

### Practical 1a

Aim: Apply the what-if analysis for data visualization. Design and generate necessary reports based on the data warehouse data.

- Given the data for Concert Budget which requires the cost as follows:

Venue (total number of seats), Talent (performances), Venue Rent, Sound, Lights, Tickets, Security, Insurance.

Calculate the total cost. Calculate the revenue generated – Price per ticket, Ticket sales, Merchandising, Food and beverage, Total revenue.

Apply the formula to calculate profit or loss.

Note:

Cost of one merchandise = 5rs

Cost of one food box = 15rs

Price per tickets = 30rs

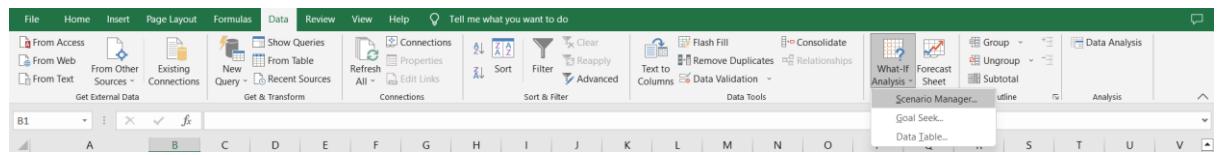
Steps:

	A	B
1		
2	<b>Concert Budget</b>	
3		
4	<b>Costs</b>	
5	Venue (No. of seats)	300
6		
7		
8	Talent	4,000
9	Venue rent	500
10	Sound	300
11	Lights	250
12	Tickets incharge	200
13	Security	150
14	Insurance	100
15	<b>Total Costs</b>	5,500
16		
17	<b>Revenues</b>	
18	Price/ticket	30
19		
20	Ticket sales	9000
21	Merchandising	1500
22	Food and Beverage	4500
23	<b>Total Revenue</b>	15000
24		
25	<b>Profit or Loss</b>	9500

### Formulas:

1. Calculating Total cost:  $\text{SUM}(\text{B8:B14})$  or  $\text{SUM}(\text{Talent} + \text{Venue Rental} + \text{sound} + \text{lights} + \text{Ticket Incharge} + \text{security} + \text{insurance})$
2. Calculating Ticket sales:  $\text{B19} * \text{B6}$  or  $\text{Ticket} * \text{venue} (\text{no of seats})$
3. Calculating Merchandising:  $\text{B6} * 5$  or  $\text{venue} (\text{no of seats}) * 5$
4. Calculating Food and Beverage:  $\text{B6} * 15$  or  $\text{venue} (\text{no of seats}) * 15$
5. Calculating Total Revenue:  $\text{SUM}(\text{B21:B23})$  or  $\text{SUM}(\text{Ticket sales} + \text{Merchandising} + \text{Food and Beverage})$
6. Calculating Profit and loss :  $\text{B24-B15}$  or  $(\text{Total Revenue} - \text{Total cost})$

Create a scenario manager for original data, medium size venue, large size venue and very large size venue.



The screenshot shows the Microsoft Excel ribbon with the 'Data' tab selected. The ribbon includes tabs for File, Home, Insert, Page Layout, Formulas, Data, Review, View, and Help. Under the Data tab, there are sections for Get External Data, Connections, Sort & Filter, Data Tools, and Analysis. The Analysis section contains buttons for Scenario Manager, What-If Analysis, Group, Ungroup, Subtotal, and Data Analysis.

**Scenario Manager Dialog Box:**

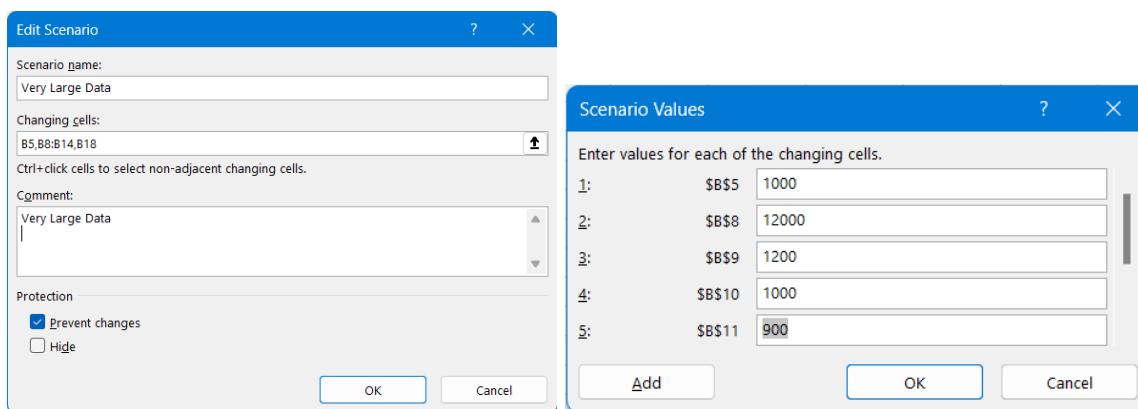
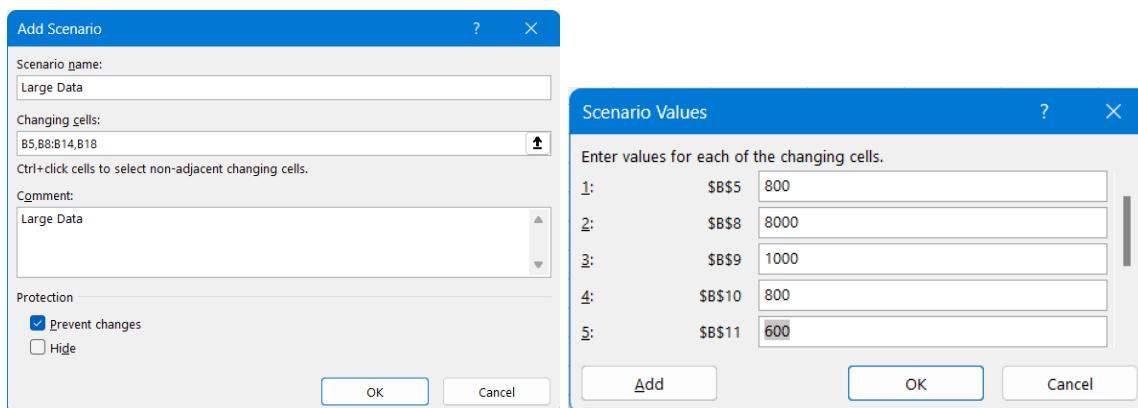
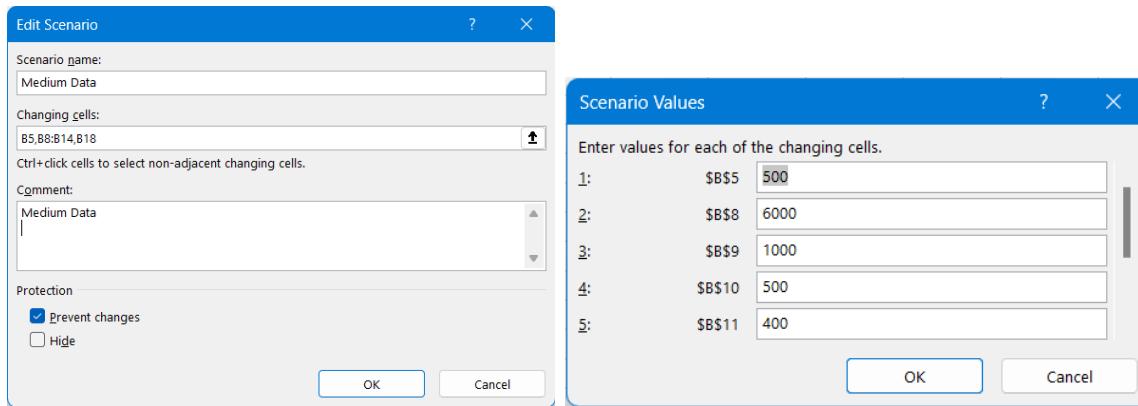
The 'Scenario Manager' dialog box is open. It has a 'Scenarios:' list box which displays 'No Scenarios defined. Choose Add to add'. Below the list box are five buttons: 'Add...', 'Delete', 'Edit...', 'Merge...', and 'Summary...'. There are also 'Changing cells:' and 'Comment:' input fields. At the bottom are 'Show' and 'Close' buttons.

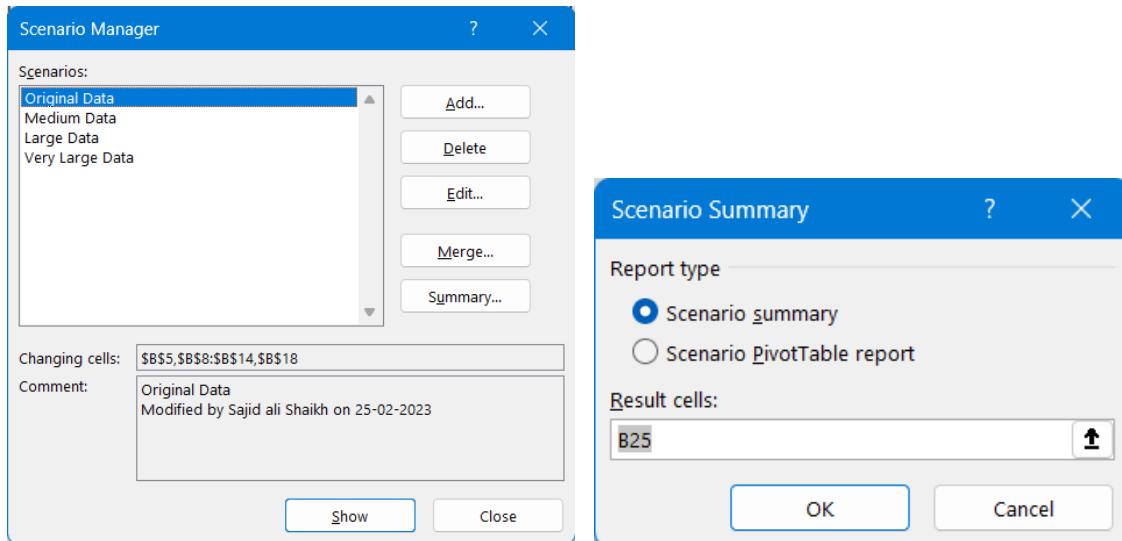
**Edit Scenario Dialog Box:**

The 'Edit Scenario' dialog box is open. It shows a 'Scenario name:' field containing 'Original Data', a 'Changing cells:' field containing '\$B\$5,\$B\$8:\$B\$14,\$B\$18', a 'Comment:' field containing 'Original Data', and a 'Protection' section with checkboxes for 'Prevent changes' (checked) and 'Hide'. At the bottom are 'OK' and 'Cancel' buttons.

**Scenario Values Dialog Box:**

The 'Scenario Values' dialog box is open. It has a title 'Enter values for each of the changing cells.' and a table with five rows. The table lists changing cells and their corresponding values: 1: \$B\$5 (300), 2: \$B\$8 (4000), 3: \$B\$9 (500), 4: \$B\$10 (300), and 5: \$B\$11 (250). At the bottom are 'OK' and 'Cancel' buttons.





	Current Values:	Original Data	Medium Data	Large Data	Very Large Data	
Changing Cells:	\$B\$5 \$B\$8 \$B\$9 \$B\$10 \$B\$11 \$B\$12 \$B\$13 \$B\$14 \$B\$18	300 4,000 500 300 250 200 150 100 30	300 4,000 500 300 250 200 150 100 30	500 6,000 1000 500 400 400 300 300 50	800 8,000 1000 800 600 500 400 400 100	1000 12,000 1200 1000 900 850 800 800 120
Result Cells:	\$B\$25	9500	9500	6100	3300	-2550

Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.

2. Consider a scenario for a family's budget and expense.

The budget is as follows: Father's income, Mother's income and income from other sources will give the total income.

The expenses for the family consist of maintenance, electricity bill, internet charges, food, new clothes which gives the total expenses.

Calculate the formula for total income, total expenses and final savings.

A	B
31	
32 Incomes	
33	
34 Father Income	50,000
35 Mother Income	15,000
36 Other sources Income	5,000
37	
38 Total Income	70,000
39	
40	
41 Expensives	
42	
43 Maintenance	1,000
44 Electricity	1,500
45 Internet Charges	1,000
46 Food	10,000
47 New Clothes	1,000
48	
49 Total Expenses	14,500
50	
51 Savings	55,500

Create three scenarios for current savings, low savings and high savings by changing the values for food and clothes.

The image shows two overlapping dialog boxes from Microsoft Excel:

- Edit Scenario Dialog (Left):**
  - Scenario name: Low Savings
  - Changing cells: \$B\$34:\$B\$36,\$B\$43:\$B\$47
  - Comment: Low Savings
  - Protection: Prevent changes (checkbox checked)
- Scenario Values Dialog (Right):**
  - Enter values for each of the changing cells.
  - Values listed:
    - 1: \$B\$43 1200
    - 2: \$B\$44 2500
    - 3: \$B\$45 1500
    - 4: \$B\$46 20000
    - 5: \$B\$47 5000
  - Buttons: OK and Cancel

**Add Scenario**

Scenario name: High Savings

Changing cells: B34:B36,B43:B47

Comment: High Savings

Protection:  Prevent changes  Hide

**Scenario Values**

Enter values for each of the changing cells.

1:	\$B\$43	1000
2:	\$B\$44	1500
3:	\$B\$45	1000
4:	\$B\$46	15000
5:	\$B\$47	3000

**Scenario Summary**

Report type:  Scenario summary  Scenario PivotTable report

Result cells: =B\$51

**Scenario Summary**

	Current Values:	Original Data	Low Savings	High Savings
<b>Changing Cells:</b>	\$B\$34 \$B\$35 \$B\$36 \$B\$43 \$B\$44 \$B\$45 \$B\$46 \$B\$47	50,000 15,000 5,000 1,200 2,500 1,500 20,000 5,000	50,000 15,000 5,000 1,000 1,500 1,000 10,000 1,000	50,000 15,000 5,000 1,200 2,500 1,500 20,000 5,000
<b>Result Cells:</b>	\$B\$51	39,800	55,500	48,500

Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.

## Practical 1b

Aim: Implementing goal seek function in excel.

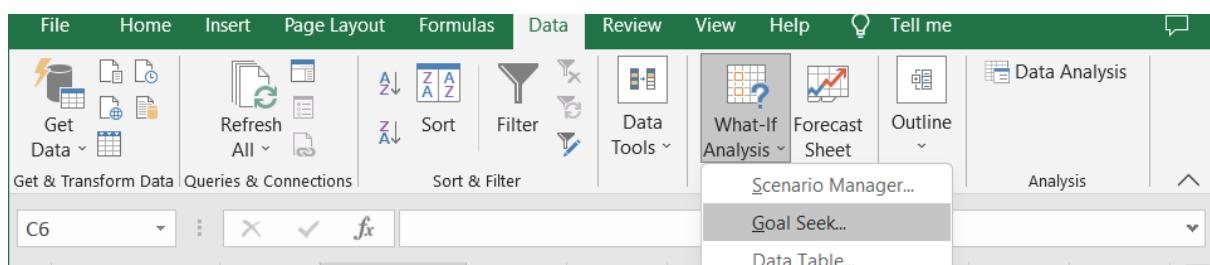
- Given the data about no. of items sold (quantity), price per item find the revenue.

