Data & Visual Analytics

**1. Aim: Implement ROLLUP & CUBE Operations on the following table**

|  |  |  |
| --- | --- | --- |
| **COLUMN NAME** | **DATA TYPE** | **SIZE** |
| **PET\_TYPE** | **VARCHAR2** | **15** |
| **STORE** | **VARCHAR2** | **15** |
| **NO** | **NUMBER** | **15** |

# Query:

## create table pet(pet\_type varchar(15),store varchar(15),no number(15));

### Output:

Table Created.

### Query:

insert into pet values('cat','miami',18); insert into pet values('dog', 'miami',12); insert into pet values('dog', 'tampa',14); insert into pet values('turtle', 'tampa',4); insert into pet values('dog', 'naples',5); insert into pet values('turtle', 'naples',1); **Output:**

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

### Query:

select \* from pet;

### Output:

|  |  |  |
| --- | --- | --- |
| pet\_type | store | no |
| cat | miami | 18 |
| dog | miami | 12 |
| dog | tampa | 14 |
| turtle | tampa | 4 |
| dog | naples | 5 |
| turtle | naples | 1 |

1. **Cube Query:**

select pet\_type, store, sum(no) from pet group by cube(pet\_type, store);

### Output:

|  |  |  |
| --- | --- | --- |
| **PET\_TYPE** | **STORE** | **SUM(NO)** |
| - | - | 54 |
| - | miami | 30 |
| - | tampa | 18 |
| - | naples | 6 |
| cat | - | 18 |
| cat | miami | 18 |
| dog | - | 31 |
| dog | miami | 12 |
| dog | tampa | 14 |
| dog | naples | 5 |
| turtle | - | 5 |
| turtle | tampa | 4 |
| turtle | naples | 1 |

13 rows selected.

### Rollup Query:

select pet\_type, store, sum(no) from pet group by rollup(pet\_type, store);

**Output:**

|  |  |  |
| --- | --- | --- |
| **PET\_TYPE** | **STORE** | **SUM(NO)** |
| cat | miami | 18 |
| cat | - | 18 |
| dog | miami | 12 |
| dog | tampa | 14 |
| dog | naples | 5 |
| dog | - | 31 |
| turtle | tampa | 4 |
| turtle | naples | 1 |
| turtle | - | 5 |
| - | - | 54 |

10 rows selected.

# Data for 2, 3, 4 & 5:

create table customer(cust\_id varchar2(20), cust\_name varchar2(20), cust\_city varchar2(20), cust\_state varchar2(20), cust\_country varchar2(20), cust\_age number(3), cust\_income number(9,3), primary key(cust\_id));

insert into customer values('C1','MANIDEEP','GUNTUR','AP','INDIA',23,35000); insert into customer values('C2','MADHU','ONGOLE','AP','INDIA',23,40000); insert into customer values('C3','ARUNBABU','GUNTUR','AP','INDIA',23,26000);

insert into customer values('C4','RAKESH','BENGALORE','KARNATAKA','INDIA',24,25000);

insert into customer values('C5','SHIRAJ','CHENNAI','TN','INDIA',25,38000);

create table item(item\_id varchar2(20), item\_name varchar2(20), item\_brand varchar2(20), item\_type varchar2(20), primary key (item\_id));

insert into item values('I1','HDTV','SAMSUNG','TV');

insert into item values('I2','LAPTOP','DELL','COMPUTER'); insert into item values('I3','MICROWAVE OVEN','LG','HAP');

create table employee(emp\_id varchar2(20),emp\_name varchar2(20),emp\_category varchar2(30), primary key(emp\_id));

insert into employee values('E1','JOHN','HOMEENTERTAIN'); insert into employee values('E2','SMITH','ELECTRONICS'); insert into employee values('E3','MILLER','ELECTRONICS');

insert into employee values('E4','SCOTT','HOUSEELECTRONICS'); insert into employee values('E5','KEVIN','AUTOMOBILE');

insert into employee values('E6','WARNE','HOMEENTERTAIN'); insert into employee values('E7','WATSON','ELECTRONICS'); insert into employee values('E8','HAYES','ELECTRONICS');

insert into employee values('E9','RODES','HOUSEELECTRONICS'); insert into employee values('E10','PETER','AUTOMOBILE');

