



SGCP's
Guru Nanak Khalsa College
of Arts, Science & Commerce (Autonomous)



National Facility for Biopharmaceuticals (NFB)

On – Job Training: Training in Data Analytics

Presentation by:

Prarthi Kothari

Roll No. 115

M. Sc. Bioinformatics – Part I



Contents

- Mentors
- Most used Programming Language and tools used for Data Analytics
- Python Programming Language
- SQL (Structured Query Language)
- Advanced Excel
- Power BI



Mentors



Ms. Valencia D'Souza

In - Charge of the
Training in Data
Analytics course.



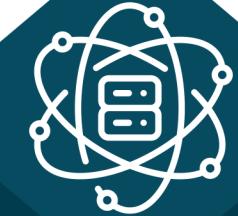
Mr. Mandar Katkar

Data Science
Trainer at
DeveLearn



Ms. Ira Kode

Student
Coordinator of the
Training in Data
Analytics course.



Mrs. Sushma Ghadge

MIS and Graphics
Teacher at
DeveLearn



Ms. Sanvee Khot

Data Science
Trainer at
DeveLearn

Introduction Data Analytics

- involves collecting, processing, and analyzing large datasets to uncover meaningful insights and patterns
- Role of Data Analytics in Scientific Field -
 - a. Analyze experimental data to test hypotheses
 - b. Identify patterns and relationships in complex datasets
 - c. Develop predictive models
- Advantages - enhanced decision making and improved efficiency and reproducibility
- Disadvantages - data quality and integrity, skill gaps, computational resources, ethical considerations
- Applications - Genomics, Neuroscience, Particle Physics
- Future Prospects - Integration with AI and ML



Most Used Programming Language and Tools

1. Python



2. MS-EXCEL



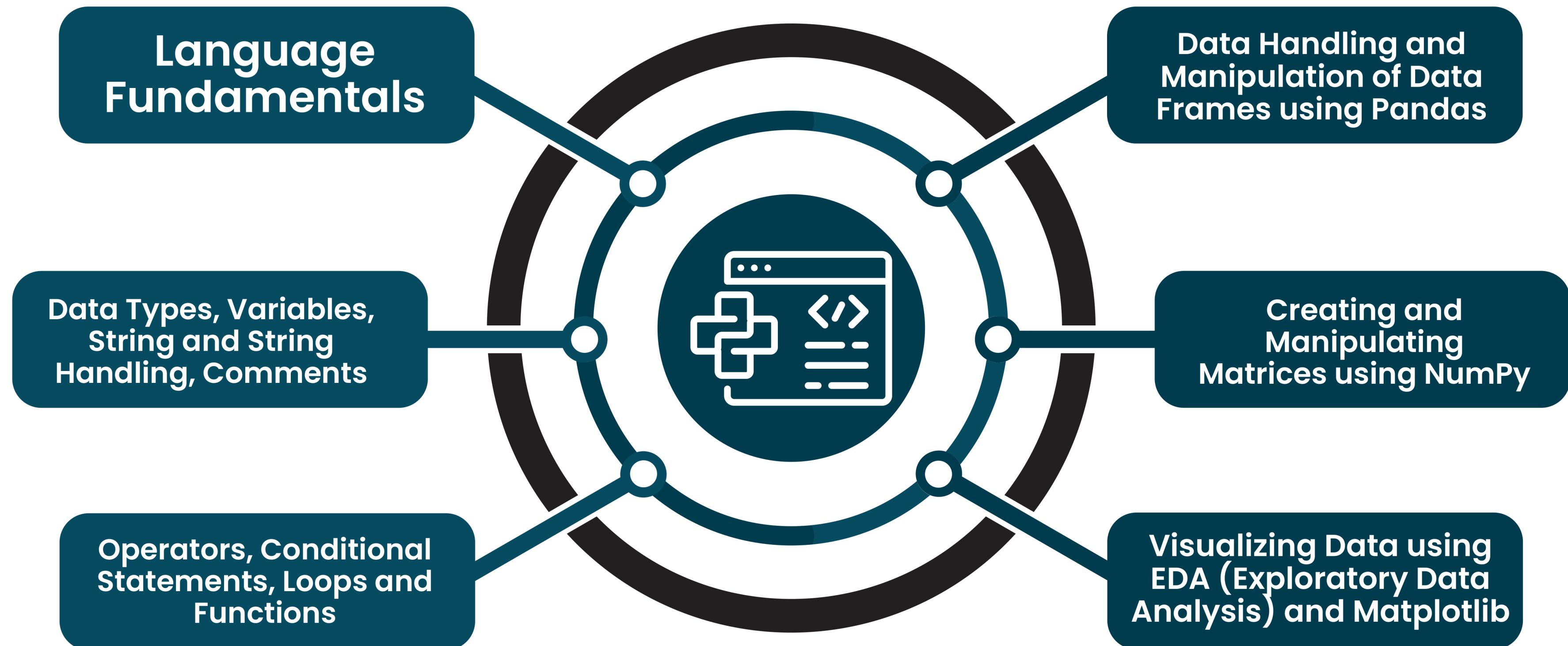
3. MySQL



4. Power BI



Python Programming Language



Language Fundamentals

Property	Description
Interpreted	Code runs line by line without prior compilation.
Object-Oriented	Supports object-based programming concepts like classes and objects.
High-Level	Abstractions hide machine-level details, making code easier to read and write.
Dynamic Semantics	Variable types are determined at runtime, offering flexibility.