Using goal seek function how many items do you have to sell to make 1000 rupees as the revenue.

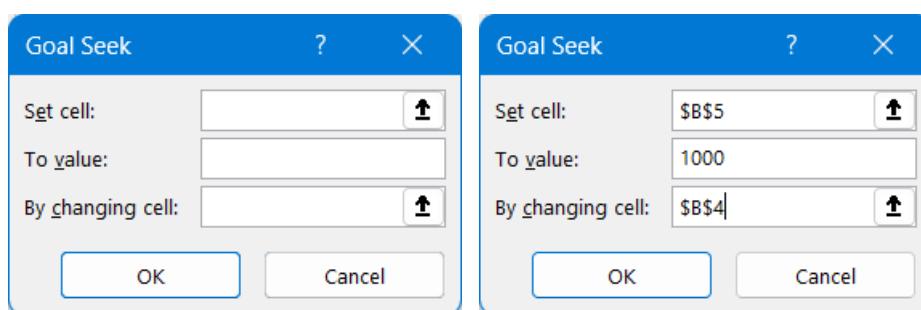
Calculate Revenue: Item price \* Quantity

2	<b>Example 1</b>	
3	Item Price	5
4	Quantity	100 variable data
5	Revenue	500

On the Data tab, go What-If Analysis and click on the Goal Seek option. The Goal Seek dialog box appears.



Set cell value: B5, Set To value: 1000, By changing cell: B4

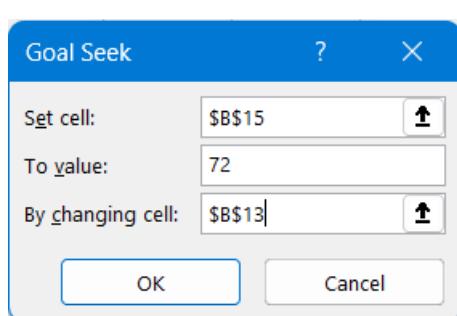


Click on ok

2	<b>Example 1</b>	
3	Item Price	5
4	Quantity	200 variable data
5	Revenue	1000

2. At the end of the course a student takes 3 exams. The passing score is 72. All the exams have the same weight, so the overall score is calculated by taking the average of all 3 exams. The student has already taken 2 out of 3 exams. What score does the student need to get for the third exam to pass the entire course?

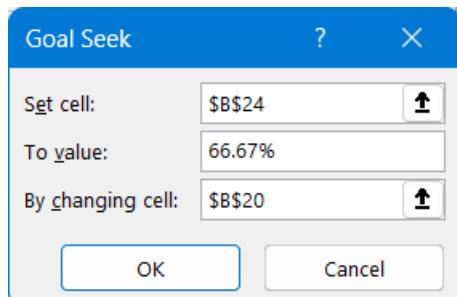
8	<b>Example 2</b>	
9	Marks	
10	Exam	Score
11	Exam 1	70
12	Exam 2	62
13	Exam 3	
14		
15	Final Score	66



8	<b>Example 2</b>	
9	Marks	
10	Exam	Score
11	Exam 1	70
12	Exam 2	62
13	Exam 3	84
14		
15	Final Score	72

3. You are running for some elected positions where a  $\frac{2}{3}$ rd of majority (66.67% of votes) is required to win the election. Assuming there are 200 total voting members, how many votes do you have to secure? Currently you have 98 votes.

18	<b>Example 3</b>	
19	Election	
20	Current votes	98
21	Total voters	200
22		
23		
24	Required votes	49.00%



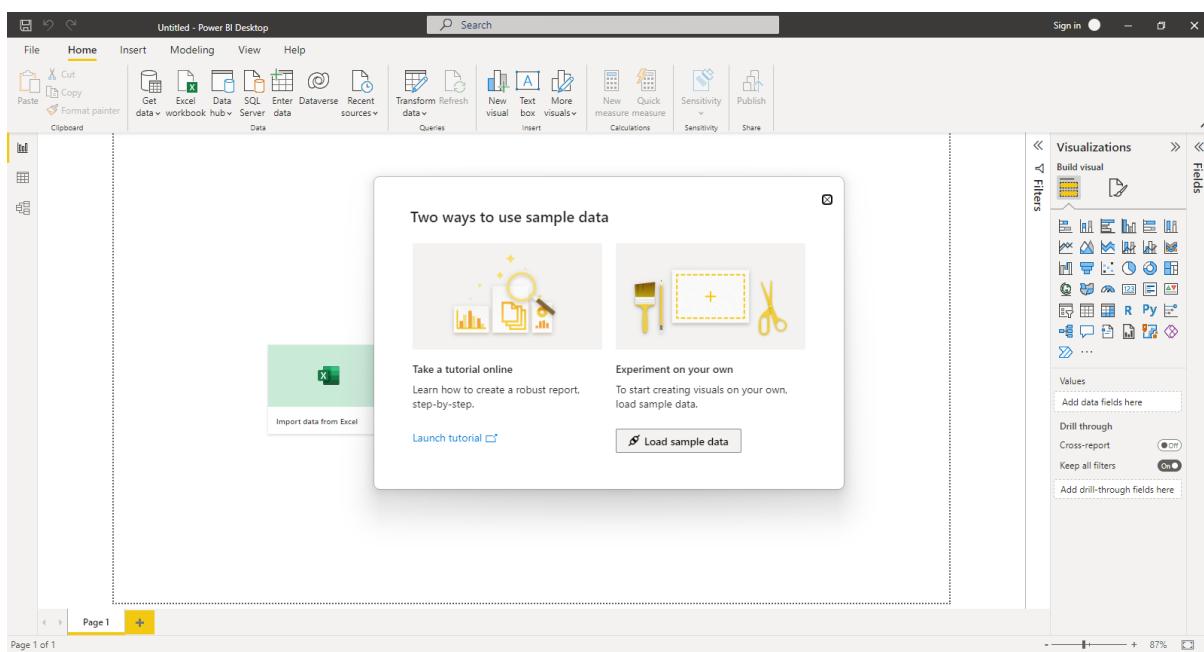
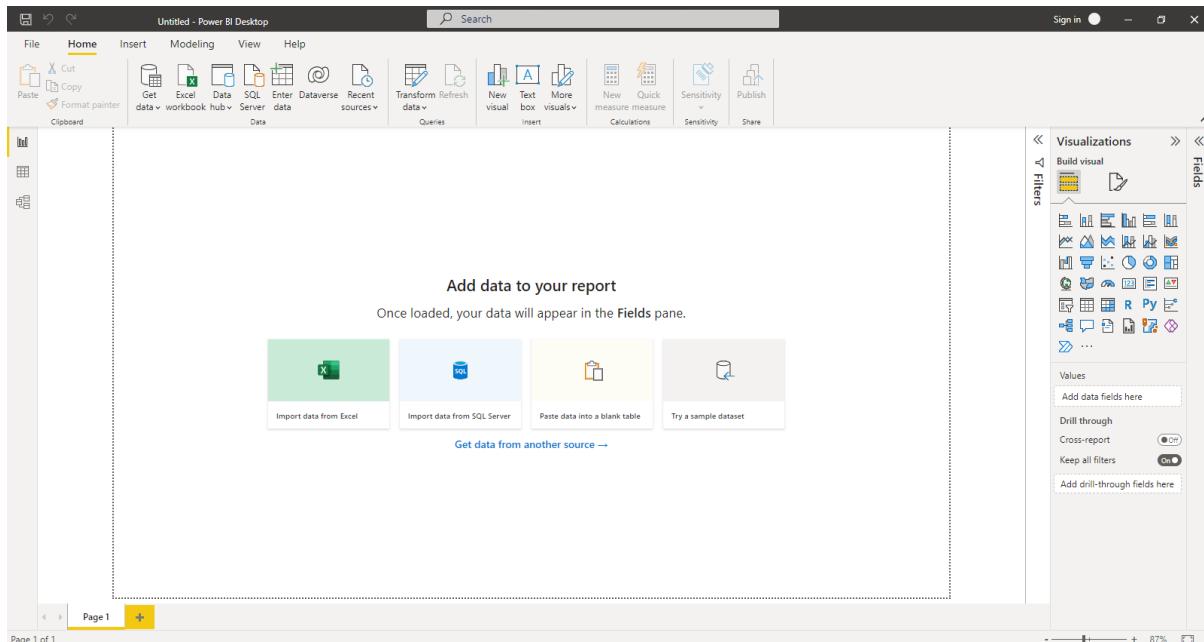
18	<b>Example 3</b>	
19	Election	
20	Current votes	133.34
21	Total voters	200
22		
23		
24	Required votes	66.67%

## Practical 2

Aim: Import the legacy data from different sources and load in the target system using Power BI

(you can download sample databases such as adventure works, north wind, food mart, financial, etc.)

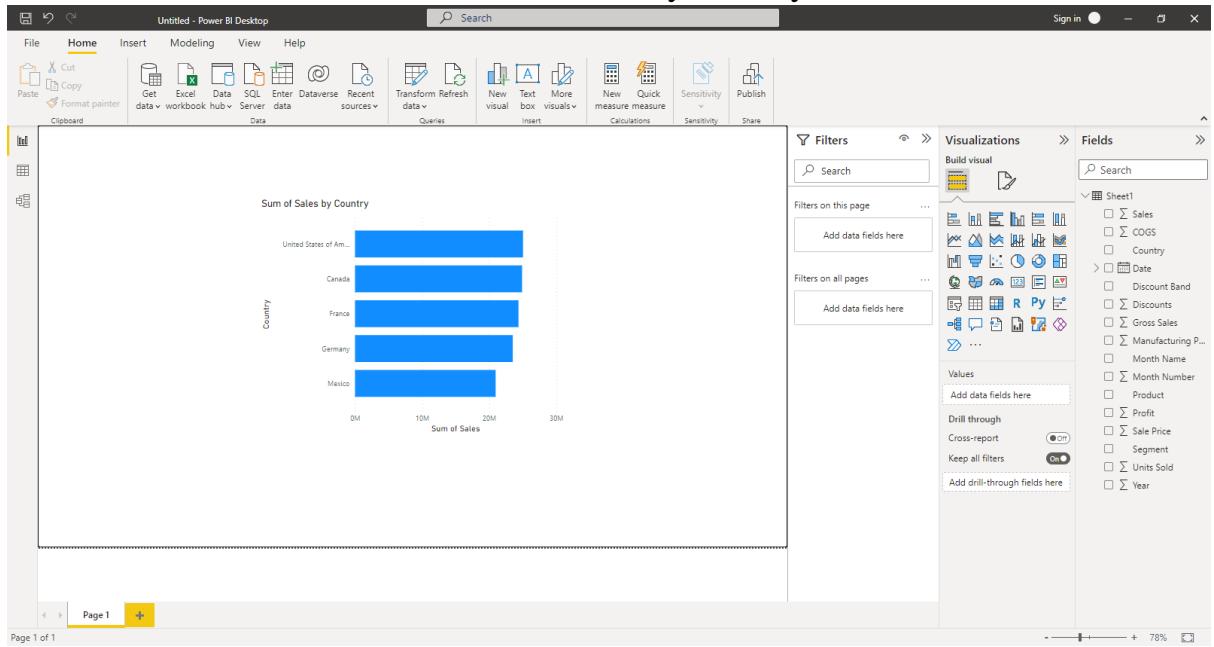
Consider the financial dataset and perform the following:



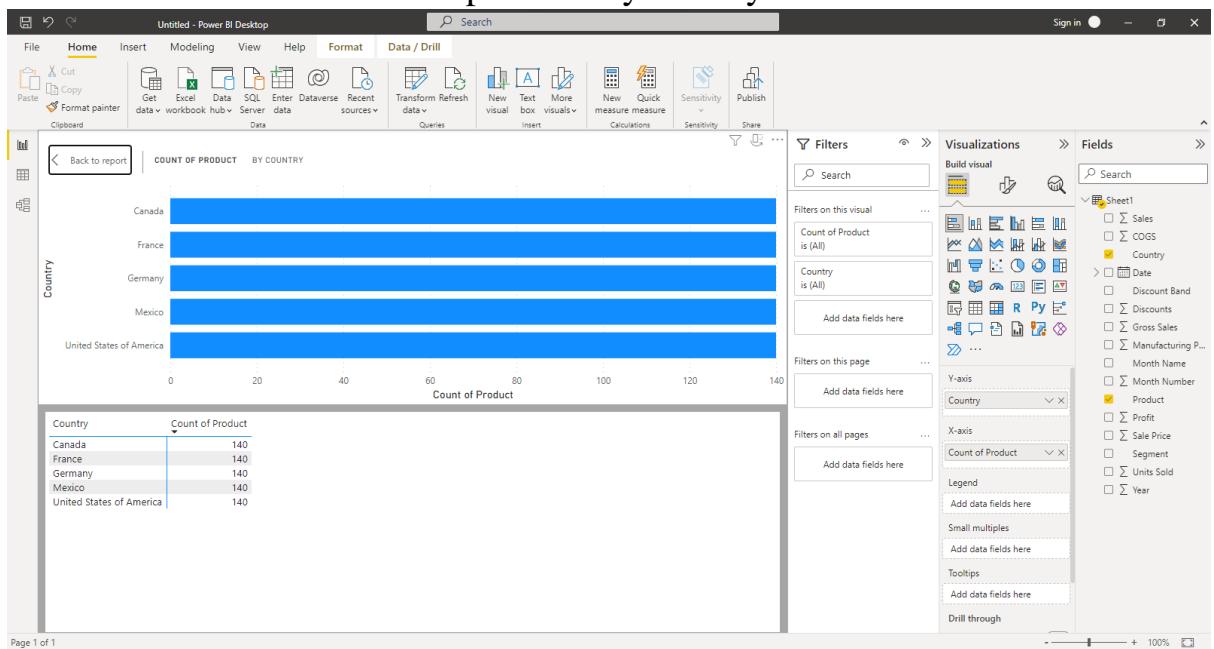
Screenshot of Power BI Desktop showing the Navigator pane open. The Navigator pane displays a hierarchical list of data sources and tables from a file named "Financial Sample.xlsx". The "financials" table is selected, showing its schema with columns: Segment, Country, Product, Discount Band, and Unit. The pane also includes options for "Load", "Transform Data", and "Cancel". The main canvas area is empty, and the ribbon shows the Home tab selected.

Screenshot of Power BI Desktop showing the Home tab selected. The main canvas area displays the message "Build visuals with your data" and "Select or drag fields from the Fields pane onto the report canvas." The Fields pane on the right lists various data fields categorized under "Sheet1" and "Date". The ribbon shows the Home tab selected.