create table branch(branch\_id varchar2(20), branch\_name varchar2(20), branch\_city varchar2(20), branch\_state varchar2(20), branch\_country varchar2(20), primary key(branch\_id));

insert into branch values('B1','CITYSQ','GUNTUR','AP','INDIA'); insert into branch values('B2','POTHIES','CHENNAI','TN','INDIA'); insert into branch values('B3','CMR','HYD','AP','INDIA');

insert into branch values('B4','MCM','BNG','KTK','INDIA');

insert into branch values('B5','GLAND','HYD','AP','INDIA');

create table purchases( trans\_id varchar2(20), cust\_id varchar2(20), emp\_id varchar2(20), date1 date, primary key(trans\_id), foreign Key (cust\_id) references customer(cust\_id),foreign Key (emp\_id) references employee(emp\_id) );

insert into purchases values('T100','C1','E1','03-JAN-06'); insert into purchases values('T101','C2','E2','01-FEB-06'); insert into purchases values('T102','C3','E3','05-MAR-07'); insert into purchases values('T103','C4','E4','08-APR-08'); insert into purchases values('T104','C5','E5','20-JUN-09'); insert into purchases values('T105','C1','E6','03-JAN-07');

insert into purchases values('T106','C2','E7','01-FEB-07'); insert into purchases values('T107','C3','E8','05-MAR-08'); insert into purchases values('T108','C4','E9','08-APR-09'); insert into purchases values('T109','C5','E10','20-JUN-06'); insert into purchases values('T110','C1','E6','03-JAN-08'); insert into purchases values('T111','C2','E7','01-FEB-08'); insert into purchases values('T112','C3','E8','05-MAR-09'); insert into purchases values('T113','C4','E9','08-APR-06'); insert into purchases values('T114','C5','E10','20-JUN-07'); insert into purchases values('T115','C1','E6','03-JAN-09'); insert into purchases values('T116','C2','E7','01-FEB-09'); insert into purchases values('T117','C3','E8','05-MAR-06'); insert into purchases values('T118','C4','E9','08-APR-07'); insert into purchases values('T119','C5','E10','20-JUN-08');

create table items\_sold(trans\_id varchar2(20), item\_id varchar2(20), qty number(10),foreign Key (trans\_id) references purchases(trans\_id),foreign Key (item\_id) references item(item\_id));

insert into items\_sold values('T100','I1',1); insert into items\_sold values('T100','I2',2); insert into items\_sold values('T100','I3',3); insert into items\_sold values('T101','I1',2); insert into items\_sold values('T101','I2',4); insert into items\_sold values('T101','I3',5); insert into items\_sold values('T102','I1',1); insert into items\_sold values('T102','I2',3); insert into items\_sold values('T102','I3',6); insert into items\_sold values('T103','I1',2); insert into items\_sold values('T103','I2',4); insert into items\_sold values('T103','I3',5); insert into items\_sold values('T104','I1',3); insert into items\_sold values('T104','I2',2); insert into items\_sold values('T104','I3',1); insert into items\_sold values('T105','I1',5); insert into items\_sold values('T105','I2',7); insert into items\_sold values('T105','I3',8); insert into items\_sold values('T106','I1',2); insert into items\_sold values('T106','I2',3); insert into items\_sold values('T106','I3',7); insert into items\_sold values('T107','I1',2); insert into items\_sold values('T107','I2',4); insert into items\_sold values('T107','I3',7); insert into items\_sold values('T108','I1',3); insert into items\_sold values('T108','I2',2); insert into items\_sold values('T108','I3',8); insert into items\_sold values('T109','I1',6);

insert into items\_sold values('T109','I2',5); insert into items\_sold values('T109','I3',2); insert into items\_sold values('T110','I1',4); insert into items\_sold values('T110','I2',7); insert into items\_sold values('T110','I3',8); insert into items\_sold values('T111','I1',4); insert into items\_sold values('T111','I2',5); insert into items\_sold values('T111','I3',8); insert into items\_sold values('T112','I1',9); insert into items\_sold values('T112','I2',3); insert into items\_sold values('T112','I3',6); insert into items\_sold values('T113','I1',7); insert into items\_sold values('T113','I2',2); insert into items\_sold values('T113','I3',5); insert into items\_sold values('T114','I1',3); insert into items\_sold values('T114','I2',2); insert into items\_sold values('T114','I3',1); insert into items\_sold values('T115','I1',3); insert into items\_sold values('T115','I2',2); insert into items\_sold values('T115','I3',1); insert into items\_sold values('T116','I1',3); insert into items\_sold values('T116','I2',2); insert into items\_sold values('T116','I3',9); insert into items\_sold values('T117','I1',3); insert into items\_sold values('T117','I2',2); insert into items\_sold values('T117','I3',8); insert into items\_sold values('T118','I1',3); insert into items\_sold values('T118','I2',2); insert into items\_sold values('T118','I3',6); insert into items\_sold values('T119','I1',4); insert into items\_sold values('T119','I2',2); insert into items\_sold values('T119','I3',1);