Category	Types	Description	Examples
Numeric Types	int (Integers)	Whole numbers	5, 10, -3
	float (Floats)	Numbers with decimals	2.5, -0.1, 3.14
	complex	Numbers with real and imaginary parts	2 + 3j, -4j
Boolean Type	bool	True or False values	True, False
String Type	str (Strings)	Text data in quotes	"Hello, World!", 'Python is cool'
Sequence Types	list	Ordered collections of items in square brackets	[1, 2, 3], ['mango', 'pineapple']
	tuple	Immutable ordered collections in parentheses	(1, 2, 3), ('red', 'green')
	range	Sequence of numbers for loops	range(5) for looping 5 times
Mapping Type	dict (Dictionary)	Key-value pairs in curly braces	{1: 'This', 2: 'is', 3: 'a dictionary'}
Set Type	set	Unordered collections of unique items in curly braces	{1, 2, 'apple'}

Python Data Types



List, Set and Dictionary

```
[ ] # list
# []
l=[1,2,3,4]

[ ] l
1
[1, 2, 3, 4]

[ ] type(l)
list
```

```
[ ] # set.....{ }
s={12,23,43,23434,6,3,23,12,46,3,12,3,1,2}

[ ] s
{1, 2, 3, 6, 12, 23, 43, 46, 23434}

[ ] type(s)
set
```

```
[ ] # dictionary { key:value}

d={'name':'Mandar','marks':70,'roll_no':21}

[ ] d
{'name': 'Mandar', 'marks': 70, 'roll_no': 21}

[ ] type(d)
dict
```



String Handling

1. Concatenation: "Hello " + "World!" => "Hello World!"
2. Repetition: "Python " * 3 => "Python Python Python "
3. Indexing: "Python" => "P"
4. Slicing: "Python"[0:3] => "Pyt"
5. Built-in methods: "python".upper() => "PYTHON", " trim ".strip() => "trim"

```
[ ] q='Mandar Janardan'  
  
[ ] q.upper()  
→ 'MANDAR JANARDAN'  
  
[ ] q.lower()  
→ 'mandar janardan'  
  
[ ] q.title()  
→ 'Mandar Janardan'  
  
[ ] q.capitalize()  
→ 'Mandar janardan'
```

```
[ ] # indexing,slicing  
  
[ ] q  
→ 'Mandar Janardan'  
  
[ ] q[5]  
→ 'r'  
  
[ ] q[12]  
→ 'd'  
  
[ ] q[-3]  
→ 'd'  
  
[ ] s='Automatic'  
  
[ ] s[2:5]  
→ 'tom'
```

```
[ ] # var_name[start:end-1:step]  
  
[ ] s[1:7:2]  
→ 'uo'  
  
[ ] s[::-1]  
→ 'citamotuA'
```



Operators

Arithmetic Operators (+, -, *, /, %, //)

Relational Operators (==, !=, <, >, <=, >=)

Logical Operators (and, or, not)

Membership Operators (in, not in)

```
x = 'Hello world'  
y = {1:'a',2:'b'}  
  
[ ] print('H' in x)  
  
→ True  
  
[ ] print('hello' not in x)  
  
→ True
```

Conditional Statements

1. if: Executes a block of code if a condition is true.
2. if-else: Executes one block if condition is true, else another block.
3. if-elif-else: Checks multiple conditions and executes the corresponding block.
4. Nested if: An if statement inside another if block.

Iterative Statements / Loops

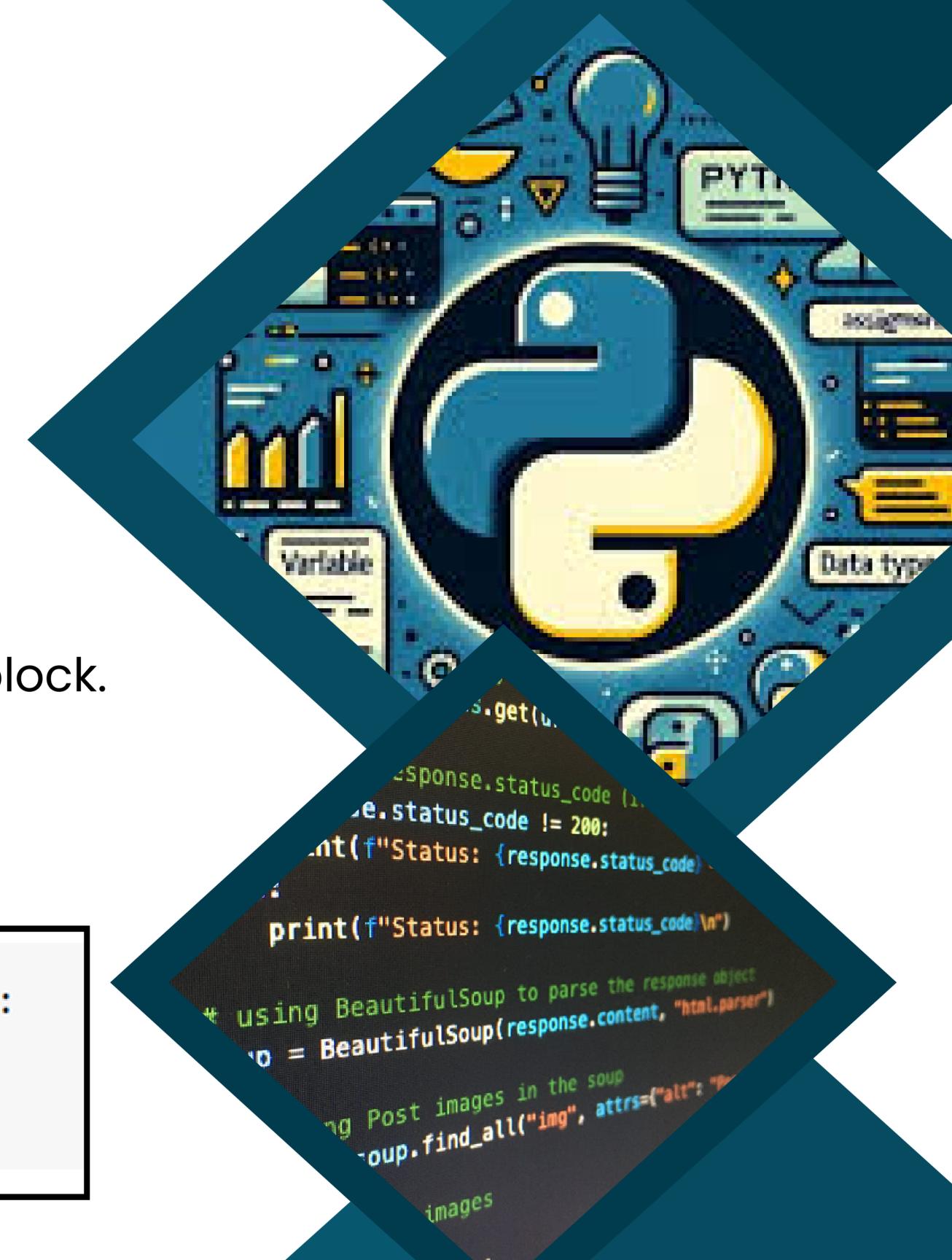
1. for loop
2. while loop

```
▶ for i in range(1,11,1):  
    print(5*i)  
  
→ 5  
10  
15  
20  
25  
30  
35  
40  
45  
50
```

```
▶ a=1  
while a<=10:  
    print(a)  
    a=a+2  
  
→ 1  
3  
5  
7  
9
```