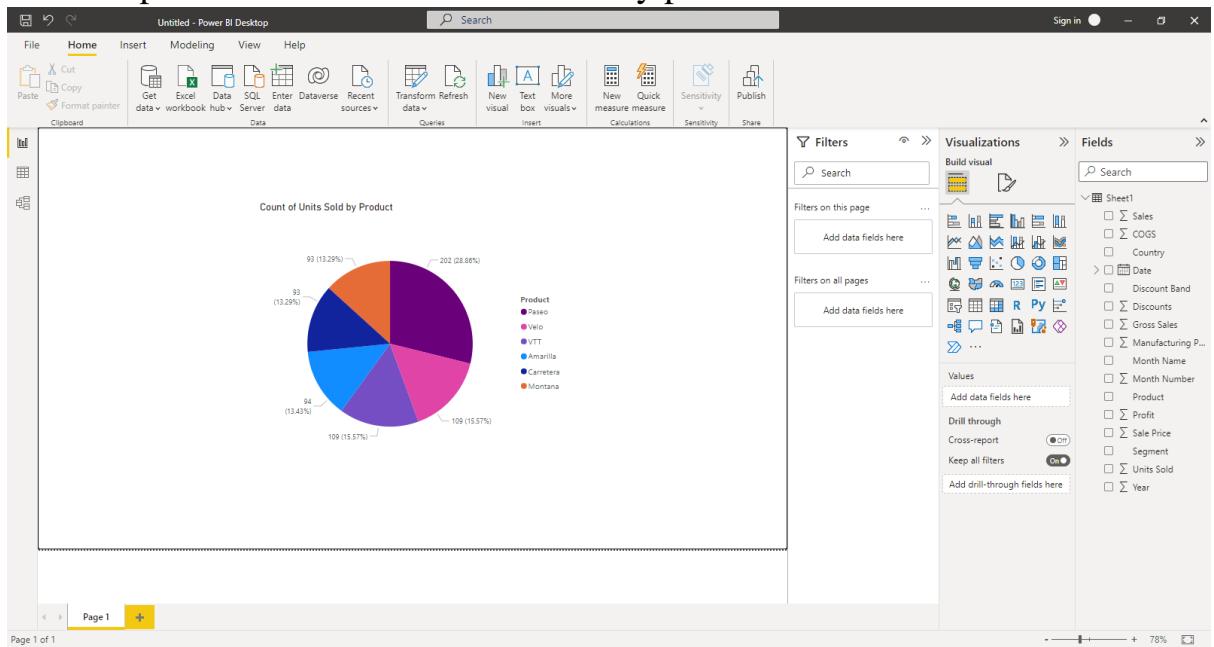
1. Create a stacked bar chart for sum of sales by country.



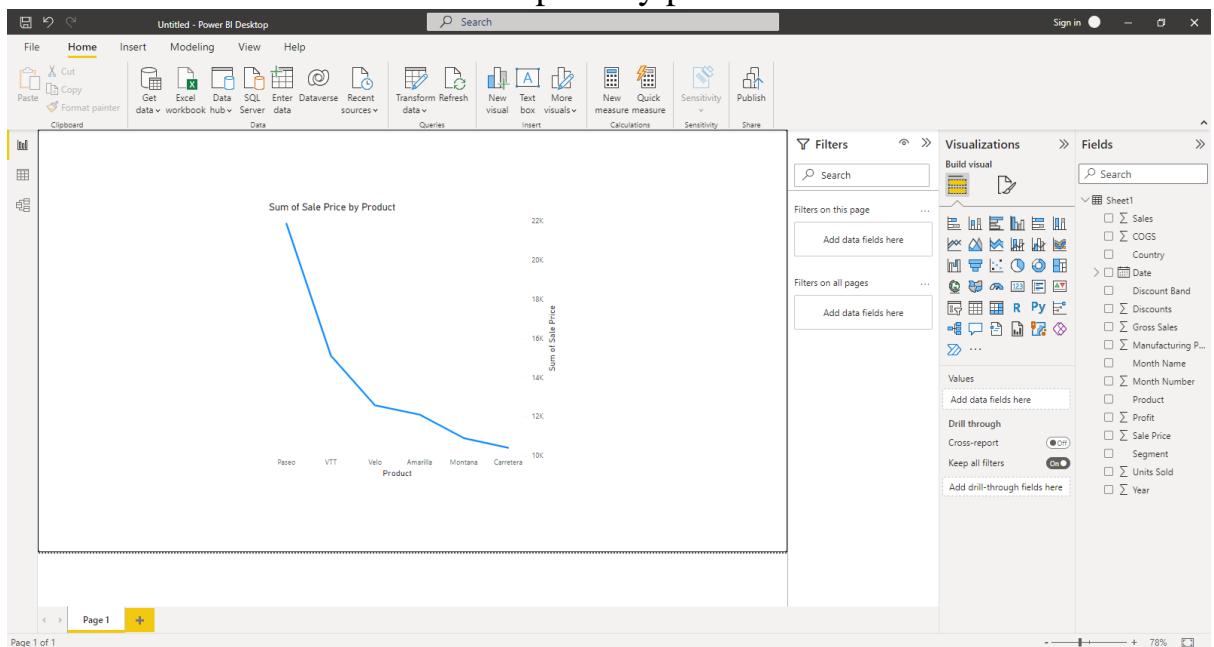
2. Create a bar chart for count of products by country.



**3. Create a pie chart for count of units sold by products.**



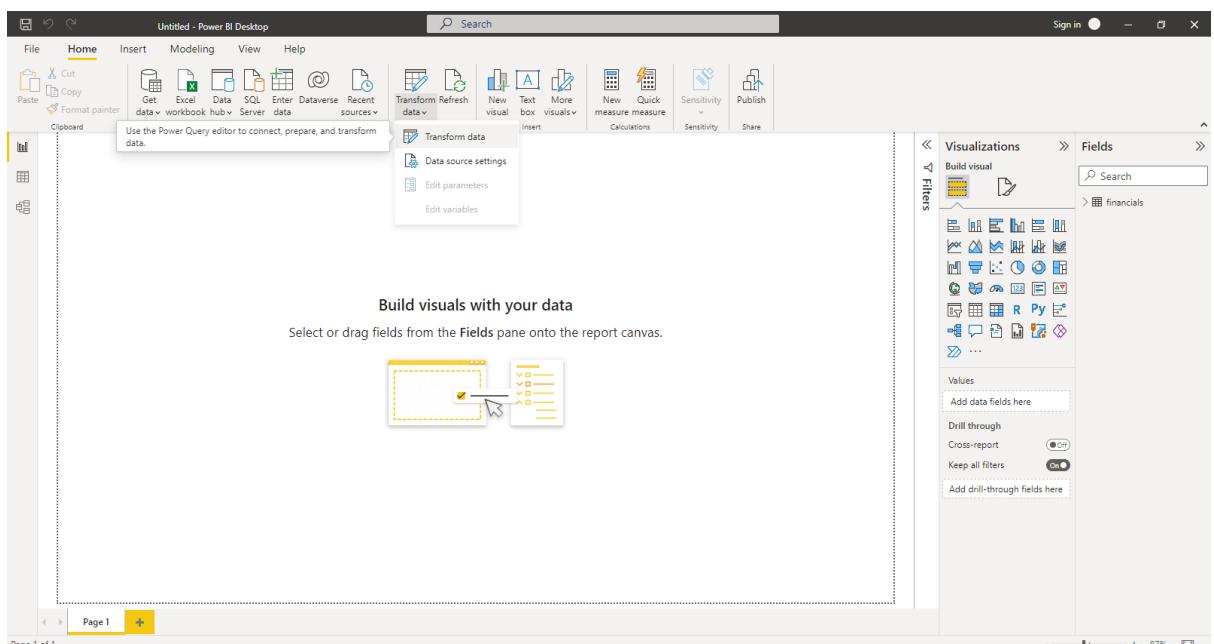
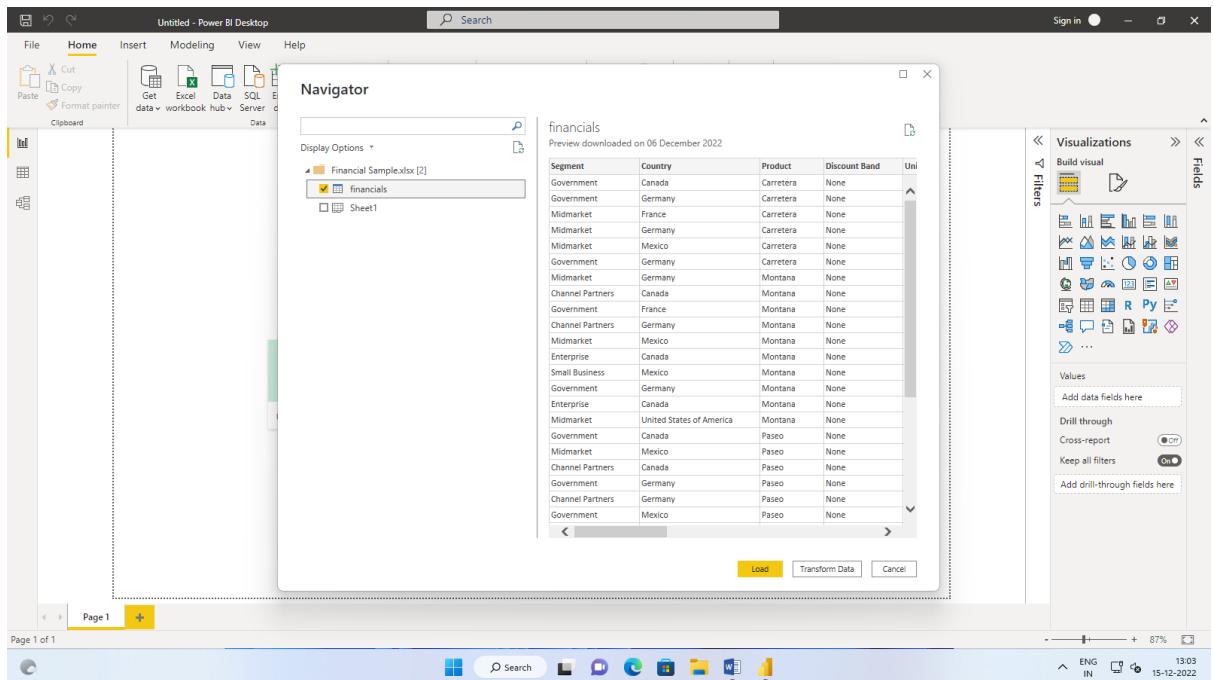
**4. Create a line chart for sum of sales price by product.**



### Practical 3

Aim: Perform the ETL process to construct the database in Power BI

1. Load the dataset financial and click on Transform data tab.



16 COLUMNS, 700 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

## 2. Change the table name.

16 COLUMNS, 700 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

16 COLUMNS, 700 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

### 3. Replace values (United States of America to USA).

16 COLUMNS, 700 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

Untitled - Power Query Editor

**File Home Transform View Tools Help**

**Queries [1]** **fx** = Table.TransformColumnTypes(Financials\_Table, {{"Segment", type text}, {"Country", type text}, {"Product", type text}, {"Discount Band", type text}, {"Units Sold", type number}, {"Manufacturing Price", type number}, {"Sale Price", type number}}

**Finance Data**

Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price
Government	Canada	Carretera	None	1618.5	3	5
Government	Germany	Carretera	None	1321	3	5
Midmarket	France	Carretera	None	2178	3	5
Midmarket	Germany	Carretera	None	---	3	5
Midmarket	Mexico	Carretera	None	---	3	5
Government	Germany	Carretera	None	---	5	5
Channel Partners	Canada	Carretera	None	---	5	5
Government	France	Carretera	None	---	5	5
Channel Partners	Germany	Carretera	None	---	5	5
Midmarket	Mexico	Carretera	None	---	5	5
Enterprise	Canada	Carretera	None	---	5	5
Small Business	Mexico	Carretera	None	---	5	5
Government	Germany	Carretera	None	---	5	5
Enterprise	Canada	Carretera	None	---	5	5
Midmarket	United States	Carretera	None	---	10	10
Government	Canada	Paseo	None	2518	10	10
Midmarket	Mexico	Paseo	None	1006	10	10
Channel Partners	Canada	Paseo	None	367	10	10
Government	Mexico	Paseo	None	883	10	10
Midmarket	France	Paseo	None	549	10	10
United States	United States	Paseo	None	---	10	10

**Replace Values**

Replace one value with another in the selected columns.

**Value To Find:** United States of America

**Replace With:** USA

**OK Cancel Advanced options**

PREVIEW DOWNLOADED ON 06 DECEMBER 2022

16 COLUMNS, 700 ROWS Column profiling based on top 1000 rows

13:05 ENG IN 15-12-2022

Untitled - Power Query Editor

**File Home Transform View Tools Help**

**Queries [1]** **fx** = Table.ReplaceValue("Changed Type", "United States of America", "USA", Replacer.ReplaceText, {"Country"})

**Finance Data**

Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price
Government	Canada	Carretera	None	1618.5	3	5
Government	Germany	Carretera	None	1321	3	5
Midmarket	France	Carretera	None	2178	3	5
Midmarket	Germany	Carretera	None	888	3	5
Midmarket	Mexico	Carretera	None	2470	3	5
Government	Germany	Carretera	None	1513	3	5
Midmarket	Germany	Montana	None	921	5	5
Channel Partners	Canada	Montana	None	2518	5	5
Government	France	Montana	None	1899	5	5
Channel Partners	Germany	Montana	None	1545	5	5
Midmarket	Mexico	Montana	None	2470	5	5
Enterprise	Canada	Montana	None	2655.5	5	5
Small Business	Mexico	Montana	None	958	5	5
Government	Germany	Montana	None	2146	5	5
Enterprise	Canada	Montana	None	345	5	5
Midmarket	USA	Montana	None	615	5	5
Government	Canada	Paseo	None	292	10	10
Midmarket	Mexico	Paseo	None	974	10	10
Channel Partners	Canada	Paseo	None	2518	10	10
Government	Germany	Paseo	None	1006	10	10
Channel Partners	Germany	Paseo	None	367	10	10
Government	Mexico	Paseo	None	883	10	10
Midmarket	France	Paseo	None	549	10	10
Small Business	Mexico	Paseo	None	788	10	10
Midmarket	Mexico	Paseo	None	2472	10	10
Government	USA	Paseo	None	1143	10	10
Government	Canada	Paseo	None	1725	10	10
Channel Partners	USA	Paseo	None	912	10	10
Midmarket	Canada	Paseo	None	2152	10	10

**Query Settings**

**PROPERTIES** Name: Finance Data All Properties

**APPLIED STEPS** Source: Navigation: Changed Type: Replaced Value

PREVIEW DOWNLOADED ON 06 DECEMBER 2022

16 COLUMNS, 700 ROWS Column profiling based on top 1000 rows

#### 4. Rename the column segment to segment name.

The screenshot shows the Power Query Editor interface with the 'Finance Data' query selected. In the 'Segment' column, the value 'Segment' is highlighted, and a context menu is open with the option 'Rename...' selected. The 'APPLIED STEPS' pane on the right shows the step 'Replaced Value' has been applied to the 'Segment' column.