create table works\_at(empl\_id varchar2(20), branch\_id varchar2(20),foreign Key (empl\_id) references employee(emp\_id),foreign Key (branch\_id) references branch(branch\_id)); insert into works\_at values('E1','B1');

insert into works\_at values('E2','B2'); insert into works\_at values('E3','B3'); insert into works\_at values('E4','B4'); insert into works\_at values('E5','B5'); insert into works\_at values('E6','B1'); insert into works\_at values('E7','B2'); insert into works\_at values('E8','B3'); insert into works\_at values('E9','B4'); insert into works\_at values('E10','B5');

***Note: 2, 3, 4, 5 are not supported in SQL plus since it doesn’t support some keywords like JOIN etc.***

They should be run online using Live SQL. Steps are as follows:

1. On google, type Live SQL

## Open first link i.e, Oracle Live SQL

1. Click Start Coding

## Login if you have already created an account or signup as new user.

1. Paste the above data and execute it first.

## Run your SQL query.

### Implement CUBE SLICING- COME WITH 2-D VIEW DATA Implement Cube operation Slice

**Query:**

-- Slicing on customer and item dimensions

SELECT cust\_name, item\_name, SUM(qty) as total\_qty FROM customer

JOIN purchases ON customer.cust\_id = purchases.cust\_id JOIN items\_sold ON purchases.trans\_id = items\_sold.trans\_id JOIN item ON items\_sold.item\_id = item.item\_id

GROUP BY cust\_name, item\_name;

### Output:

**CUST\_NAME ITEM\_NAME**

ARUNBABU MANIDEEP MADHU MANIDEEP ARUNBABU SHIRAJ MADHU SHIRAJ SHIRAJ MANIDEEP ARUNBABU RAKESH MADHU RAKESH

RAKESH

LAPTOP

LAPTOP

**TOTAL\_QTY**

12

18

MICROWAVE OVEN 29

HDTV

13

MICROWAVE OVEN 27

LAPTOP HDTV

HDTV

11

11

16

MICROWAVE OVEN 5

MICROWAVE OVEN 20

HDTV LAPTOP LAPTOP

HDTV

15

10

14

15

MICROWAVE OVEN 24

15 rows selected.

### Aim: Implement Drill-down or Roll-down going from summary to more detailed data implement Cube operations.

**Query:**

-- Drill-down on customer to view purchases and items\_sold SELECT cust\_name, purchases.trans\_id, item\_name, qty FROM customer

JOIN purchases ON customer.cust\_id = purchases.cust\_id JOIN items\_sold ON purchases.trans\_id = items\_sold.trans\_id JOIN item ON items\_sold.item\_id = item.item\_id