Functions

```
[ ] def add(s,v):  
    c=s+v  
    print(c)
```



Data Handling and Manipulation of Data Frames using Pandas

A popular library for data manipulation and analysis that provides data structures like Series and DataFrame for working with tabular, multidimensional, potentially heterogeneous data.

▼ Data Frames groupby method

Using "group by" method we can:

- Split the data into groups based on some criteria
- Calculate statistics (or apply a function) to each group
- Similar to dplyr() function in R

```
[ ] #Group data using rank  
df_rank = df.groupby('rank')
```

```
[ ] #Calculate mean value for each numeric column per each group  
df_rank['salary'].mean()
```

```
rank  
AssocProf      91786.230769  
AsstProf       81362.789474  
Prof           123347.895833  
Name: salary, dtype: float64
```

```
▶ df.info()  
▶ <class 'pandas.core.frame.DataFrame'>  
RangeIndex: 80 entries, 0 to 79  
Data columns (total 6 columns):  
 #   Column      Non-Null Count  Dtype     
---    
 0   rank        80 non-null    object    
 1   discipline   79 non-null    object    
 2   phd         79 non-null    float64  
 3   service     80 non-null    int64     
 4   sex          80 non-null    object    
 5   salary       80 non-null    int64     
dtypes: float64(1), int64(2), object(3)  
memory usage: 3.9+ KB
```

```
[ ] df.shape
```

```
▶ (80, 6)
```

```
[ ] df.size
```

```
▶ 480
```

```
[ ] #Import Python Libraries  
  
import pandas as pd  
  
▶ df = pd.read_csv('/content/Salaries (1).csv')  
df
```

	rank	discipline	phd	service	sex	salary
0	Prof		B	56.0	49	Male 186960
1	Prof		A	12.0	6	Male 93000
2	Prof		A	23.0	20	Male 110515
3	Prof		A	40.0	31	Male 131205
4	Prof		B	20.0	18	Male 104800

▼ Selecting a column in a Data Frame

Method 1: Subset the data frame using column name: df['phd']

Method 2: Use the column name as an attribute: df.phd

```
[ ] xyz = df[['phd', 'service', 'rank']]
```

```
[ ] a=df[['service', 'rank']]
```

```
▶ a.head()
```

	service	rank
0	49	Prof
1	6	Prof
2	20	Prof
3	31	Prof
4	18	Prof

Creating and Manipulating Matrices using NumPy

A library for scientific computing in Python.

It provides support for large, multi-dimensional arrays and matrices.

It also includes a large collection of high-level mathematical functions to operate on these arrays.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

[ ] a=np.array([1,2,3,45,5])

[ ] a
array([ 1,  2,  3, 45,  5])

[ ] a[3]
45

[ ] a[1:4]
array([ 2,  3, 45])

[ ] type(a)
numpy.ndarray

[ ] q=np.arange(1,50)

[ ] q
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
       18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
       35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49])
```

```
[ ] r=np.arange(27).reshape(3,3,3)

