plt.title('Sales Distribution')
plt.show()
```



Page → 06

Ques -

Briefly discuss about Stacked Bar Graph with example?

Ans ->

A Stacked Bar graph is a type of Bar chart that represent data in Categories & sub-categories, where the values for each sub-category are stacked up on top of one another. This allow for comparison of total values as well as Contribution of individual Component within each category.

Stacked bar graph:

```
In [5]: import matplotlib.pyplot as plt

regions = ['North', 'South', 'West']
electronics_sales = [5000, 7000, 6000]
apparel_sales = [3000, 4000, 3500]

plt.bar(regions, electronics_sales, label='Electronics', color='blue')
plt.bar(regions, apparel_sales, bottom=electronics_sales, label='Apparel', color='orange')

plt.xlabel('Region')
plt.ylabel('Sales (in USD)')
plt.title('Sales Data by Region and Product Category')
plt.legend()
plt.show()
```