### Output:

|  |  |  |  |
| --- | --- | --- | --- |
| **CUST\_NAME TRANS\_ID ITEM\_NAME QTY** | | | |
| MANIDEEP | T100 | HDTV | 1 |
| MADHU | T101 | HDTV | 2 |
| ARUNBABU | T102 | HDTV | 1 |
| RAKESH | T103 | HDTV | 2 |
| SHIRAJ | T104 | HDTV | 3 |
| MANIDEEP | T105 | HDTV | 5 |
| MADHU | T106 | HDTV | 2 |
| ARUNBABU | T107 | HDTV | 2 |
| RAKESH | T108 | HDTV | 3 |
| SHIRAJ | T109 | HDTV | 6 |
| MANIDEEP | T110 | HDTV | 4 |
| MADHU | T111 | HDTV | 4 |
| ARUNBABU | T112 | HDTV | 9 |
| RAKESH | T113 | HDTV | 7 |
| SHIRAJ | T114 | HDTV | 3 |
| MANIDEEP | T115 | HDTV | 3 |
| MADHU | T116 | HDTV | 3 |
| ARUNBABU | T117 | HDTV | 3 |
| RAKESH | T118 | HDTV | 3 |
| SHIRAJ | T119 | HDTV | 4 |
| MANIDEEP | T100 | LAPTOP | 2 |
| MADHU | T101 | LAPTOP | 4 |
| ARUNBABU | T102 | LAPTOP | 3 |
| RAKESH | T103 | LAPTOP | 4 |
| SHIRAJ | T104 | LAPTOP | 2 |
| MANIDEEP | T105 | LAPTOP | 7 |
| MADHU | T106 | LAPTOP | 3 |
| ARUNBABU | T107 | LAPTOP | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| RAKESH | T108 | LAPTOP | 2 |
| SHIRAJ | T109 | LAPTOP | 5 |
| MANIDEEP | T110 | LAPTOP | 7 |
| MADHU | T111 | LAPTOP | 5 |
| ARUNBABU | T112 | LAPTOP | 3 |
| RAKESH | T113 | LAPTOP | 2 |
| SHIRAJ | T114 | LAPTOP | 2 |
| MANIDEEP | T115 | LAPTOP | 2 |
| MADHU | T116 | LAPTOP | 2 |
| ARUNBABU | T117 | LAPTOP | 2 |
| RAKESH | T118 | LAPTOP | 2 |
| SHIRAJ | T119 | LAPTOP | 2 |
| MANIDEEP | T100 | MICROWAVE OVEN | 3 |
| MADHU | T101 | MICROWAVE OVEN | 5 |
| ARUNBABU | T102 | MICROWAVE OVEN | 6 |
| RAKESH | T103 | MICROWAVE OVEN | 5 |
| SHIRAJ | T104 | MICROWAVE OVEN | 1 |
| MANIDEEP | T105 | MICROWAVE OVEN | 8 |
| MADHU | T106 | MICROWAVE OVEN | 7 |
| ARUNBABU | T107 | MICROWAVE OVEN | 7 |
| RAKESH | T108 | MICROWAVE OVEN | 8 |
| SHIRAJ | T109 | MICROWAVE OVEN | 2 |

Rows 1 - 50. More rows exist.

### Aim: Implement Rollup - summarize data along a dimension hierarchy Implement Cube operations

**Query:**

-- Roll-up to summarize data

SELECT cust\_city, item\_type, sum(qty) as total\_quantity FROM customer c

JOIN purchases p ON c.cust\_id = p.cust\_id JOIN items\_sold isd ON p.trans\_id = isd.trans\_id JOIN item i ON isd.item\_id = i.item\_id

GROUP BY ROLLUP(cust\_city, item\_type);

### Output:

**CUST\_CITY** GUNTUR GUNTUR

GUNTUR

**ITEM\_TYPE TOTAL\_QUANTITY**

TV

HAP

28

47

COMPUTER 30

|  |  |  |
| --- | --- | --- |
| GUNTUR | - | 105 |
| ONGOLE | TV | 11 |
| ONGOLE | HAP | 29 |
| ONGOLE | COMPUTER | 14 |
| ONGOLE | - | 54 |
| CHENNAI | TV | 16 |
| CHENNAI | HAP | 5 |
| CHENNAI | COMPUTER | 11 |
| CHENNAI | - | 32 |
| BENGALORE | TV | 15 |
| BENGALORE | HAP | 24 |
| BENGALORE COMPUTER 10  BENGALORE - 49  - - 240 | | |

17 rows selected.

### Aim: To Implement Dicing – project 2-D view of data Implement Cube operations.

**Query:**

-- Dicing on customer and employee dimensions SELECT cust\_name, emp\_name, sum(qty) as total\_qty FROM customer

JOIN purchases ON customer.cust\_id = purchases.cust\_id JOIN items\_sold ON purchases.trans\_id = items\_sold.trans\_id JOIN employee ON purchases.emp\_id = employee.emp\_id GROUP BY cust\_name, emp\_name;

### Output:

|  |  |  |
| --- | --- | --- |
| **CUST\_NAME EMP\_NAME TOTAL\_QTY** | | |
| ARUNBABU | HAYES | 44 |
| SHIRAJ | PETER | 26 |
| MADHU | SMITH | 11 |
| ARUNBABU | MILLER | 10 |
| RAKESH | SCOTT | 11 |
| MANIDEEP | WARNE | 45 |
| RAKESH | RODES | 38 |
| MANIDEEP | JOHN | 6 |
| SHIRAJ | KEVIN | 6 |
| MADHU | WATSON | 43 |

10 rows selected.