[ ] r
array([[[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8]],

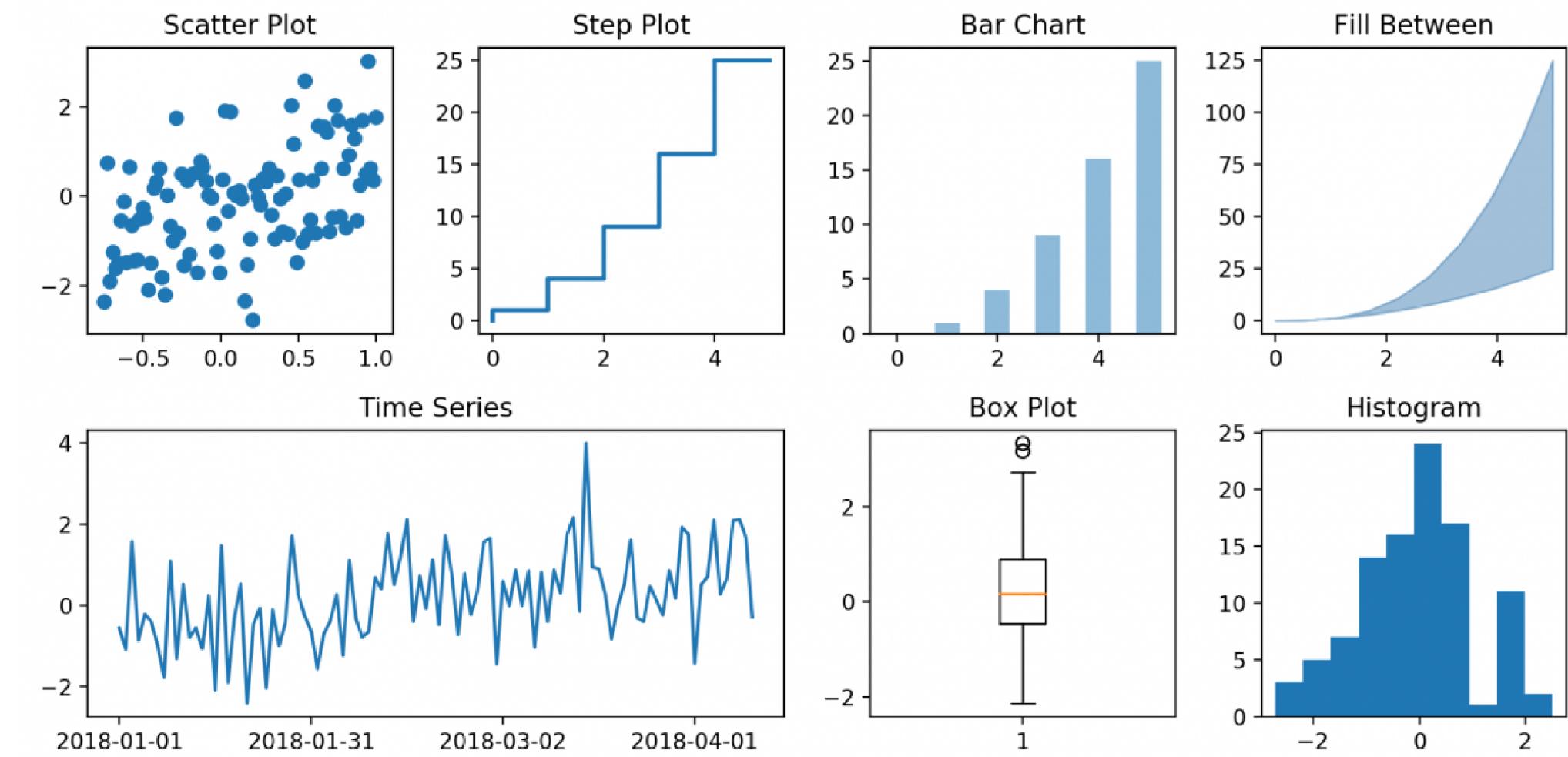
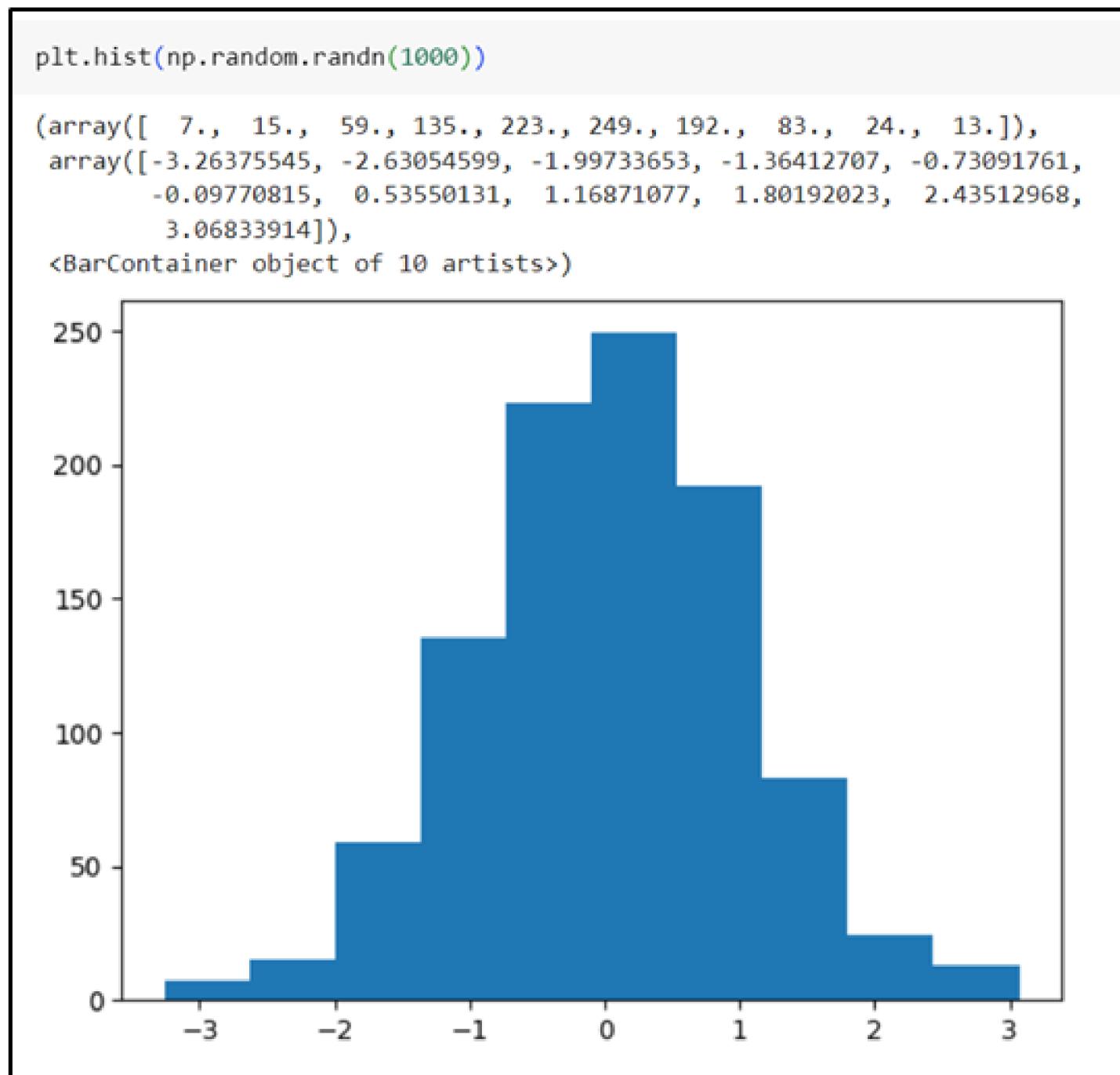
       [[ 9, 10, 11],
       [12, 13, 14],
       [15, 16, 17]],