Segment	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price
Government	Canada	Carretera	None	1618.5	3	
Government	Germany	Carretera	None	1321	3	
Midmarket	France	Carretera	None	2178	3	
Midmarket	Germany	Carretera	None	888	3	
Midmarket	Mexico	Carretera	None	2470	3	
Government	Montana	Carretera	None	1513	3	
Midmarket	Montana	Carretera	None	921	5	
Channel Partners	Montana	Carretera	None	2518	5	
Government	Montana	Carretera	None	1899	5	
Channel Partners	Montana	Carretera	None	1545	5	
Midmarket	Montana	Carretera	None	2470	5	
Enterprise	Montana	Carretera	None	2665.5	5	
Small Business	Montana	Carretera	None	958	5	
Government	Montana	Carretera	None	2146	5	
Enterprise	Montana	Carretera	None	345	5	
Midmarket	Montana	Carretera	None	615	5	
Government	Montana	Carretera	None	292	10	
Channel Partners	Montana	Carretera	None	974	10	
Government	Montana	Carretera	None	2518	10	
Channel Partners	Montana	Carretera	None	1006	10	
Government	Montana	Carretera	None	367	10	
Channel Partners	Montana	Carretera	None	883	10	
Government	Montana	Carretera	None	1725	10	
Channel Partners	Montana	Carretera	None	912	10	
Midmarket	Montana	Carretera	None	2152	10	
Small Business	Mexico	Paseo	None	2472	10	
Midmarket	Mexico	Paseo	None	1143	10	
Government	Mexico	Paseo	None	1725	10	
Channel Partners	Mexico	Paseo	None	912	10	
Midmarket	Mexico	Paseo	None	2152	10	

The screenshot shows the Power Query Editor interface with the 'Finance Data' query selected. The 'Segment' column has been renamed to 'Segment Name'. The 'APPLIED STEPS' pane on the right shows the step 'Renamed Columns' has been applied to the 'Segment' column.

Segment Name	Country	Product	Discount Band	Units Sold	Manufacturing Price	Sale Price
Government	Canada	Carretera	None	1618.5	3	
Government	Germany	Carretera	None	1321	3	
Midmarket	France	Carretera	None	2178	3	
Midmarket	Germany	Carretera	None	888	3	
Midmarket	Mexico	Carretera	None	2470	3	
Government	Montana	Carretera	None	1513	3	
Midmarket	Montana	Carretera	None	921	5	
Channel Partners	Montana	Carretera	None	2518	5	
Government	Montana	Carretera	None	1899	5	
Channel Partners	Montana	Carretera	None	1545	5	
Midmarket	Montana	Carretera	None	2470	5	
Enterprise	Montana	Carretera	None	2665.5	5	
Small Business	Montana	Carretera	None	958	5	
Government	Montana	Carretera	None	2146	5	
Enterprise	Montana	Carretera	None	345	5	
Midmarket	Montana	Carretera	None	615	5	
Government	Montana	Carretera	None	292	10	
Channel Partners	Montana	Carretera	None	974	10	
Government	Montana	Carretera	None	2518	10	
Channel Partners	Montana	Carretera	None	1006	10	
Government	Montana	Carretera	None	367	10	
Channel Partners	Montana	Carretera	None	883	10	
Government	Montana	Carretera	None	1725	10	
Channel Partners	Montana	Carretera	None	912	10	
Midmarket	Montana	Carretera	None	2152	10	
Small Business	Mexico	Paseo	None	2472	10	
Midmarket	Mexico	Paseo	None	1143	10	
Government	Mexico	Paseo	None	1725	10	
Channel Partners	Mexico	Paseo	None	912	10	
Midmarket	Mexico	Paseo	None	2152	10	

## 5. Remove the first row from the dataset.

The screenshot shows the Power Query Editor interface with the 'Transform' tab selected. A context menu is open over the top row of the table, with the 'Remove Top Rows' option highlighted. The 'APPLIED STEPS' pane on the right shows the step 'Renamed Columns' has been applied. The table contains 700 rows and 16 columns, with the top row being removed.

The screenshot shows the Power Query Editor with the 'Transform' tab selected. A 'Remove Top Rows' dialog box is open, prompting the user to specify the number of rows to remove from the top. The 'Number of rows' input field contains the value '5'. The 'OK' button is highlighted. The 'APPLIED STEPS' pane on the right shows the step 'Renamed Columns' has been applied. The table contains 700 rows and 16 columns, with the top 5 rows removed.

The screenshot shows the Power Query Editor interface with a table named "Finance Data". The table has 16 columns and 699 rows. The columns are: Segment Name, Country, Product, Discount Band, Units Sold, Manufacturing Price, and Sale Price. The "Applied Steps" pane on the right shows the "Removed Top Rows" step.

## 6. Remove the column for day, month and year.

The screenshot shows the Power Query Editor interface with a table named "Finance Data". The table has 16 columns and 699 rows. The columns are: Sales, L2 COGS, L2 Profit, Date, Month Number, Month Name, and Year. A context menu is open over the "Year" column, showing options like Copy, Remove, and Replace Values.

Untitled - Power Query Editor

**Queries [1]**

Finance Data

Discounts	1.2 Sales	1.2 COGS	1.2 Profit	Date	Month Number	Month Name	
1	0	26420	13210	01-01-2014	1	January	
2	0	32670	21780	01-06-2014	6	June	
3	0	13320	8880	01-06-2014	6	June	
4	0	37050	24700	01-06-2014	6	June	
5	0	529550	393380	136170	12	December	
6	0	13815	9210	01-03-2014	3	March	
7	0	30216	7554	01-06-2014	6	June	
8	0	37980	18990	01-06-2014	6	June	
9	0	18540	4635	13905	01-06-2014	6	June
10	0	37050	24700	01-06-2014	6	June	
11	0	333187.5	319860	13327.5	01-07-2014	7	July
12	0	287400	239500	47900	01-08-2014	8	August
13	0	15022	10730	4292	01-09-2014	9	September
14	0	49125	41400	1725	01-10-2014	10	October
15	0	9225	6150	3075	01-12-2014	12	December
16	0	5840	2920	2920	01-02-2014	2	February
17	0	14610	9740	4870	01-02-2014	2	February
18	0	30216	7554	22662	01-06-2014	6	June
19	0	352100	261560	90540	01-06-2014	6	June
20	0	4404	1101	3303	01-07-2014	7	July
21	0	6181	4415	1766	01-08-2014	8	August
22	0	8235	5490	2745	01-09-2014	9	September
23	0	236400	197000	39400	01-09-2014	9	September
24	0	37080	24720	12380	01-09-2014	9	September
25	0	8001	5715	2286	01-10-2014	10	October
26	0	603750	448500	155250	01-11-2014	11	November
27	0	10944	2736	8208	01-11-2014	11	November
28	0	32280	21520	10760	01-12-2014	12	December
29	0	36340	18170	18170	01-12-2014	12	December
30							

15 COLUMNS, 699 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

Untitled - Power Query Editor

**Queries [1]**

Finance Data

Sale Price	1.2 Gross Sales	1.2 Discounts	1.2 Sales	1.2 COGS	1.2 Profit	Date
20	26420	0	26420	13210	13210	01-01-2014
15	32670	0	32670	10890	10890	01-06-2014
15	13320	0	13320	8880	4440	01-06-2014
15	37050	0	37050	24700	12250	01-06-2014
350	529550	0	529550	393380	136170	136170
15	13815	0	13815	9210	4605	01-03-2014
12	30216	0	30216	7554	22662	01-06-2014
20	37980	0	37980	18990	18990	01-06-2014
12	18540	0	18540	4635	13905	01-06-2014
15	37050	0	37050	24700	12250	01-06-2014
125	333187.5	0	333187.5	319860	13327.5	01-07-2014
300	287400	0	287400	239500	47900	01-08-2014
7	15022	0	15022	10730	4292	01-09-2014
125	49125	0	49125	41400	1725	01-10-2014
15	9225	0	9225	6150	3075	01-11-2014
20	5840	0	5840	2920	2920	01-12-2014
15	14610	0	14610	9740	4870	01-12-2014
12	30216	0	30216	7554	22662	01-12-2014
350	352100	0	352100	261560	90540	01-12-2014
12	4404	0	4404	1101	3303	01-12-2014
7	6181	0	6181	4415	1766	01-12-2014
15	8235	0	8235	5490	2745	01-12-2014
300	236400	0	236400	197000	39400	01-12-2014
15	37080	0	37080	24720	12380	01-12-2014
7	8001	0	8001	5715	2286	01-12-2014
350	603750	0	603750	448500	155250	01-12-2014
12	10944	0	10944	2736	8208	01-12-2014
15	32280	0	32280	21520	10760	01-12-2014
20	36340	0	36340	18170	18170	01-12-2014
30						

13 COLUMNS, 699 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

## 7. Apply number filter for the column manufacturing price that equals 5.

The screenshot shows the Power Query Editor interface. A context menu is open over the 'Manufacturing Price' column header. The 'Number Filters' option is selected, bringing up a submenu with the 'Equals...' option highlighted. Other filter options like 'Does Not Equal...', 'Greater Than...', etc., are also listed. The main table area displays 699 rows of data with columns for Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Gross Sales, and Discounts.

This screenshot shows the 'Filter Rows' dialog box open in the Power Query Editor. The 'equals' dropdown is set to '5'. The 'And' radio button is selected. The dialog box has 'OK' and 'Cancel' buttons. The main table below shows rows of data, with the fifth row in the Manufacturing Price column highlighted in yellow, indicating it matches the filter criteria.

12 COLUMNS, 93 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

## 8. Split the column segment name into 2 columns using the space delimiter.

12 COLUMNS, 93 ROWS Column profiling based on top 1000 rows PREVIEW DOWNLOADED ON 06 DECEMBER 2022

Untitled - Power Query Editor

**Queries [1]**

1. Finance Data

**Split Column by Delimiter**

Specify the delimiter used to split the text column.

Select or enter delimiter: Space

Split at: Each occurrence of the delimiter

Advanced options: Quote Character: " Insert special character: ▾

OK Cancel

PREVIEW DOWNLOADED ON 06 DECEMBER 2022

12 COLUMNS, 93 ROWS Column profiling based on top 1000 rows

Search ENG IN 15-12-2022

Untitled - Power Query Editor

**Queries [1]**

1. Finance Data

**Table.TransformColumnTypes#“Split Column by Delimiter”,{“Segment Name.1”, type text}, {“Segment Name.2”, type text}}**

PREVIEW DOWNLOADED ON 06 DECEMBER 2022

13 COLUMNS, 93 ROWS Column profiling based on top 1000 rows

Search ENG IN 15-12-2022

## 9. Merge the splitted columns.

The screenshot shows the Power Query Editor interface. The ribbon has 'Merge Columns' selected. A dialog box titled 'Merge Columns' is open, prompting the user to choose how to merge selected columns. The 'Separator' dropdown is set to 'Space', and a new column name 'Merged' is typed into the 'New column name (optional)' input field. The 'OK' button at the bottom right of the dialog is highlighted.

The screenshot shows the Power Query Editor interface. The ribbon has 'Merge Columns' selected. A dialog box titled 'Merge Columns' is open, prompting the user to choose how to merge selected columns. The 'Separator' dropdown is set to 'Space', and a new column name 'Merged' is typed into the 'New column name (optional)' input field. The 'OK' button at the bottom right of the dialog is highlighted.

The screenshot shows the Power Query Editor interface with the following details:

- File**, **Home**, **Transform**, **Add Column**, **View**, **Tools**, **Help** menu items.
- Queries [1]** dropdown showing "Finance Data".
- Table.AddColumn** formula bar: `= Table.AddColumn(#"Changed Type1", "Merged", each Text.Combine({[Segment Name.1], [Segment Name.2]}, " "), type text)`
- Finance Data** table view with 14 columns and 93 rows. The columns are: Sale Price, #. Gross Sales, #. Discounts, #. Sales, #. COGS, #. Profit, and Merged.
- Properties** pane on the right showing "Name: Finance Data" and "All Properties".
- Applied Steps** pane on the right showing the steps taken: Source, Navigation, Changed Type, Replaced Value, Renamed Columns, Removed Top Rows, Removed Columns, Filtered Rows, Split Column by Delimiter, and Inserted Merged Column.
- Query Settings** pane on the right.
- Bottom status bar: "14 COLUMNS, 93 ROWS" and "Column profiling based on top 1000 rows".
- Bottom right corner: "PREVIEW DOWNLOADED ON 06 DECEMBER 2022".

## 10. Add a conditional column country abbreviations based on the column country.

The screenshot shows the Power Query Editor interface with the following details:

- File**, **Home**, **Transform**, **Add Column**, **View**, **Tools**, **Help** menu items.
- Queries [1]** dropdown showing "Finance Data".
- Table.AddColumn** formula bar: `= Table.AddColumn(#"Changed Type1", "Merged", each Text.Combine({[Segment Name.1], [Segment Name.2]}, " "), type text)`
- Finance Data** table view with 14 columns and 93 rows. The columns are: #. Segment Name.1, #. Segment Name.2, #. Country, #. Product, #. Discount Band, #. Units Sold, and #. Manufacturing Price.
- Properties** pane on the right showing "Name: Finance Data" and "All Properties".
- Applied Steps** pane on the right showing the steps taken: Source, Navigation, Changed Type, Replaced Value, Renamed Columns, Removed Top Rows, Removed Columns, Filtered Rows, Split Column by Delimiter, and Inserted Merged Column.
- Query Settings** pane on the right.
- Bottom status bar: "14 COLUMNS, 93 ROWS" and "Column profiling based on top 1000 rows".
- Bottom right corner: "PREVIEW DOWNLOADED ON 06 DECEMBER 2022".

Untitled - Power Query Editor

**Queries [1]**

Finance Data

Table.AddColumn("Changed Type1", "Merged", each Text.Combine({[Segment Name.1], [Segment Name.2]}, " "), type text)

	Segment Name.1	Segment Name.2	Country	Product	Discount Band	Units Sold	Manufacturing Price
1	Midmarket	null	Germany	Montana	None	921	921
2	Channel						
3	Government						
4	Channel						
5	Midmarket						
6	Enterprise						
7	Small						
8	Government						
9	Enterprise						
10	Midmarket						
11	Midmarket						
12	Small						
13	Government						
14	Government						
15	Small						
16	Enterprise						
17	Channel						
18	Government						
19	Channel						
20	Enterprise						
21	Government						
22	Channel	Partners	France	Montana	Low	1901	1901
23	Government	null	France	Montana	Low	544	544
24	Government	null	Germany	Montana	Low	1797	1797
25	Enterprise	null	France	Montana	Low	1287	1287
26	Enterprise	null	Germany	Montana	Low	1706	1706
27	Midmarket	null	Mexico	Montana	Low	2031	2031
28	Midmarket	null	Canada	Montana	Low	1967	1967
29	Small	Business	Germany	Montana	Low	1859	1859
30							

14 COLUMNS, 93 ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED ON 06 DECEMBER 2022

Search ENG IN 13:20 15-12-2022

Untitled - Power Query Editor

**Queries [1]**

Finance Data

Table.AddColumn("Inserted Merged Column", "Country Abbreviations", each if [Country] = "Germany" then "GRE" else if [Country] = "France" then "FRN" else null)

	1.2 Sales	1.2 Discounts	1.2 Sales	1.2 COGS	1.2 Profit	Avg Merged	Country Abbreviations
1	13815	0	13815	9210	4605	Midmarket	GRE
2	30216	0	30216	7554	22662	Channel Partners	null
3	37980	0	37980	18990	18990	Government	FRN
4	18540	0	18540	4535	13905	Channel Partners	GRE
5	37050	0	37050	24700	12350	Midmarket	null
6	333187.5	0	333187.5	319860	13327.5	Enterprise	null
7	287400	0	287400	239500	47900	Small Business	null
8	15022	0	15022	10730	4292	Government	GRE
9	43125	0	43125	41400	2725	Enterprise	null
10	9225	0	9225	6150	3075	Midmarket	null
11	33210	332.1	32877.9	22140	10737.9	Midmarket	null
12	690300	6903	683397	575260	108147	Small Business	null
13	27510	275.1	27234.9	13755	13477.9	Government	FRN
14	12810	128.1	12681.9	9150	3531.9	Government	null
15	749400	7494	741906	624500	117406	Small Business	null
16	82875	828.75	82046.25	79560	2486.25	Enterprise	null
17	13704	274.08	13429.92	3426	10000.92	Channel Partners	null
18	31320	626.4	30695.6	155660	15033.6	Government	null
19	8280	165.6	8114.4	2070	6044.4	Channel Partners	null
20	207500	4150	203350	199200	4150	Enterprise	null
21	13706	411.18	13924.82	9790	3504.82	Government	GRE
22	22812	684.36	22127.64	5703	1642.64	Channel Partners	FRN
23	3808	114.24	5695.76	2720	973.76	Government	FRN
24	628950	18868.5	610081.5	467220	142861.5	Government	GRE
25	160875	4826.25	156048.75	154440	1608.75	Enterprise	FRN
26	213250	6397.5	206852.5	20470	2132.5	Enterprise	GRE
27	30465	1218.6	29246.4	20310	8936.4	Midmarket	null
28	29505	1180.2	28324.8	19670	8654.8	Midmarket	null
29	557700	22308	535392	464750	70642	Small Business	GRE
30							

15 COLUMNS, 93 ROWS Column profiling based on top 1000 rows

PREVIEW DOWNLOADED ON 06 DECEMBER 2022

## 11. Change the data type.

The screenshot shows the Power Query Editor interface with the 'Finance Data' query selected. In the 'Country Abbreviations' column, a context menu is open, and the 'Change Type' option is selected, with 'Currency' checked. The 'APPLIED STEPS' pane shows the step 'Changed Type1' has been applied to this column.