In [6]:

*#6. Bar plot*

**import** matplotlib.pyplot **as** plt

x**=**list(map(str,input("enter the x-axis values/labels: ")**.**split())) y**=**list(map(int,input("enter the y-axis values: ")**.**split()))

xlabel**=**input("enter x-axis label: ") ylabel**=**input("enter y-axis label: ")

title**=**input("enter title for the barplot: ") plt**.**bar(x,y,color**=**'red',width**=**0.4)

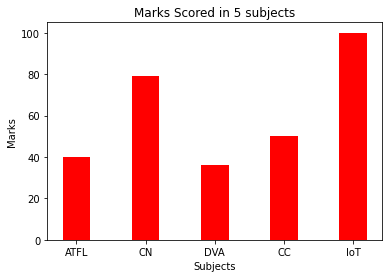
plt**.**xlabel(xlabel) plt**.**ylabel(ylabel) plt**.**title(title) plt**.**show()

In [4]:

enter the x-axis values/labels: ATFL CN DVA CC IoT enter the y-axis values: 40 79 36 50 100

enter x-axis label: Subjects enter y-axis label: Marks

enter title for the barplot: Marks Scored in 5 subjects



enter the y-axis group1 values: 34 56 12 89 67

*#7. Grouped and Stacked Bar Plot* **import** matplotlib.pyplot **as** plt **import** numpy **as** np

y1 **=** list(map(int,input("enter the y-axis group1 values: ")**.**split())) y2 **=** list(map(int,input("enter the y-axis group2 values: ")**.**split())) y3 **=** list(map(int,input("enter the y-axis group3 values: ")**.**split())) x**=**np**.**arange(len(y1))

width **=** 0.2

plt**.**bar(x**-**0.2, y1, width, color**=**'red') plt**.**bar(x, y2, width, color**=**'green')

plt**.**bar(x**+**0.2, y3, width, color**=**'blue')

xlabel**=**input("enter x-axis label: ") ylabel**=**input("enter y-axis label: ")

labels**=**list(map(str,input("enter the x-axis values/labels: ")**.**split())) plt**.**xticks(x, labels)

plt**.**xlabel(xlabel) plt**.**ylabel(ylabel)

plt**.**title("Grouped Bars")

plt**.**legend(["Group 1", "Group 2", "Group 3"]) plt**.**show()

y1 **=** np**.**array(y1) y2 **=** np**.**array(y2) y3 **=** np**.**array(y3)

plt**.**bar(x, y1, color**=**'r')

plt**.**bar(x, y2, bottom**=**y1, color**=**'b')

plt**.**bar(x, y3, bottom**=**y1 **+** y2, color**=**'y') plt**.**xticks(x, labels)

plt**.**xlabel(xlabel) plt**.**ylabel(ylabel)

plt**.**title("Stacked Bars")

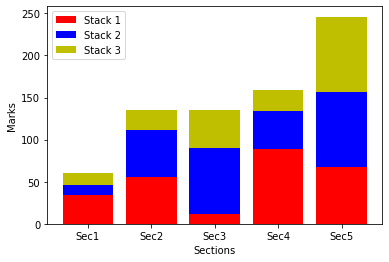
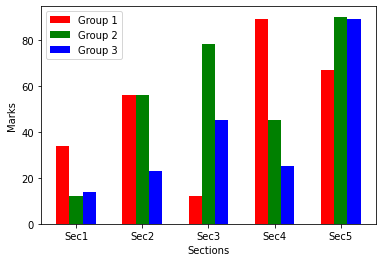
plt**.**legend(["Stack 1", "Stack 2", "Stack 3"]) plt**.**show()

enter the y-axis group2 values: 12 56 78 45 90

enter the y-axis group3 values: 14 23 45 25 89 enter x-axis label: Sections

enter y-axis label: Marks

enter title for the barplot: Marks in different sections enter the x-axis values/labels: Sec1 Sec2 Sec3 Sec4 Sec5



In [1]:

*#8. Dot Plot*

**import** matplotlib.pyplot **as** plt

**import** numpy **as** np

x **=** list(map(int,input("enter the x-axis values: ")**.**split()))

y **=** list(map(str,input("enter the y-axis values/labels: ")**.**split())) r**=**np**.**arange(len(x))

xlabel **=** input("Enter x-axis label: ")

ylabel **=** input("Enter y-axis label: ")

title **=** input("Enter title for the bar plot: ") plt**.**xlabel(xlabel)

plt**.**ylabel(ylabel)

plt**.**title(title) plt**.**plot(x,y,'o') plt**.**show()

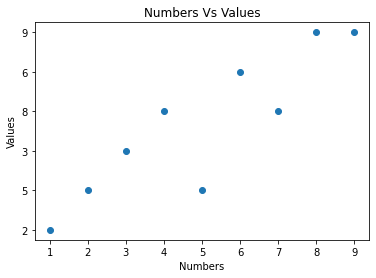
In [2]:

enter the x-axis values: 1 2 3 4 5 6 7 8 9

enter the y-axis values/labels: 2 5 3 8 5 6 8 9 9 Enter x-axis label: Numbers

Enter y-axis label: Values

Enter title for the bar plot: Numbers Vs Values



In [5]:

Enter city names separated by spaces: A B C D

*#9. Heat Map*

**import** numpy **as** np

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

cities **=** input("Enter city names separated by spaces: ")**.**split() months **=** input("Enter month names separated by spaces: ")**.**split()

temperature\_data **=** []

**for** city **in** cities:

temperatures **=** list(map(float, input(f"Enter temperatures for {city} separated by spaces: ")**.**split())) temperature\_data**.**append(temperatures)

sns**.**set()

plt**.**figure(figsize**=**(10, 6))

sns**.**heatmap(temperature\_data, annot**=True**, xticklabels**=**months, yticklabels**=**cities) plt**.**xlabel('Months')

plt**.**ylabel('Cities')

plt**.**title('Temperature Recorded Across Various Cities') plt**.**show()

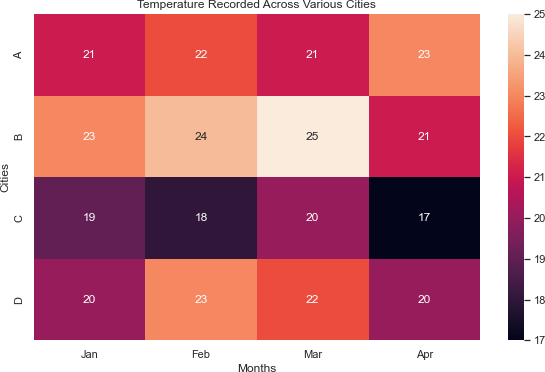
Enter month names separated by spaces: Jan Feb Mar Apr

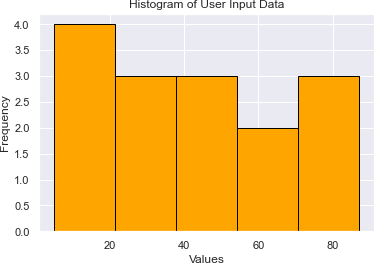
Enter temperatures for A separated by spaces: 21 22 21 23

Enter temperatures for B separated by spaces: 23 24 25 21

Enter temperatures for C separated by spaces: 19 18 20 17

Enter temperatures for D separated by spaces: 20 23 22 20



Enter a list of numerical data separated by spaces: 22 87 5 43 56 73 55 54 11 20 51 5 79 31 27

*#10. Histogram*

**import** matplotlib.pyplot **as** plt

data **=** list(map(float, input("Enter a list of numerical data separated by spaces: ")**.**split())) plt**.**hist(data, bins**=**'auto', color**=**'orange', edgecolor**=**'black')

plt**.**xlabel('Values')

plt**.**ylabel('Frequency')

plt**.**title('Histogram of User Input Data') plt**.**show()

In [1]:

*#11. Density Plot*

**import** seaborn **as** sns

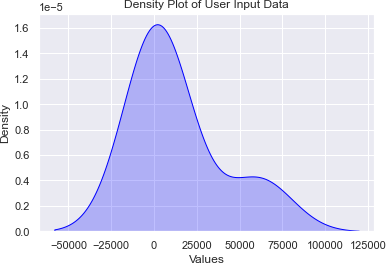
**import** matplotlib.pyplot **as** plt

data **=** list(map(float, input("Enter a list of numerical data separated by spaces: ")**.**split())) sns**.**set()

sns**.**kdeplot(data, color**=**'blue', fill**=True**) plt**.**xlabel('Values')

plt**.**ylabel('Density')

plt**.**title('Density Plot of User Input Data') plt**.**show()