       [[18, 19, 20],
       [21, 22, 23],
       [24, 25, 26]]])
```

Visualizing Data using EDA (Exploratory Data Analysis) and Matplotlib

A plotting library for Python, producing publication-quality figures in a variety of hardcopy formats and interactive environments.

It is commonly used for exploratory data analysis (EDA) to gain insights from data.



SQL (Structured Query Language)



Examples

```
SELECT DISTINCT department_id FROM  
departments;
```

Query 1 table creation Prarthi_Kothari_SQL_Day2_Assi... x

5 -- Write a SELECT query to retrieve all columns from the table.
6 • SELECT * FROM departments;
7
8 • SELECT * FROM employees;
9
10 -- Write a SELECT query to retrieve only the first name and email columns.
11 • SELECT first_name, email FROM employees;
12
13 -- Write a query to retrieve unique department IDs from the table.
14 • SELECT DISTINCT department_id FROM departments;
15

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content | Result Grid Form Editor Field Types

department_id
10
20
30
40
50
60
70
80
90
100
110
120
130

departments 6 x

```
SELECT * FROM employees WHERE salary BETWEEN  
4000 AND 6000;
```

36 -- Retrieve details of employees with a salary between \$4000 and \$6000.

37 • SELECT * FROM employees

38 WHERE salary BETWEEN 4000 AND 6000;

39

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	employee_id	first_name	last_name	email	phone_number	hire_date	salary	commission_pct	manager_id	department_id
▶	104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-1991	6000	NULL	103	60
	105	David	Austin	DAUSTIN	590.423.4569	25-JUN-1997	4800	NULL	103	60
	106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-1998	4800	NULL	103	60
	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-1999	4200	NULL	103	60
	124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-NOV-1999	5800	NULL	100	50
	184	Nandita	Sarchand	NSARCHAN	650.509.1876	27-JAN-1996	4200	NULL	121	50
	185	Alexis	Bull	ABULL	650.509.2876	20-FEB-1997	4100	NULL	121	50
	192	Sarah	Bell	SBELL	650.501.1876	04-FEB-1996	4000	NULL	123	50
	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-1987	4400	NULL	101	10
	202	Pat	Fay	PFAY	603.123.6666	17-AUG-1997	6000	NULL	201	20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

employees 15 ×

Apply Revert

Examples

```
SELECT * FROM employees WHERE department_id IN (10, 20, 30);
```

```
5  -- Retrieve details of employees whose department IDs are 10, 20, or 30.
6 •   SELECT * FROM employees
7 WHERE department_id IN (10, 20, 30);
```

	employee_id	first_name	last_name	email	phone_number	hire_date	salary	commission_pct	manager_id	department_id
▶	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-1994	11000	NULL	100	30
	115	Alexander	Khoo	AKHOO	515.127.4562	18-MAY-1995	3100	NULL	114	30
	116	Shelli	Baida	SBAIDA	515.127.4563	24-DEC-1997	2900	NULL	114	30
	117	Sigal	Tobias	STOBIAS	515.127.4564	24-JUL-1997	2800	NULL	114	30
	118	Guy	Himuro	GHIMURO	515.127.4565	15-NOV-1998	2600	NULL	114	30
	119	Karen	Colmenares	KCOLMENA	515.127.4566	10-AUG-1999	2500	NULL	114	30
	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-1987	4400	NULL	101	10
	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-1996	13000	NULL	100	20
	202	Pat	Fay	PFAY	603.123.6666	17-AUG-1997	6000	NULL	201	20
*	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE	NONE

employees 4 × Apply Revert

Examples

```
SELECT department_id, COUNT(*) FROM employees  
GROUP BY department_id  
ORDER BY department_id ASC;
```

The screenshot shows a database query interface with the following details:

- Query Text:**

```
13      -- Group employees by department ID and display the count of employees in each department.  
14 •  SELECT department_id, COUNT(*) FROM employees  
15      GROUP BY department_id  
16      ORDER BY department_id ASC;
```
- Result Grid:** A table showing the count of employees per department. The columns are `department_id` and `COUNT(*)`. The data is as follows:

department_id	COUNT(*)
NULL	1
10	1
20	2
30	6
40	1
50	45
60	5
70	1
80	34
90	3
100	6
110	2

- Toolbar:** Includes buttons for **Result Grid**, **Form Editor**, and **Field Types**.
- Status Bar:** Shows "Result 7" and "Read Only".

Examples