Gross Sales	1.2 Discounts	1.2 Sales	1.2 COGS	1.2 Profit	4.85 Country Abbreviations
13815	0	13815	9210	4605.00	Midmarket
30216	0	30216	7554	22,662.00	Channel Partners
37980	0	37980	18990	18,990.00	Small Business
18540	0	18540	4635	13,905.00	Channel Partners
37050	0	37050	24700	12,350.00	Midmarket
333187.5	0	333187.5	319860	13,927.50	Enterprise
287400	0	287400	239500	47,900.00	Small Business
15022	0	15022	10730	4,292.00	Government
43125	0	43125	41400	1,725.00	Enterprise
9225	0	9225	6150	3,075.00	Midmarket
33210	332.1	32877.9	22140	10,737.90	Midmarket
690300	6903	683397	575250	15,033.80	Government
27510	275.1	27234.9	13755	10,003.92	Channel Partners
12810	128.1	12681.9	9150	1,472.90	Government
749400	7494	741906	624500	1,428.51	Government
82875	828.75	82046.25	79560	1,608.75	Enterprise
13704	274.08	13429.92	3426	2,132.50	Enterprise
31320	626.4	30693.8	15660	8,936.40	Midmarket
8280	165.6	8114.4	2070	8,654.80	Midmarket
207500	4150	203350	199200	6,044.40	Channel Partners
13706	411.18	13294.82	9790	3,504.82	Government
22812	684.36	22127.64	5703	16,424.64	Channel Partners
3808	114.24	3693.76	2720	973.76	Government
628950	18668.5	610081.5	467220	1,428.51	Government
160875	4826.25	156048.75	154440	1,608.75	Enterprise
213250	6397.5	206852.5	204720	2,132.50	Enterprise
30465	1218.6	29246.4	20310	8,936.40	Midmarket
29505	1180.2	28324.8	19670	8,654.80	Midmarket
557700	22308	535392	464750	70,642.00	Small Business

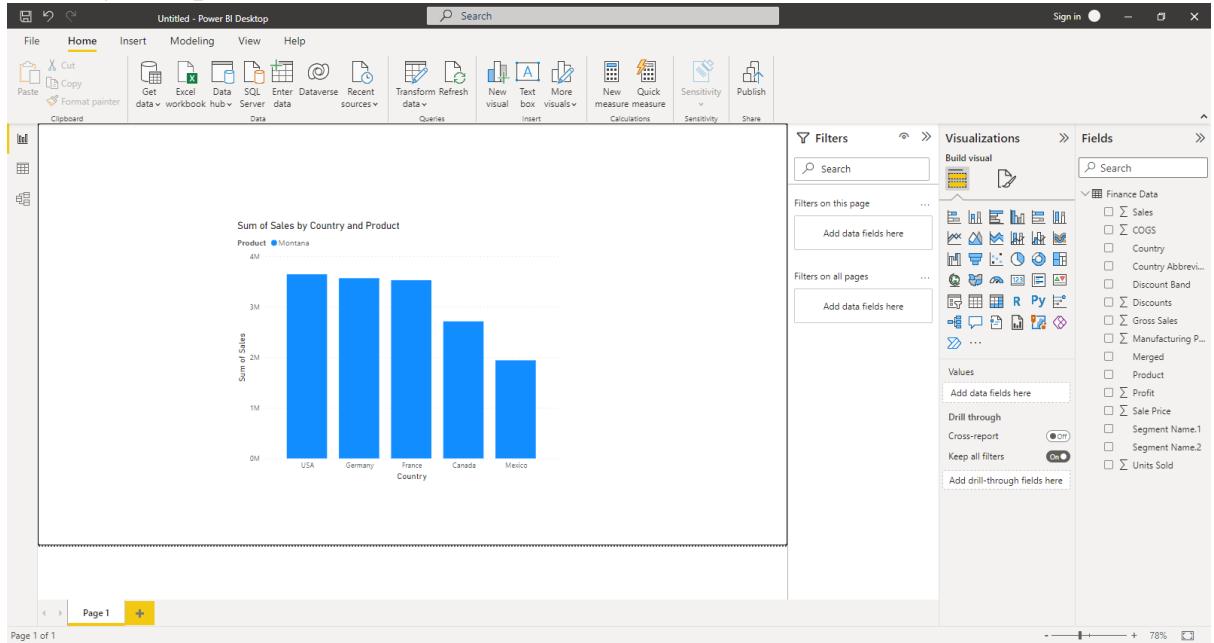
The screenshot shows the Power Query Editor interface with the 'Finance Data' query selected. In the 'Country Abbreviations' column, a context menu is open, and the 'Change Type' option is selected, with 'Currency' checked. The 'APPLIED STEPS' pane shows the step 'Changed Type2' has been applied to this column.

Gross Sales	1.2 Discounts	1.2 Sales	1.2 COGS	1.2 Profit	4.85 Country Abbreviations
13815	0	13815	9210	4605.00	Midmarket
30216	0	30216	7554	22,662.00	Channel Partners
37980	0	37980	18990	18,990.00	Small Business
18540	0	18540	4635	13,905.00	Channel Partners
37050	0	37050	24700	12,350.00	Midmarket
333187.5	0	333187.5	319860	13,927.50	Enterprise
287400	0	287400	239500	47,900.00	Small Business
15022	0	15022	10730	4,292.00	Government
43125	0	43125	41400	1,725.00	Enterprise
9225	0	9225	6150	3,075.00	Midmarket
33210	332.1	32877.9	22140	10,737.90	Midmarket
690300	6903	683397	575250	15,033.80	Government
27510	275.1	27234.9	13755	10,003.92	Channel Partners
12810	128.1	12681.9	9150	1,472.90	Government
749400	7494	741906	624500	1,428.51	Government
82875	828.75	82046.25	79560	2,486.25	Enterprise
13704	274.08	13429.92	3426	10,003.92	Channel Partners
31320	626.4	30693.8	15660	15,033.80	Government
8280	165.6	8114.4	2070	6,044.40	Channel Partners
207500	4150	203350	199200	4,150.00	Enterprise
13706	411.18	13294.82	9790	3,504.82	Government
22812	684.36	22127.64	5703	16,424.64	Channel Partners
3808	114.24	3693.76	2720	973.76	Government
628950	18668.5	610081.5	467220	1,428.51	Government
160875	4826.25	156048.75	154440	1,608.75	Enterprise
213250	6397.5	206852.5	204720	2,132.50	Enterprise
30465	1218.6	29246.4	20310	8,936.40	Midmarket
29505	1180.2	28324.8	19670	8,654.80	Midmarket
557700	22308	535392	464750	70,642.00	Small Business

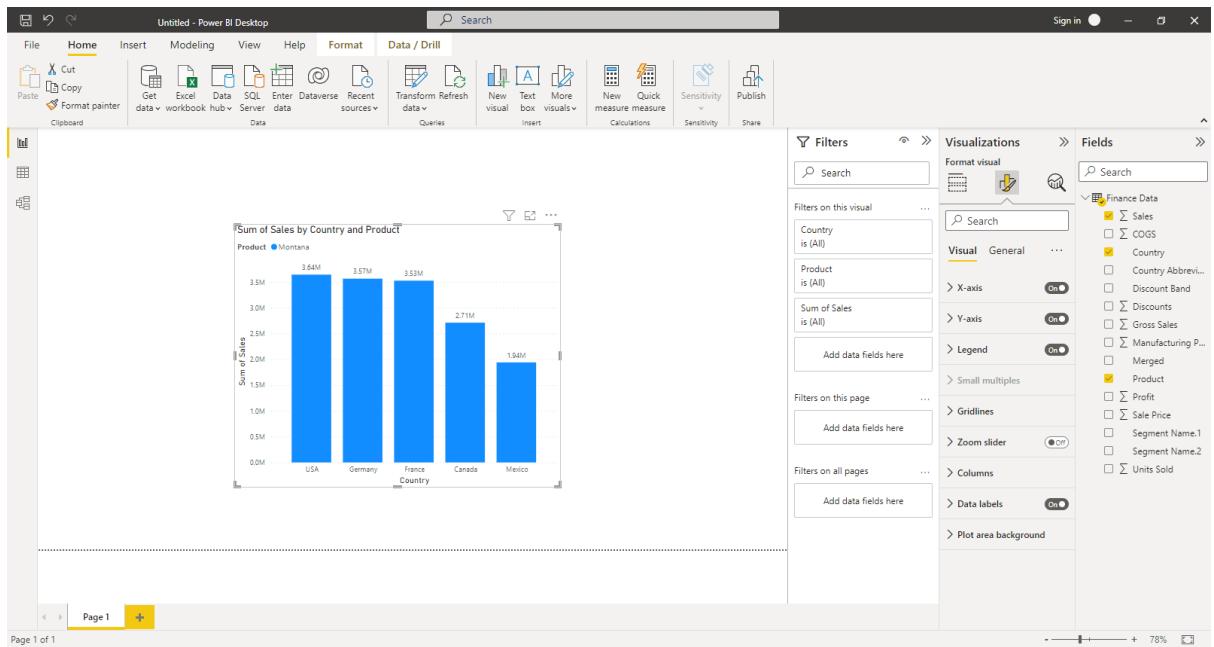
## Practical 4

Aim: Create the data staging area for the selected database (Financial dataset)

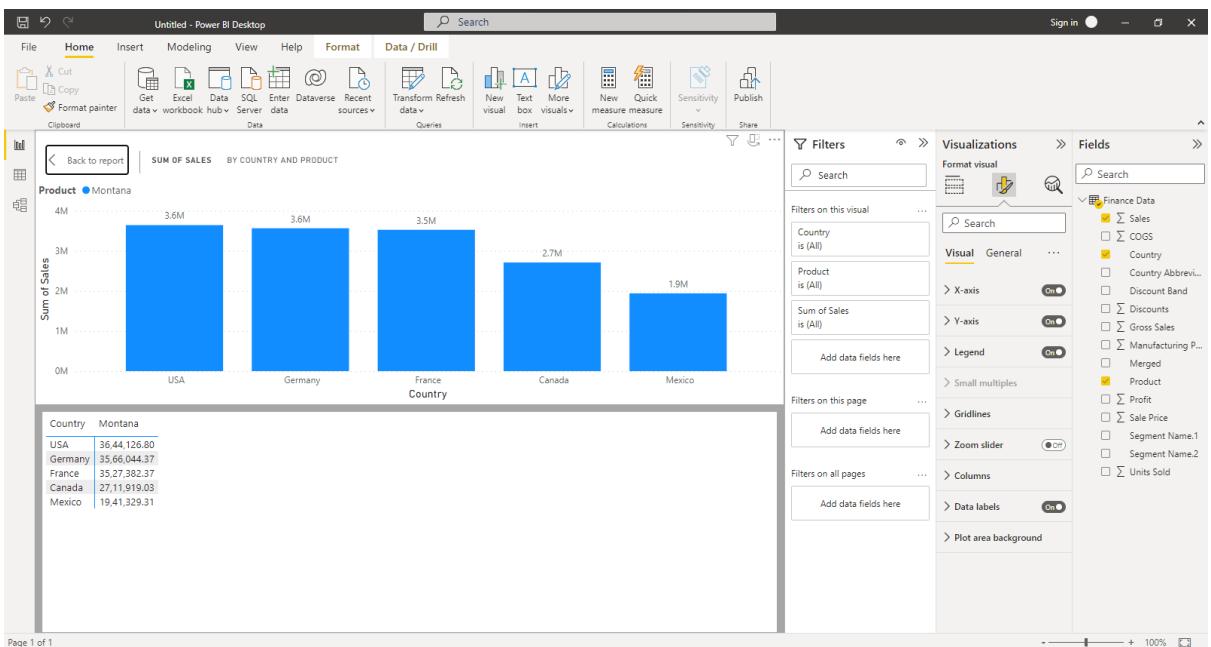
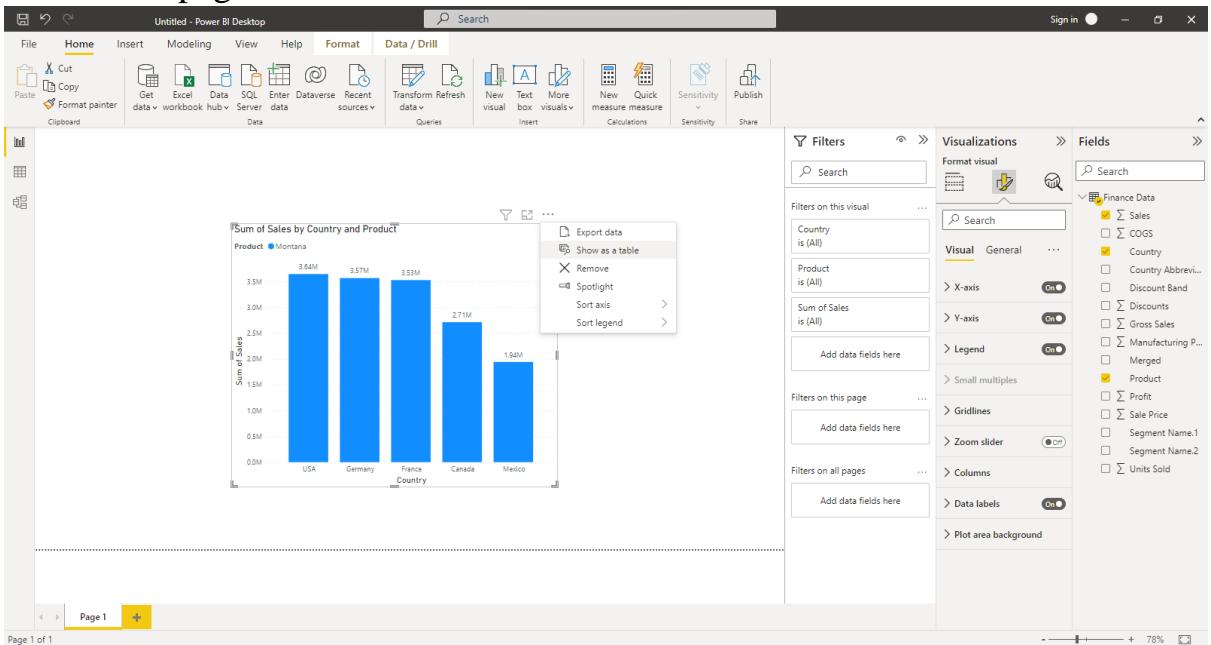
1. Create a clustered column chart for submission of sales according to country and product.