Enter a list of numerical data separated by spaces: 1 45 276 6736 61288

In [ ]:

In [3]:

*#12. ECDF Plot*

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

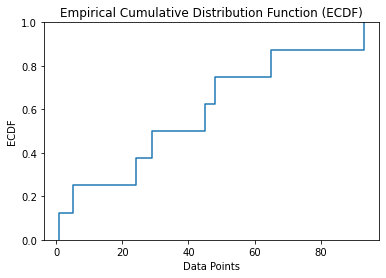
data**=**list(map(float,input("Enter list of values: ")**.**split())) sns**.**ecdfplot(data)

plt**.**xlabel('Data Points') plt**.**ylabel('ECDF')

plt**.**title('Empirical Cumulative Distribution Function (ECDF)')

plt**.**show()

Enter list of values: 1 93 45 65 5 24 48 29



In [16]:

*#13. Q-Q Plot*

**import** numpy **as** np

**import** statsmodels.api **as** sm

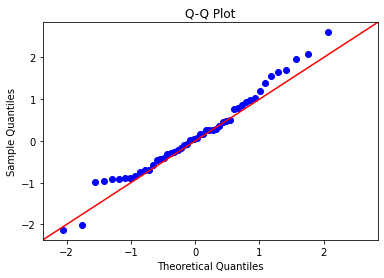
**import** matplotlib.pyplot **as** plt

n **=** int(input("Enter number of items: "))

data **=** np**.**random**.**normal(loc**=**0, scale**=**1, size**=**n) sm**.**qqplot(data, line **=**'45')

plt**.**title("Q-Q Plot") plt**.**show()

Enter number of items: 50



In [ ]:

In [ ]:

In [1]:

C:\Users\tejan\AppData\Roaming\Python\Python39\site-packages\matplotlib\projections\ init .py:63: UserWarning: Unable to import Axes3D. This may be due to multiple versions of Ma tplotlib being installed (e.g. as a system package and as a pip package). As a result, the 3D projection is not available.

warnings.warn("Unable to import Axes3D. This may be due to multiple versions of "

*#14. Aim: To draw a Box plot for given input data* **import** matplotlib.pyplot **as** plt n**=**int(input("Enter the number of temperatures: ")) temp\_data**=**[]

**for** i **in** range(n):

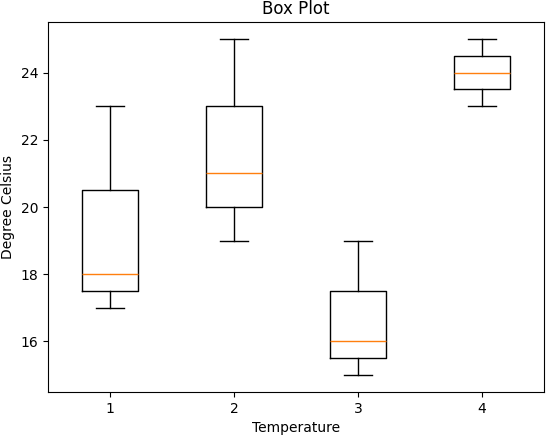
temp\_data**.**append(list(map(float,input(f"Enter temperature(s) of {i**+**1}: ")**.**split()))) plt**.**boxplot(temp\_data)

plt**.**xlabel('Temperature')

plt**.**ylabel('Degree Celsius') plt**.**title('Box Plot') plt**.**show()

Enter the number of temperatures: 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Enter | temperature(s) | of | 1: | 18 | 23 | 17 |
| Enter | temperature(s) | of | 2: | 25 | 21 | 19 |
| Enter | temperature(s) | of | 3: | 15 | 16 | 19 |
| Enter | temperature(s) | of | 4: | 23 | 24 | 25 |



In [5]:

pip install joypy

Collecting joypy

Using cached joypy-0.2.6-py2.py3-none-any.whl (8.6 kB)

Requirement already satisfied: numpy>=1.16.5 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from joypy) (1.26.2) Requirement already satisfied: scipy>=0.11.0 in d:\anaconda\lib\site-packages (from joypy) (1.9.1)