```
UPDATE employees  
SET salary = 6000  
WHERE employee_id = 100;
```

```
UPDATE employees  
SET salary = 6000  
WHERE employee_id = 101;
```

```
SELECT * FROM employees;
```

```
5   -- Update the salary of any 2 employees to $6000.  
6 • UPDATE employees  
7   SET salary = 6000  
8   WHERE employee_id = 100;  
9  
10 • UPDATE employees  
11  SET salary = 6000  
12  WHERE employee_id = 101;  
13  
14 • SELECT * FROM employees;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Apply | Revert

employee_id	first_name	last_name	email	phone_number	hire_date	salary	commission_pct	manager_id	department_id
100	Steven	King	SKING	515.123.4567	17/JUN/1987	6000	NULL	NULL	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-1989	6000	NULL	100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-1993	17000	NULL	100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	03-JAN-1990	9000	NULL	102	60
104	Bruce	Ernst	BERNST	590.423.4568	21-MAY-1991	6000	NULL	103	60
105	David	Austin	DAUSTIN	590.423.4569	25-JUN-1997	4800	NULL	103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	05-FEB-1998	4800	NULL	103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	07-FEB-1999	4200	NULL	103	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-AUG-1994	12000	NULL	101	100
109	Daniel	Faviet	DFAVIET	515.124.4169	16-AUG-1994	9000	NULL	108	100
110	John	Chen	JCHEN	515.124.4269	28-SEP-1997	8200	NULL	108	100
111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-SEP-1997	7700	NULL	108	100
112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-MAR-1998	7800	NULL	108	100

employees 5 x

Examples

-- department_id of 'Sales' department = 80

```
SELECT * FROM employees WHERE department_id = 80;
```

```
UPDATE employees SET salary = salary + (salary * 0.10) WHERE department_id = 80;
```

employee_id	first_name	last_name	email	phone_number	hire_date	salary	manager_id	department_id	date_of_birth
145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-1996	14000	100	80	NULL
146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-1997	13500	100	80	NULL
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-MAR-1997	12000	100	80	NULL
148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	15-OCT-1999	11000	100	80	NULL
149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-2000	10500	100	80	NULL
150	Peter	Tucker	PTUCKER	011.44.1344.129268	30-JAN-1997	10000	145	80	NULL
151	David	Bernstein	DBERNSTE	011.44.1344.345268	24-MAR-1997	9500	145	80	NULL
152	Peter	Hall	PHALL	011.44.1344.478968	20-AUG-1997	9000	145	80	NULL
153	Christopher	Olsen	COLSEN	011.44.1344.498718	30-MAR-1998	8000	145	80	NULL
154	Nanette	Cambrault	NCAMBRAU	011.44.1344.987668	09-DEC-1998	7500	145	80	NULL
155	Oliver	Tuvault	OTUVault	011.44.1344.486508	23-NOV-1999	7000	145	80	NULL
156	Janette	King	JKING	011.44.1345.429268	30-JAN-1996	10000	146	80	NULL
157	Patrick	Sully	PSULLY	011.44.1345.929268	04-MAR-1996	9500	146	80	NULL

	employee_id	first_name	last_name	email	phone_number	hire_date	salary	manager_id	department_id	date_of_birth
▶	145	John	Russell	JRUSSEL	011.44.1344.429268	01-OCT-1996	15400	100	80	NULL
	146	Karen	Partners	KPARTNER	011.44.1344.467268	05-JAN-1997	14850	100	80	NULL
	147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	10-MAR-1997	13200	100	80	NULL
	148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	15-OCT-1999	12100	100	80	NULL
	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-2000	11550	100	80	NULL
	150	Peter	Tucker	PTUCKER	011.44.1344.129268	30-JAN-1997	11000	145	80	NULL
	151	David	Bernstein	DBERNSTE	011.44.1344.345268	24-MAR-1997	10450	145	80	NULL
	152	Peter	Hall	PHALL	011.44.1344.478968	20-AUG-1997	9900	145	80	NULL
	153	Christopher	Olsen	COLSEN	011.44.1344.498718	30-MAR-1998	8800	145	80	NULL
	154	Nanette	Cambrault	NCAMBRAU	011.44.1344.987668	09-DEC-1998	8250	145	80	NULL
	155	Oliver	Tuvault	OTUVault	011.44.1344.486508	23-NOV-1999	7700	145	80	NULL
	156	Janette	King	JKING	011.44.1345.429268	30-JAN-1996	11000	146	80	NULL
	157	Patrick	Sully	PSULLY	011.44.1345.929268	04-MAR-1996	10450	146	80	NULL

Advanced Excel



1. Adding Numerical Values

Fruit	Amount
Apples	50
Oranges	20
Bananas	60
Lemons	40

=SUM(D4:D7)

2. Filling and Autofill

This:	Plus this:	Equals:	Plus this:	Equals:
50	50	100	75	175
50	60	110	75	185
50	70	120	75	195
50	80	130	75	205

This:	Plus this:	Equals:	Plus this:	Equals:
50	50	100	75	175
50	60	110	75	185
50	70	120	75	195
50	80	130	75	205

3. Transposing Data

Item	Bread	Donuts	Cookies	Cakes	Pies
Amount	50	100	40	50	20

Item	Amount
Bread	50
Donuts	100
Cookies	40
Cakes	50
Pies	20

(Ctrl)

4. Data Splitting

Email	First name	Last name
Nancy.Smith@contoso.com		Smith
Andy.North@fabrikam.com		
Jan.Kotas@relecloud.com		
Mariya.Jones@contoso.com		
Yvonne.McKay@fabrikam.com		

Email	First name	Last name
Nancy.Smith@contoso.com	Nancy	Smith
Andy.North@fabrikam.com	Andy	North
Jan.Kotas@relecloud.com	Jan	Kotas
Mariya.Jones@contoso.com	Mariya	Jones
Yvonne.McKay@fabrikam.com	Yvonne	McKay