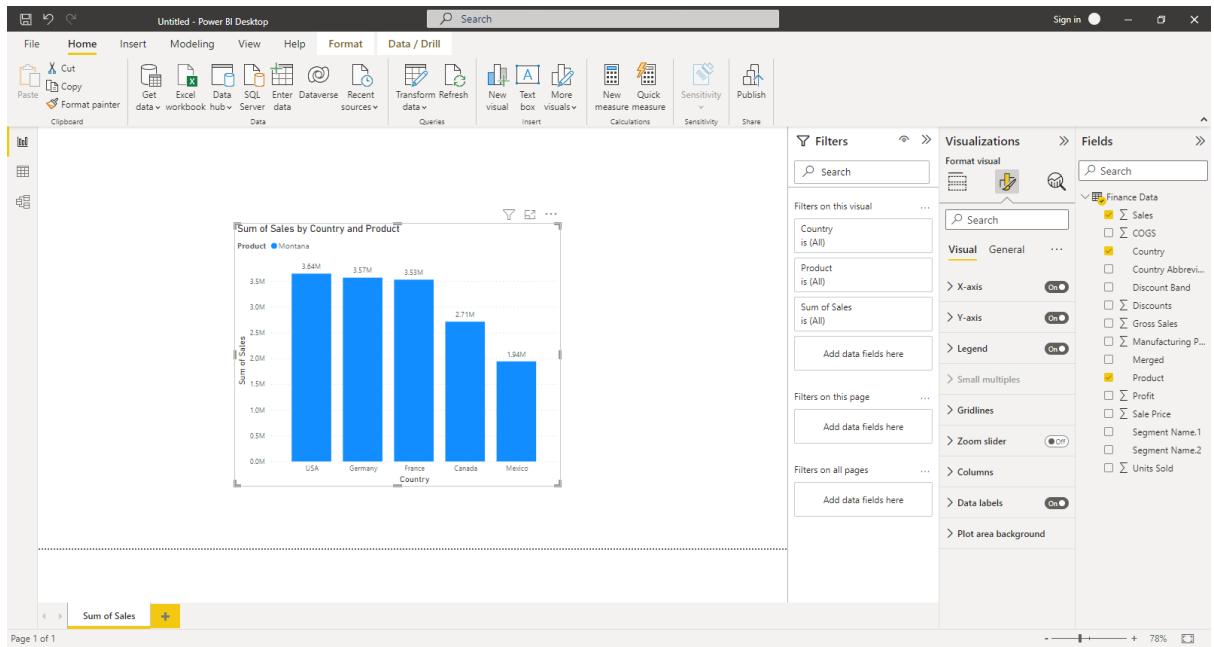


2. For the above chart turn on the data labels.

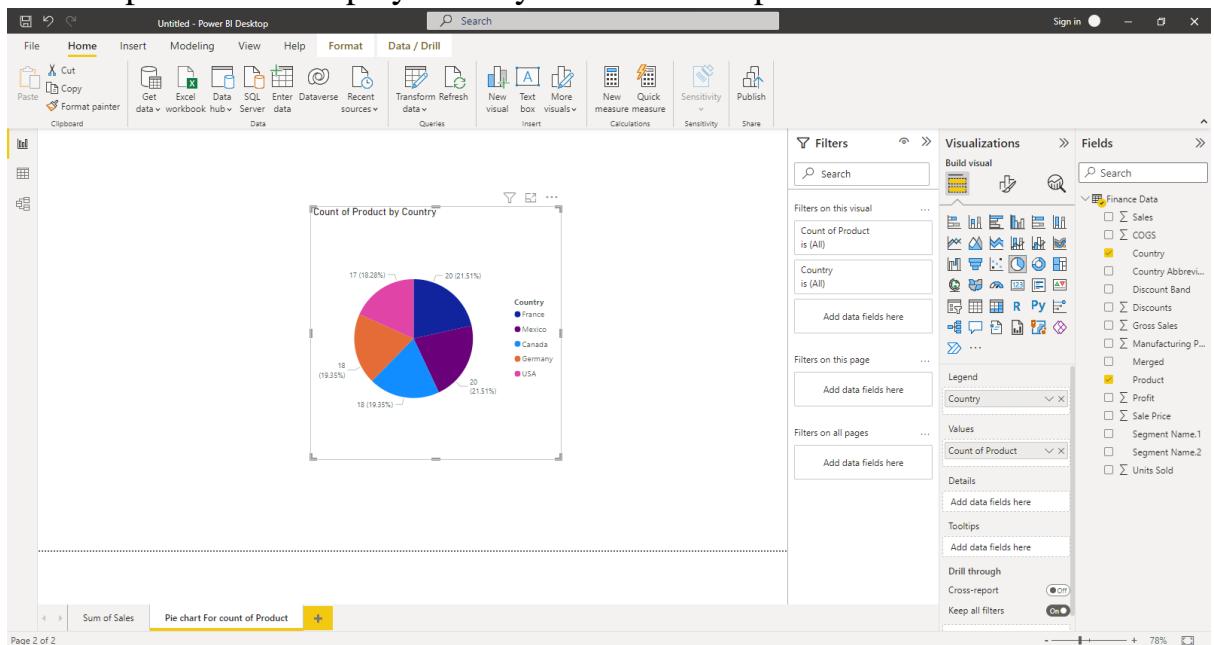


3. Display the table using the option show as a table for the above chart also rename the page

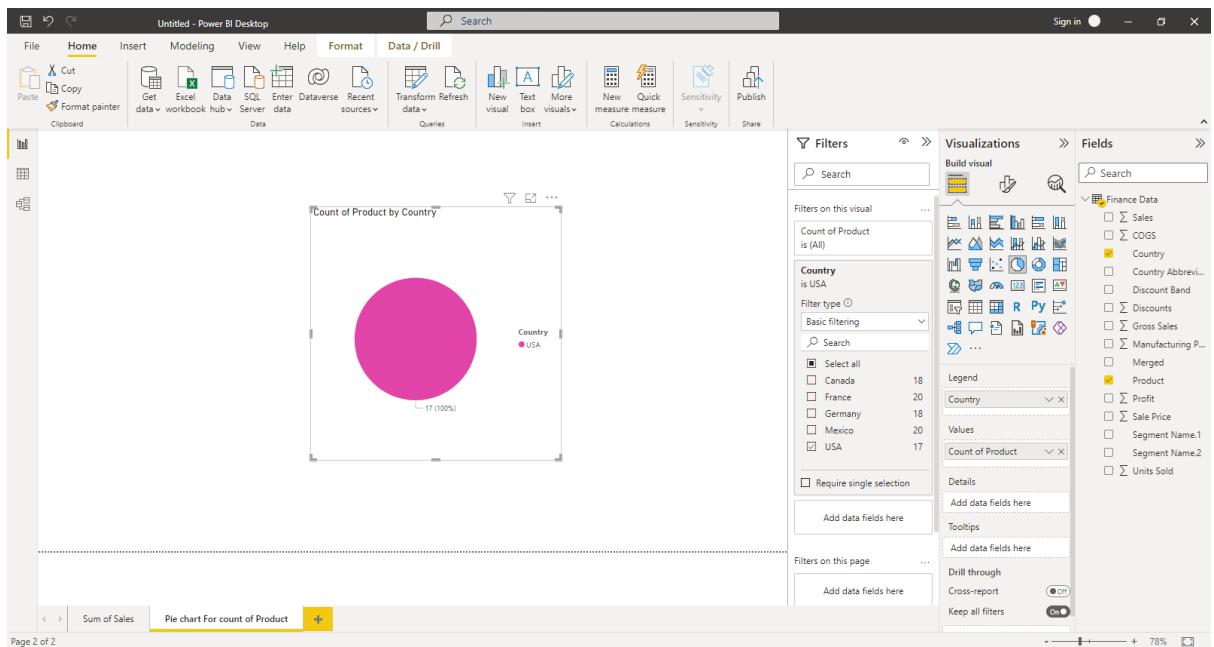




#### 4. Draw a pie chart to display country wise count of product details



5. Using the filters tab for the country field display the data for United States of America



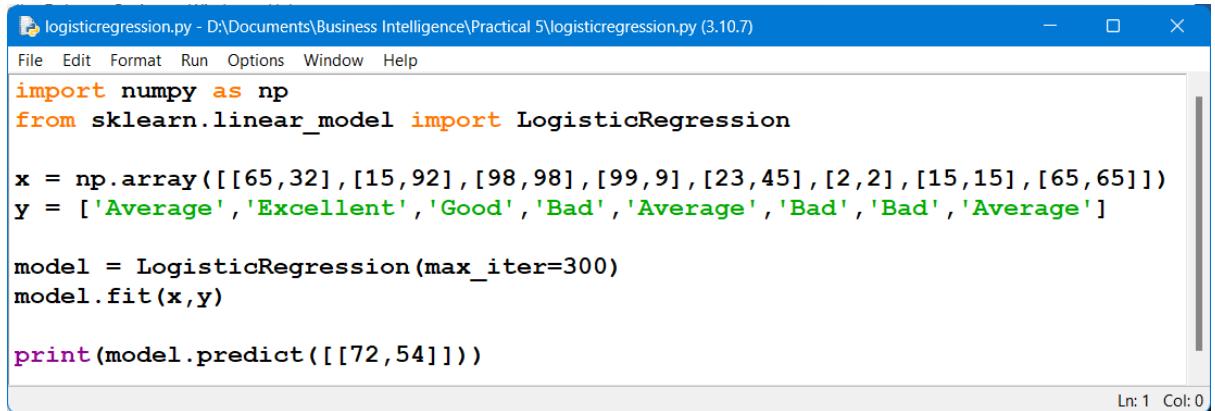
## Practical 5

Aim: Perform the Logistic Regression on the given dataset.

1. Implement the Logistic Regression based on the marks and attendance of the student predict which class or category entity belongs to

Marks	65	15	98	99	23	2	15	65	72
Attendance	32	92	98	9	45	2	15	65	54
Category	Average	Excellent	Good	Bad	Average	Bad	Bad	Average	?

Code:



The screenshot shows a code editor window titled "logisticregression.py - D:\Documents\Business Intelligence\Practical 5\logisticregression.py (3.10.7)". The code is as follows:

```
import numpy as np
from sklearn.linear_model import LogisticRegression

x = np.array([[65,32],[15,92],[98,98],[99,9],[23,45],[2,2],[15,15],[65,65]])
y = ['Average', 'Excellent', 'Good', 'Bad', 'Average', 'Bad', 'Bad', 'Average']

model = LogisticRegression(max_iter=300)
model.fit(x,y)

print(model.predict([[72,54]]))
```

Ln: 1 Col: 0

Output:

```
>>> [ 'Average' ]
```

## 2. Implement the logistic regression

3. Time Spent	20	10	30	4	15	12	10	2
Exam Score	80	40	30	40	45	20	50	15
Result	Pass	Pass	Fail	Pass	Pass	Fail	Pass	?

Code:

```
logisticregression2.py - D:\Documents\Business Intelligence\Practical 5\logisticregression2.py (3.10.7)
File Edit Format Run Options Window Help
import numpy as np
from sklearn.linear_model import LogisticRegression

x = np.array([[20,80],[10,40],[30,30],[4,40],[15,45],[12,20],[10,50]])
y = ['Pass','Pass','Fail','Pass','Pass','Fail','Pass']

model = LogisticRegression(max_iter=300)
model.fit(x,y)

print(model.predict([[2,15]]))
```

Output:

```
>>> | ['Fail']
```

## Practical 6

Aim: Perform the Data classification using classification algorithm.

Code:

The screenshot shows a code editor window titled "knn.py - D:\Documents\Business Intelligence\Practical 6\knn.py (3.10.7)". The code is written in Python and performs K-Nearest Neighbors (KNN) classification. It starts by importing necessary libraries from sklearn: preprocessing and neighbors. It then defines three lists: "weather" (containing weather conditions like Sunny, Overcast, Rainy), "temp" (containing temperatures like Hot, Mild, Cool), and "play" (containing play outcomes like No, Yes). These lists are converted into numerical values using LabelEncoder. The features are combined into a single list of tuples, and a KNeighborsClassifier model is trained with n\_neighbors=3. Finally, the model is used to predict the output for the first two samples in the dataset.

```
*knn.py - D:\Documents\Business Intelligence\Practical 6\knn.py (3.10.7)*
File Edit Format Run Options Window Help
from sklearn import preprocessing
from sklearn.neighbors import KNeighborsClassifier

# First Feature
weather=['Sunny','Sunny','Overcast','Rainy','Rainy','Rainy','Overcast','Sunny','Sunny','Rainy',
         'Sunny','Overcast','Overcast','Rainy']
# Second Feature
temp=['Hot','Hot','Mild','Cool','Cool','Cool','Mild','Mild','Mild','Hot','Mild']
# Label or target variable
play=['No','No','Yes','Yes','No','Yes','No','Yes','Yes','Yes','Yes','Yes','Yes','Yes']

le = preprocessing.LabelEncoder()
# Converting string labels into numbers.
weather_encoded=le.fit_transform(weather)
print(weather_encoded)
temp_encoded=le.fit_transform(temp)
print(temp_encoded)
label=le.fit_transform(play)
print(label)
#combinig features
features=list(zip(weather_encoded,temp_encoded))
model = KNeighborsClassifier(n_neighbors=3)
# Train the model using the training sets
model.fit(features,label)
#Predict Output
predicted= model.predict([[0,2]]) # 0:Overcast, 2:Mild
print(predicted)
```

Output:

```
[2 2 0 1 1 1 0 2 2 1 2 0 0 1]
[1 1 1 2 0 0 0 2 0 2 2 2 1 2]
[0 0 1 1 1 0 1 0 1 1 1 1 1 0]
[1]
```