Requirement already satisfied: pandas>=0.20.0 in d:\anaconda\lib\site-packages (from joypy) (1.4.4)

Requirement already satisfied: matplotlib in c:\users\tejan\appdata\roaming\python\python39\site-packages (from joypy) (3.8.2)

Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from pandas>=0.20.0->joypy) (2.8.2) Requirement already satisfied: pytz>=2020.1 in d:\anaconda\lib\site-packages (from pandas>=0.20.0->joypy) (2022.1)

Collecting numpy>=1.16.5 (from joypy)

Downloading numpy-1.24.4-cp39-cp39-win\_amd64.whl.metadata (5.6 kB)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (1.2.0) Requirement already satisfied: cycler>=0.10 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (0.12.1) Requirement already satisfied: fonttools>=4.22.0 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (4.46.0) Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (1.4.5) Requirement already satisfied: packaging>=20.0 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (23.2) Requirement already satisfied: pillow>=8 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (10.1.0) Requirement already satisfied: pyparsing>=2.3.1 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (3.1.1)

Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from matplotlib->joypy) (6.1.1)

Requirement already satisfied: zipp>=3.1.0 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from importlib-resources>=3.2.0->matplotlib->joypy) (3.17.0) Requirement already satisfied: six>=1.5 in c:\users\tejan\appdata\roaming\python\python39\site-packages (from python-dateutil>=2.8.1->pandas>=0.20.0->joypy) (1.16.0) Downloading numpy-1.24.4-cp39-cp39-win\_amd64.whl (14.9 MB)

14.9/14.9 MB 738.9 kB/s eta 0:00:00

Installing collected packages: numpy, joypy Attempting uninstall: numpy

Found existing installation: numpy 1.26.2 Uninstalling numpy-1.26.2:

Successfully uninstalled numpy-1.26.2 Successfully installed joypy-0.2.6 numpy-1.21.5

Note: you may need to restart the kernel to use updated packages.

WARNING: Failed to remove contents in a temporary directory 'C:\Users\tejan\AppData\Roaming\Python\Python39\site-packages\~umpy.libs'. You can safely remove it manually.

WARNING: Failed to remove contents in a temporary directory 'C:\Users\tejan\AppData\Roaming\Python\Python39\site-packages\~umpy'.

You can safely remove it manually.

ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts. daal4py 2021.6.0 requires daal==2021.4.0, which is not installed.

tensorflow 2.10.0 requires protobuf<3.20,>=3.9.2, but you have protobuf 3.20.3 which is incompatible.

numba 0.55.1 requires numpy<1.22,>=1.18, but you have numpy 1.24.4 which is incompatible.

tensorboard 2.10.1 requires protobuf<3.20,>=3.9.2, but you have protobuf 3.20.3 which is incompatible. tensorflow-intel 2.11.0 requires keras<2.12,>=2.11.0, but you have keras 2.10.0 which is incompatible. tensorflow-intel 2.11.0 requires protobuf<3.20,>=3.9.2, but you have protobuf 3.20.3 which is incompatible. tensorflow-intel 2.11.0 requires tensorboard<2.12,>=2.11, but you have tensorboard 2.10.1 which is incompatible.

tensorflow-intel 2.11.0 requires tensorflow-estimator<2.12,>=2.11.0, but you have tensorflow-estimator 2.10.0 which is incompatible.

In [2]:

*#15. Aim: To draw Ridgeline plot for given user input data*

**import** matplotlib.pyplot **as** plt

**from** joypy **import** joyplot

n**=**int(input("Enter the number of data sets: ")) data**=**[]

**for** i **in** range(n):

data**.**append(list(map(float,input(f"Enter data for dataset {i**+**1}: ")**.**split())))

fig, ax **=** joyplot(data, labels**=**[f'Dataset {i**+**1}' **for** i **in** range(n)], colormap**=**plt**.**cm**.**Blues\_r) plt**.**xlabel('Value')

plt**.**ylabel('Density') plt**.**title('Joyplot of datasets') plt**.**show()

In [ ]:

Enter the number of data sets: 3

Enter data for dataset 1: 1.5 2 2.5 3 3.5

Enter data for dataset 2: 2 2.5 3 3.5 4

Enter data for dataset 3: 1 1.5 2 2.5 3