5. Sorting and Filtering Data

Department	Category	Oct	Nov	Dec
Meat	Beef	\$90,000	\$1,10,000	\$1,20,000
Bakery	Desserts	\$25,000	\$80,000	\$1,20,000
Produce	Fruit	\$10,000	\$30,000	\$40,000
Produce	Veggies	\$30,000	\$80,000	\$30,000
Deli	Salads	\$90,000	\$35,000	\$25,000
Meat	Chicken	\$75,000	\$82,000	\$20,00,000
Bakery	Breads	\$30,000	\$15,000	\$20,000
Deli	Sandwiches	\$80,000	\$40,000	\$20,000

Department	Category	Oct	Nov	Dec
Bakery	Desserts	\$25,000	\$80,000	\$1,20,000
Bakery	Breads	\$30,000	\$15,000	\$20,000
Deli	Salads	\$90,000	\$35,000	\$25,000
Deli	Sandwiches	\$80,000	\$40,000	\$20,000
Meat	Beef	\$90,000	\$1,10,000	\$1,20,000
Meat	Chicken	\$75,000	\$82,000	\$20,00,000
Produce	Fruit	\$10,000	\$30,000	\$40,000
Produce	Veggies	\$30,000	\$80,000	\$30,000

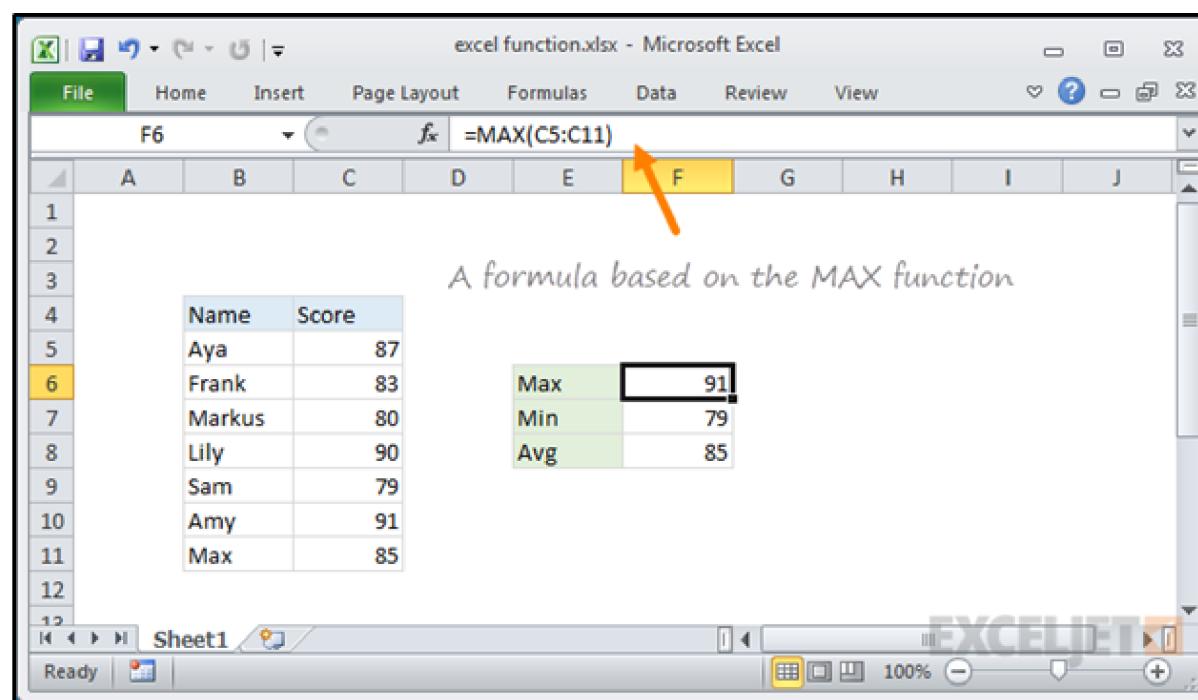


6. Drop Down Lists

Food	Department
Apples	
Beef	
Bananas	
Lemons	
Broccoli	
Kale	
Ham	
Bread	
Chicken	
Cookies	
Cakes	
Pies	

Food	Department
Apples	Produce
Beef	Meat
Bananas	Produce
Lemons	
Broccoli	Produce
Kale	Meat
Ham	
Bread	
Chicken	
Cookies	
Cakes	
Pies	

8. Functions



7. Pivot Tables

Pivot Table 1

Sales	Sep	Oct	Nov	Total
Apples	250	590	840	
John	180	180	180	
Mike	120	120	120	
Pete	290	290	290	
Sally	250	250	250	
Bananas	430	600	1030	
John	400	400	400	
Mike	200	200	200	
Pete	180	180	180	
Sally	250	250	250	
Cherries	580	910	1490	
John	250	250	250	
Mike	250	330	580	
Pete	330	330	330	
Sally	330	330	330	
Oranges	120	720	840	
John	120	120	120	
Mike	400	400	400	
Pete	120	120	120	
Sally	200	200	200	
Total	830	2050	1320	4200

Pivot Table 2

Month	(All)				
Sales	Product				
Reseller	Apples	Bananas	Cherries	Oranges	Total
John	\$180	\$400	\$250	\$120	\$950
Mike	\$120	\$200	\$580	\$400	\$1,300
Pete	\$290	\$180	\$330	\$120	\$920
Sally	\$250	\$250	\$330	\$200	\$1,030
Total	\$840	\$1,030	\$1,490	\$840	\$4,200

Pivot Table 3

Product	(All)				
Sales	Month				
Reseller	Sep	Oct	Nov	Total	
John			\$430	\$520	\$950
Mike		\$250	\$450	\$600	\$1,300
Pete			\$920		\$920
Sally		\$580	\$250	\$200	\$1,030
Total	\$830	\$2,050	\$1,320	\$4,200	