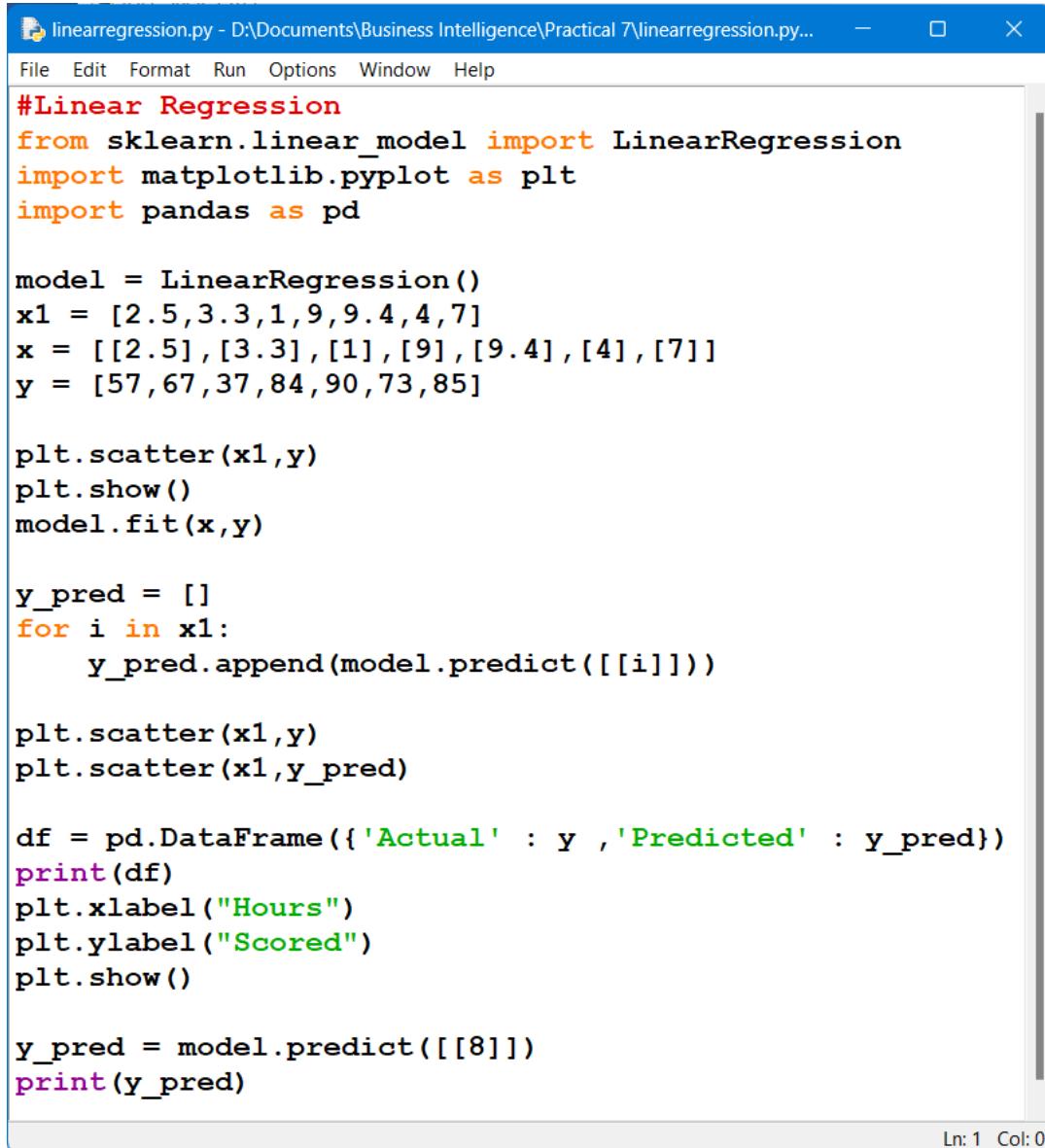
## Practical 7

Aim: Perform Linear Regression on the given data based on the number of hours the student studied and score obtained

Hours	2.5	3.3	1	9	9.4	4	7
Score	57	67	37	84	90	73	85

Predict the score if the student studies for 8 hours.

Code:



The screenshot shows a code editor window with the following Python script for linear regression:

```
#Linear Regression
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
import pandas as pd

model = LinearRegression()
x1 = [2.5,3.3,1,9,9.4,4,7]
x = [[2.5],[3.3],[1],[9],[9.4],[4],[7]]
y = [57,67,37,84,90,73,85]

plt.scatter(x1,y)
plt.show()
model.fit(x,y)

y_pred = []
for i in x1:
    y_pred.append(model.predict([[i]]))

plt.scatter(x1,y)
plt.scatter(x1,y_pred)

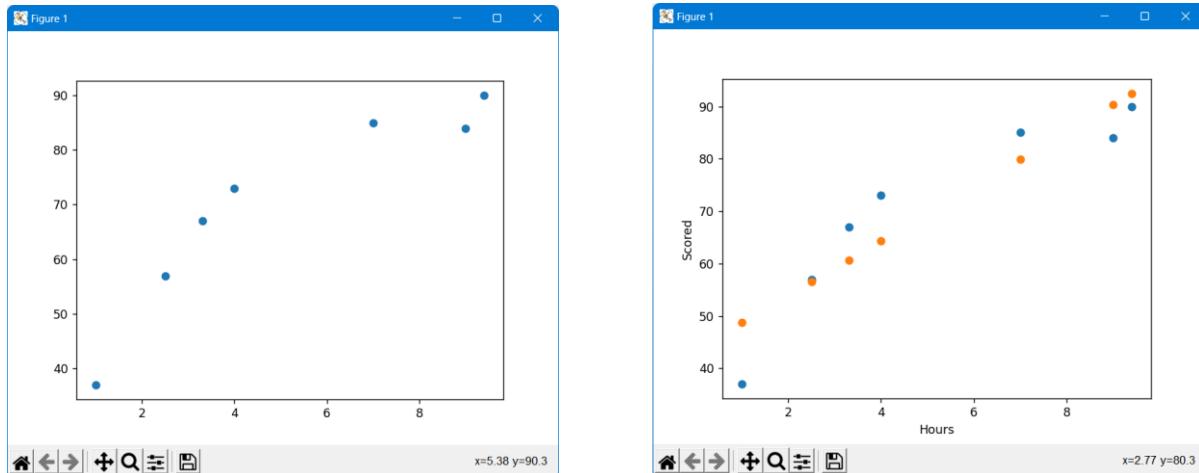
df = pd.DataFrame({'Actual' : y , 'Predicted' : y_pred})
print(df)
plt.xlabel("Hours")
plt.ylabel("Scored")
plt.show()

y_pred = model.predict([[8]])
print(y_pred)
```

The code imports necessary libraries (sklearn, matplotlib, pandas), defines a LinearRegression model, and performs a scatter plot of hours vs score. It then uses the model to predict the score for 8 hours and prints the result.

Ln: 1 Col: 0

Output:



	Actual	Predicted
0	57	[56.514418238305694]
1	67	[60.68122347175426]
2	37	[48.70165842558964]
3	84	[90.36971076007526]
4	90	[92.45311337679954]
5	73	[64.32717805102175]
6	85	[79.95269767645385]

## Practical 8

Aim: Perform the data clustering using KMeans algorithm

Confirm the data clustering using the k means clustering algorithm. Given are store coordinates of 9 stores in a area. Group the stores using k means.

Code:

```

kmeanscluster.py - D:\Documents\Business Intelligence\Practical 8\kmeanscluster.py (3.10.7)
File Edit Format Run Options Window Help
import matplotlib.pyplot as plt
import numpy as np
from sklearn.cluster import KMeans

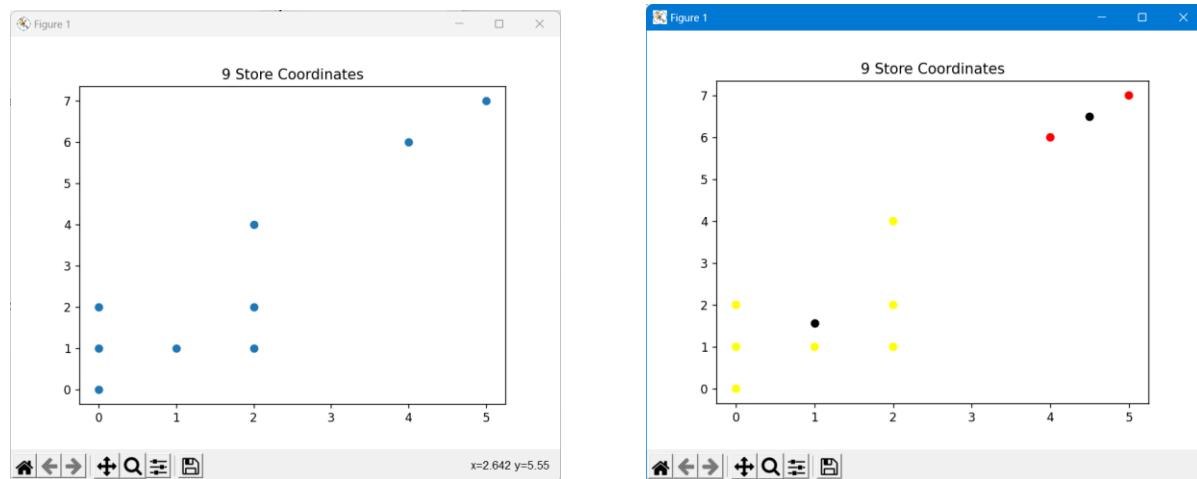
X = np.array([[0,0],[0,1],[0,2],[1,1],[2,1],[2,2],[5,7],[4,6],[2,4]])
plt.title("9 Store Coordinates")
plt.scatter(X[:,0],X[:,1], label="True Position")
plt.show()

kmeans = KMeans(n_clusters=2)
kmeans.fit(X)
print(kmeans.cluster_centers_)
print(kmeans.labels_)

plt.title("9 Store Coordinates")
plt.scatter(X[:,0],X[:,1], c = kmeans.labels_, cmap = 'autumn')
plt.scatter(kmeans.cluster_centers_[:,0] ,kmeans.cluster_centers_[:,1], color="black")
plt.show()

```

Output:



```

[[4.5      6.5      ]
 [1.       1.57142857]
 [1 1 1 1 1 1 0 0 1]
>>>

```

## Practical 9

**Aim:** Using the Sales data in Microsoft excel create pivot table and pivot chart.

Consider the Sales data

	A	B	C	D	E	F
1	Date	Color	Region	Units	Sales	
2	3-Jan-16	Red	West	1	\$11.00	
3	13-Jan-16	Blue	South	8	\$96.00	
4	21-Jan-16	Green	West	2	\$26.00	
5	30-Jan-16	Blue	North	7	\$84.00	
6	7-Feb-16	Green	North	8	\$104.00	
7	13-Feb-16	Red	South	2	\$22.00	
8	21-Feb-16	Blue	East	5	\$60.00	
9	1-Mar-16	Green	West	2	\$26.00	
10	13-Mar-16	Blue	East	8	\$96.00	
11	23-Mar-16	Blue	North	7	\$84.00	
12	28-Mar-16	Green	West	2	\$26.00	
13	3-Apr-16	Blue	South	8	\$96.00	
14	12-Apr-16	Green	South	1	\$13.00	
15	16-Apr-16	Red	East	8	\$88.00	
16	23-Apr-16	Red	West	6	\$66.00	
17	30-Apr-16	Green	South	5	\$65.00	
18	9-May-16	Blue	South	7	\$84.00	
19	#####	Red	South	5	\$55.00	
20	#####	Blue	South	1	\$12.00	
21	#####	Blue	North	4	\$48.00	
22	4-Jun-16	Blue	North	7	\$84.00	
23	13-Jun-16	Red	East	3	\$33.00	
24	21-Jun-16	Blue	South	2	\$24.00	
25	26-Jun-16	Blue	South	6	\$72.00	
26	2-Jul-16	Red	East	6	\$66.00	
27	8-Jul-16	Green	West	2	\$26.00	
28	12-Jul-16	Blue	South	4	\$48.00	
29	19-Jul-16	Green	South	1	\$13.00	
30	26-Jul-16	Blue	West	1	\$12.00	
31	31-Jul-16	Red	South	2	\$22.00	
32	7-Aug-16	Green	West	5	\$65.00	

1. Create a Pivot table in the same sheet as that of the data.

The screenshot shows a Microsoft Excel spreadsheet titled "Pivotable&chart - Excel". The ribbon is visible at the top with tabs like File, Home, Insert, Page Layout, etc. The "Home" tab is selected. The main area contains a PivotTable with data from row 2 to 32. The PivotTable Fields pane on the right lists fields: Date, Color, Region, Units, and Sales. The "Sales" field is currently selected as Row Labels, and the "Sum of Sales" option is chosen under Values. The PivotTable displays the sum of sales for each color across different regions. A "Grand Total" cell in the PivotTable shows the value 26356.

2. Display the color wise total sales of product in \$ format.

Date	Color	Region	Units	Sales
3-Jan-16	Red	West	1	\$11.00
13-Jan-16	Blue	South	8	\$96.00
21-Jan-16	Green	West	2	\$26.00
30-Jan-16	Blue	North	7	\$84.00
6-Feb-16	Green	North	8	\$104.00
7-Feb-16	Red	South	2	\$22.00
8-Feb-16	Blue	East	5	\$60.00
9-Mar-16	Green	West	2	\$26.00
10-Mar-16	Blue	East	8	\$96.00
11-Mar-16	Blue	North	7	\$84.00
12-Mar-16	Green	West	2	\$26.00
13-Apr-16	Blue	South	8	\$96.00
14-Apr-16	Green	South	1	\$13.00
15-Apr-16	Red	East	8	\$88.00
16-Apr-16	Red	West	6	\$66.00
17-Apr-16	Green	South	5	\$65.00
18-May-16	Blue	South	7	\$84.00
19-Jun-16	Red	South	5	\$55.00
20-Jun-16	Blue	South	1	\$12.00
21-Jun-16	Blue	North	4	\$48.00
22-Jun-16	Blue	North	7	\$84.00
23-Jun-16	Red	East	3	\$33.00
24-Jun-16	Blue	South	2	\$24.00
25-Jun-16	Blue	South	6	\$72.00
26-Jul-16	Red	East	6	\$66.00
27-Jul-16	Green	West	2	\$26.00
28-Jul-16	Blue	South	4	\$48.00
29-Jul-16	Green	South	1	\$13.00
30-Jul-16	Blue	West	1	\$12.00
31-Jul-16	Red	South	2	\$22.00

3. Sort the pivot table in ascending order of total sales.

Color	Sum of Sales
Red	\$5,508.00
Green	\$6,414.00
Silver	\$6,970.00
Blue	\$7,464.00
<b>Grand Total</b>	<b>\$26,356.00</b>

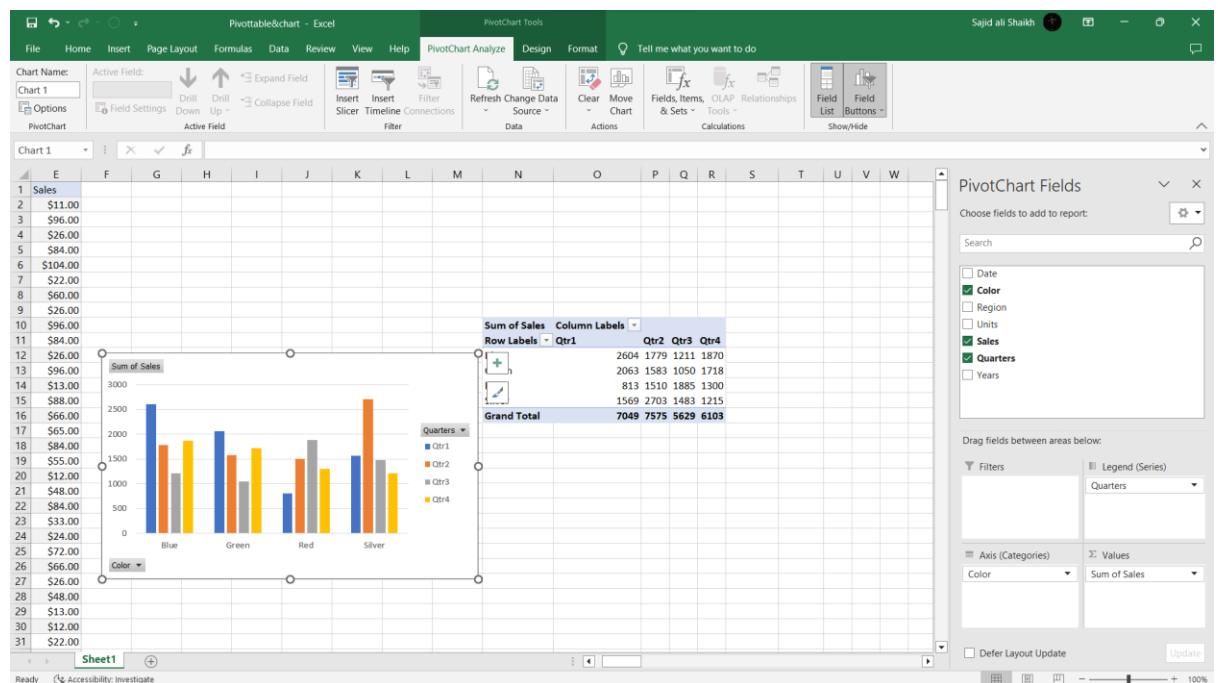
4. Add a column which displays the percentage sales of grand total.

	Sum of Sales	Column Labels			
Row Labels	East	North	South	West	Grand Total
Blue		4.25%	8.95%	6.49%	8.63%
Green		4.55%	6.37%	4.02%	9.41%
Red		2.71%	7.69%	4.27%	6.22%
Silver		2.64%	10.04%	4.36%	9.41%
<b>Grand Total</b>	<b>14.15%</b>	<b>33.04%</b>	<b>19.14%</b>	<b>33.67%</b>	<b>100.00%</b>

5. Group the data of pivot table quarterly.

	Sum of Sales	Column Labels			
Row Labels	Qtr1	Qtr2	Qtr3	Qtr4	
Blue		2604	1779	1211	1870
Green		2063	1583	1050	1718
Red		813	1510	1885	1300
Silver		1569	2703	1483	1215
<b>Grand Total</b>	<b>7049</b>	<b>7575</b>	<b>5629</b>	<b>6103</b>	

6. Draw a pivot chart of any type to display the pivot table data.



## Order dataset

	A	B	C	D	E	F	G
1	Order ID	Product	Category	Amount	Date	Country	
2	1	Carrots	Vegetables	\$4,270	06-01-2016	United States	
3	2	Broccoli	Vegetables	\$8,239	07-01-2016	United Kingdom	
4	3	Banana	Fruit	\$617	08-01-2016	United States	
5	4	Banana	Fruit	\$8,384	10-01-2016	Canada	
6	5	Beans	Vegetables	\$2,626	10-01-2016	Germany	
7	6	Orange	Fruit	\$3,610	11-01-2016	United States	
8	7	Broccoli	Vegetables	\$9,062	11-01-2016	Australia	
9	8	Banana	Fruit	\$6,906	16-01-2016	New Zealand	
10	9	Apple	Fruit	\$2,417	16-01-2016	France	
11	10	Apple	Fruit	\$7,431	16-01-2016	Canada	
12	11	Banana	Fruit	\$8,250	16-01-2016	Germany	
13	12	Broccoli	Vegetables	\$7,012	18-01-2016	United States	
14	13	Carrots	Vegetables	\$1,903	20-01-2016	Germany	
15	14	Broccoli	Vegetables	\$2,824	22-01-2016	Canada	
16	15	Apple	Fruit	\$6,946	24-01-2016	France	
17	16	Banana	Fruit	\$2,320	27-01-2016	United Kingdom	
18	17	Banana	Fruit	\$2,116	28-01-2016	United States	
19	18	Banana	Fruit	\$1,135	30-01-2016	United Kingdom	
20	19	Broccoli	Vegetables	\$3,595	30-01-2016	United Kingdom	
21	20	Apple	Fruit	\$1,161	02-02-2016	United States	
22	21	Orange	Fruit	\$2,256	04-02-2016	France	
23	22	Banana	Fruit	\$1,004	11-02-2016	New Zealand	
24	23	Banana	Fruit	\$3,642	14-02-2016	Canada	
25	24	Banana	Fruit	\$4,582	17-02-2016	United States	
26	25	Beans	Vegetables	\$3,559	17-02-2016	United Kingdom	
27	26	Carrots	Vegetables	\$5,154	17-02-2016	Australia	
28	27	Mango	Fruit	\$7,388	18-02-2016	France	
29	28	Beans	Vegetables	\$7,163	18-02-2016	United States	
30	29	Beans	Vegetables	\$5,101	20-02-2016	Germany	
31	30	Apple	Fruit	\$7,602	21-02-2016	France	

### 1. Create Pivot table in new sheet

The screenshot shows the Microsoft Excel ribbon with the PivotTable Tools tab selected. The PivotTable Fields pane on the right is open, showing fields like Order ID, Product, Category, Amount, Date, and Country. The 'Sum of Amount' field is currently selected and placed in the 'Values' area of the PivotTable Fields pane. The main worksheet area displays the raw data from the 'Order dataset' table, with the first few rows showing Order ID, Product, Category, Amount, Date, and Country.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Order ID	Product	Category	Amount	Date	Country									
2	1	Carrots	Vegetables	\$4,270	06-01-2016	United States									
3	2	Broccoli	Vegetables	\$8,239	07-01-2016	United Kingdom									
4	3	Banana	Fruit	\$617	08-01-2016	United States									
5	4	Banana	Fruit	\$8,384	10-01-2016	Canada									
6	5	Beans	Vegetables	\$2,626	10-01-2016	Germany									
7	6	Orange	Fruit	\$3,610	11-01-2016	United States									
8	7	Broccoli	Vegetables	\$9,062	11-01-2016	Australia									
9	8	Banana	Fruit	\$6,906	16-01-2016	New Zealand									
10	9	Apple	Fruit	\$2,417	16-01-2016	France									
11	10	Apple	Fruit	\$7,431	16-01-2016	Canada									
12	11	Banana	Fruit	\$8,250	16-01-2016	Germany									
13	12	Broccoli	Vegetables	\$7,012	18-01-2016	United States									
14	13	Carrots	Vegetables	\$1,903	20-01-2016	Germany									
15	14	Broccoli	Vegetables	\$2,824	22-01-2016	Canada									
16	15	Apple	Fruit	\$6,946	24-01-2016	France									
17	16	Banana	Fruit	\$2,320	27-01-2016	United Kingdom									
18	17	Banana	Fruit	\$2,116	28-01-2016	United States									
19	18	Banana	Fruit	\$1,135	30-01-2016	United Kingdom									
20	19	Broccoli	Vegetables	\$3,595	30-01-2016	United Kingdom									
21	20	Apple	Fruit	\$1,161	02-02-2016	United States									
22	21	Orange	Fruit	\$2,256	04-02-2016	France									
23	22	Banana	Fruit	\$1,004	11-02-2016	New Zealand									
24	23	Banana	Fruit	\$3,642	14-02-2016	Canada									
25	24	Beans	Vegetables	\$4,582	17-02-2016	United States									
26	25	Carrots	Vegetables	\$3,559	17-02-2016	United Kingdom									
27	26	Mango	Fruit	\$7,388	18-02-2016	France									
28	27	Beans	Vegetables	\$7,163	18-02-2016	United States									
29	28	Beans	Vegetables	\$5,101	20-02-2016	Germany									
30	29	Apple	Fruit	\$7,602	21-02-2016	France									

2. Display the total amount exported of each product, keeping country in filter section

PivotTable Fields

- Order ID
- Product
- Category
- Amount
- Date
- Country

Drag fields between areas below:

Filters: Country

Columns: (empty)

Rows: Product

Values: Sum of Amount

Order ID	Product	Category	Amount	Date	Country
1	Carrots	Vegetables	\$4,270	06-01-2016	United States
2	Broccoli	Vegetables	\$8,239	07-01-2016	United Kingdom
3	3 Banana	Fruit	\$617	08-01-2016	United States
4	4 Banana	Fruit	\$8,384	10-01-2016	Canada
5	5 Beans	Vegetables	\$2,626	10-01-2016	Germany
6	6 Orange	Fruit	\$3,610	11-01-2016	United States
7	7 Broccoli	Vegetables	\$9,062	11-01-2016	Australia
8	8 Banana	Fruit	\$6,906	16-01-2016	New Zealand
9	9 Apple	Fruit	\$2,417	16-01-2016	France
10	10 Apple	Fruit	\$7,431	16-01-2016	Canada
11	11 Banana	Fruit	\$8,250	16-01-2016	Germany
12	12 Broccoli	Vegetables	\$7,012	18-01-2016	United States
13	13 Carrots	Vegetables	\$1,903	20-01-2016	Germany
14	14 Broccoli	Vegetables	\$2,824	22-01-2016	Canada
15	15 Apple	Fruit	\$6,946	24-01-2016	France
16	16 Banana	Fruit	\$2,320	27-01-2016	United Kingdom
17	17 Banana	Fruit	\$2,116	28-01-2016	United States
18	18 Banana	Fruit	\$1,135	30-01-2016	United Kingdom
19	19 Broccoli	Vegetables	\$3,595	30-01-2016	United Kingdom
20	20 Apple	Fruit	\$1,161	02-02-2016	United States
21	21 Orange	Fruit	\$2,256	04-02-2016	France
22	22 Banana	Fruit	\$1,004	11-02-2016	New Zealand
23	23 Banana	Fruit	\$3,642	14-02-2016	Canada
24	24 Banana	Fruit	\$4,582	17-02-2016	United States
25	25 Beans	Vegetables	\$3,559	17-02-2016	United Kingdom
26	26 Carrots	Vegetables	\$5,154	17-02-2016	Australia
27	27 Mango	Fruit	\$7,388	18-02-2016	France
28	28 Beans	Vegetables	\$7,163	18-02-2016	United States
29	29 Beans	Vegetables	\$5,101	20-02-2016	Germany
30	30 Apple	Fruit	\$7,602	21-02-2016	France

3. Sort the pivot table in decreasing order of total sales

PivotTable Fields

- Order ID
- Product
- Category
- Amount
- Date
- Country

Drag fields between areas below:

Filters: Country

Columns: (empty)

Rows: Product

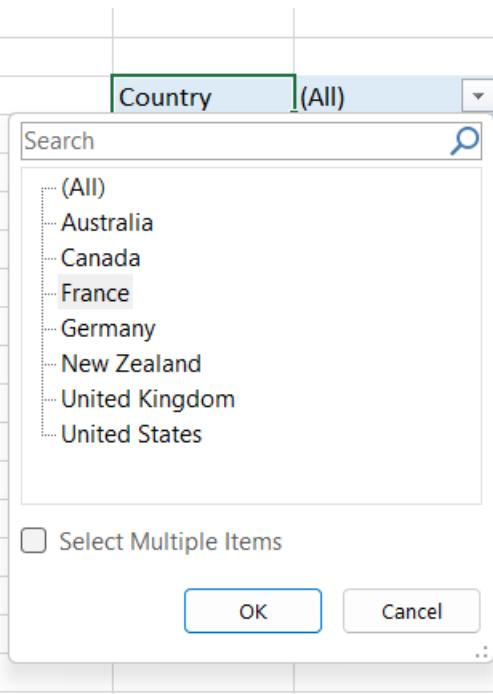
Values: Sum of Amount

Summarize Values By: Min

Order ID	Product	Category	Amount	Date	Country
1	Carrots	Vegetables	\$4,270	06-01-2016	United States
2	Broccoli	Vegetables	\$8,239	07-01-2016	United Kingdom
3	3 Banana	Fruit	\$617	08-01-2016	United States
4	4 Banana	Fruit	\$8,384	10-01-2016	Canada
5	5 Beans	Vegetables	\$2,626	10-01-2016	Germany
6	6 Orange	Fruit	\$3,610	11-01-2016	United States
7	7 Broccoli	Vegetables	\$9,062	11-01-2016	Australia
8	8 Banana	Fruit	\$6,906	16-01-2016	New Zealand
9	9 Apple	Fruit	\$2,417	16-01-2016	France
10	10 Apple	Fruit	\$7,431	16-01-2016	Canada
11	11 Banana	Fruit	\$8,250	16-01-2016	Germany
12	12 Broccoli	Vegetables	\$7,012	18-01-2016	United States
13	13 Carrots	Vegetables	\$1,903	20-01-2016	Germany
14	14 Broccoli	Vegetables	\$2,824	22-01-2016	Canada
15	15 Apple	Fruit	\$6,946	24-01-2016	France
16	16 Banana	Fruit	\$2,320	27-01-2016	United Kingdom
17	17 Banana	Fruit	\$2,116	28-01-2016	United States
18	18 Banana	Fruit	\$1,135	30-01-2016	United Kingdom
19	19 Broccoli	Vegetables	\$3,595	30-01-2016	United Kingdom
20	20 Apple	Fruit	\$1,161	02-02-2016	United States
21	21 Orange	Fruit	\$2,256	04-02-2016	France
22	22 Banana	Fruit	\$1,004	11-02-2016	New Zealand
23	23 Banana	Fruit	\$3,642	14-02-2016	Canada
24	24 Banana	Fruit	\$4,582	17-02-2016	United States
25	25 Beans	Vegetables	\$3,559	17-02-2016	United Kingdom
26	26 Carrots	Vegetables	\$5,154	17-02-2016	Australia
27	27 Mango	Fruit	\$7,388	18-02-2016	France
28	28 Beans	Vegetables	\$7,163	18-02-2016	United States
29	29 Beans	Vegetables	\$5,101	20-02-2016	Germany
30	30 Apple	Fruit	\$7,602	21-02-2016	France

Country	(All)	
Row Labels		Min of Amount
Apple	330	
Banana	107	
Beans	680	
Broccoli	277	
Carrots	339	
Mango	1641	
Orange	220	
<b>Grand Total</b>	<b>107</b>	

4. Find out which product is exported the most to France?



Country	(All)	
Search		
Select Multiple Items		
<input type="checkbox"/>	OK	Cancel
Country	France	
Row Labels		Min of Amount
Apple	1777	
Banana	107	
Beans	680	
Broccoli	5341	
Carrots	9104	
Mango	7388	
Orange	2256	
<b>Grand Total</b>	<b>107</b>	

5. Display the total amount exported to each country, of each product

Sum of Amount	Column Labels	Australia	Canada	France	Germany	New Zealand	United Kingdom	United States	Grand Total
Row Labels									
Apple		20634	24867	80193	9082	10332	17534	28615	191257
Banana		52721	33775	36094	39686	40050	42908	95061	340295
Beans		14433		680	29905		5100	7163	57281
Broccoli		17953	12407	5341	37197	4390	38436	26715	142439
Carrots		8106		9104	21636		41815	56284	136945
Mango		9186	3767	7388	8775		5600	22363	57079
Orange		8680	19929	2256	8887	12010	21744	30932	104438
<b>Grand Total</b>		<b>131713</b>	<b>94745</b>	<b>141056</b>	<b>155168</b>	<b>66782</b>	<b>173137</b>	<b>267133</b>	<b>1029734</b>

6. Create a 3d column chart (pivot chart).