9. Conditional Formatting

Excel Conditional Formatting

Month	Price
Jan	\$120.00
Feb	\$60.00
Mar	\$80.00
Apr	\$102.49
May	\$70.00
Jun	\$99.12
Jul	\$54.00
Aug	\$105.79
Sep	\$45.00
Oct	\$70.00
Nov	\$97.52
Dec	\$105.94

Month	Price
Jan	\$120.00
Feb	\$60.00
Mar	\$80.00
Apr	\$102.49
May	\$70.00
Jun	\$99.12
Jul	\$54.00
Aug	\$105.79
Sep	\$45.00
Oct	\$70.00
Nov	\$97.52
Dec	\$105.94

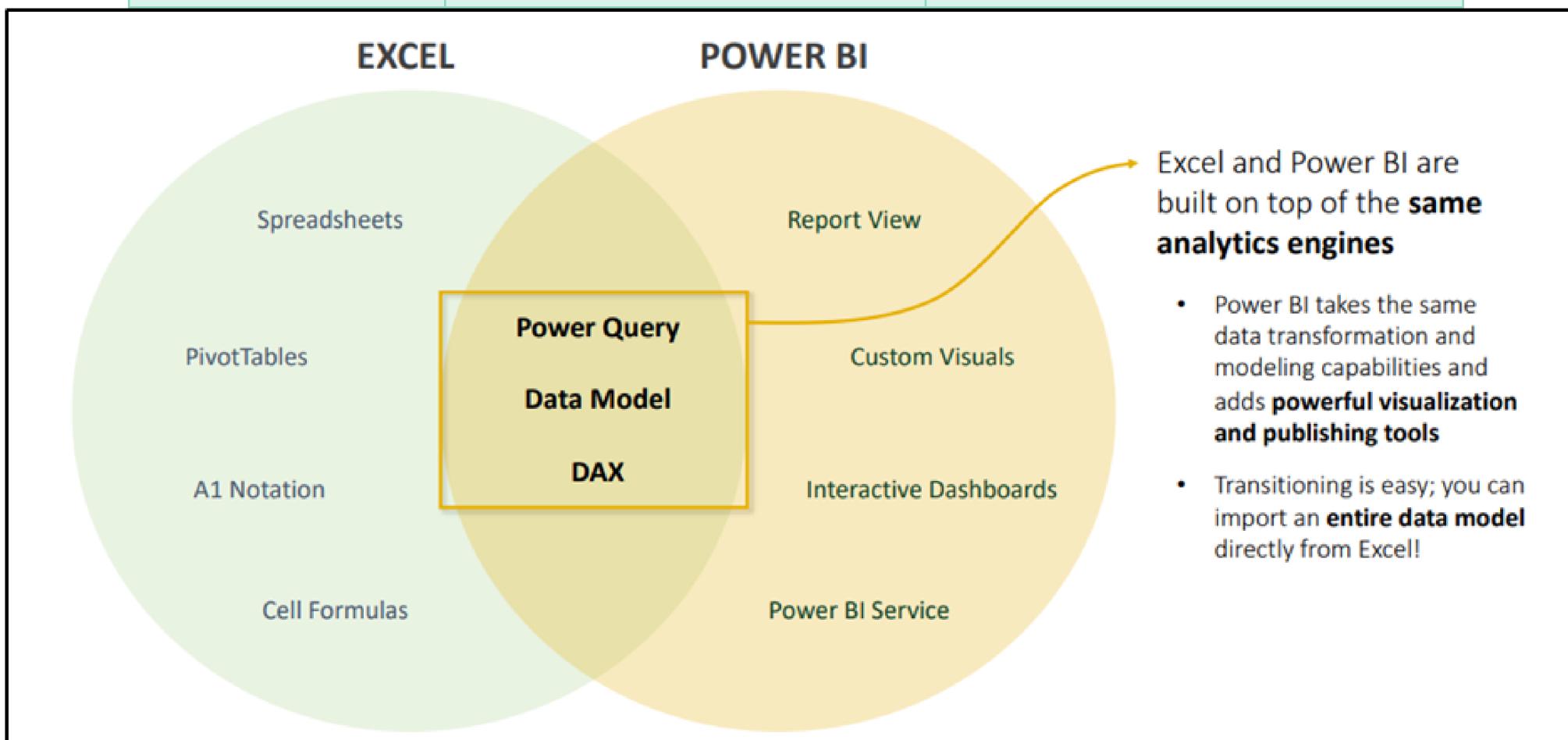
Month	Price
Jan	★ \$120.00
Feb	☆ \$60.00
Mar	★ \$80.00
Apr	★ \$102.49
May	★ \$70.00
Jun	★ \$99.12
Jul	☆ \$54.00
Aug	★ \$105.79
Sep	☆ \$45.00
Oct	★ \$70.00
Nov	★ \$97.52
Dec	★ \$105.94

Power BI



How is Power BI different from Excel ?

Feature	Excel	Power BI
Data Handling	Better for smaller datasets	Handles large datasets efficiently
	May struggle with performance for large data	Efficiently handles millions of rows
Data Visualization	Static charts and graphs	Dynamic, interactive visualizations and dashboards
	Limited set of visualization options	Wide range of built-in and custom visuals
Collaboration & Sharing	Sharing via email or online tools	Cloud-based sharing of reports and dashboards
	More challenging for collaboration	Easier for team collaboration
Data Modeling	Basic data modeling features	Advanced data modeling capabilities
	Less scalable for complex datasets	Manages complex, interconnected datasets
Formulas & Functions	Rich library of built-in functions	DAX formulas and measures



Learning about the Dataset and Handling the Dataset in Power BI

Prarthi Kothari_Power BI Assignment

File Home Help Table tools

Name Customers

Mark as date table Manage relationships New measure Relationships Quick New measure column New table Calculations

Structure Calendars

DAX

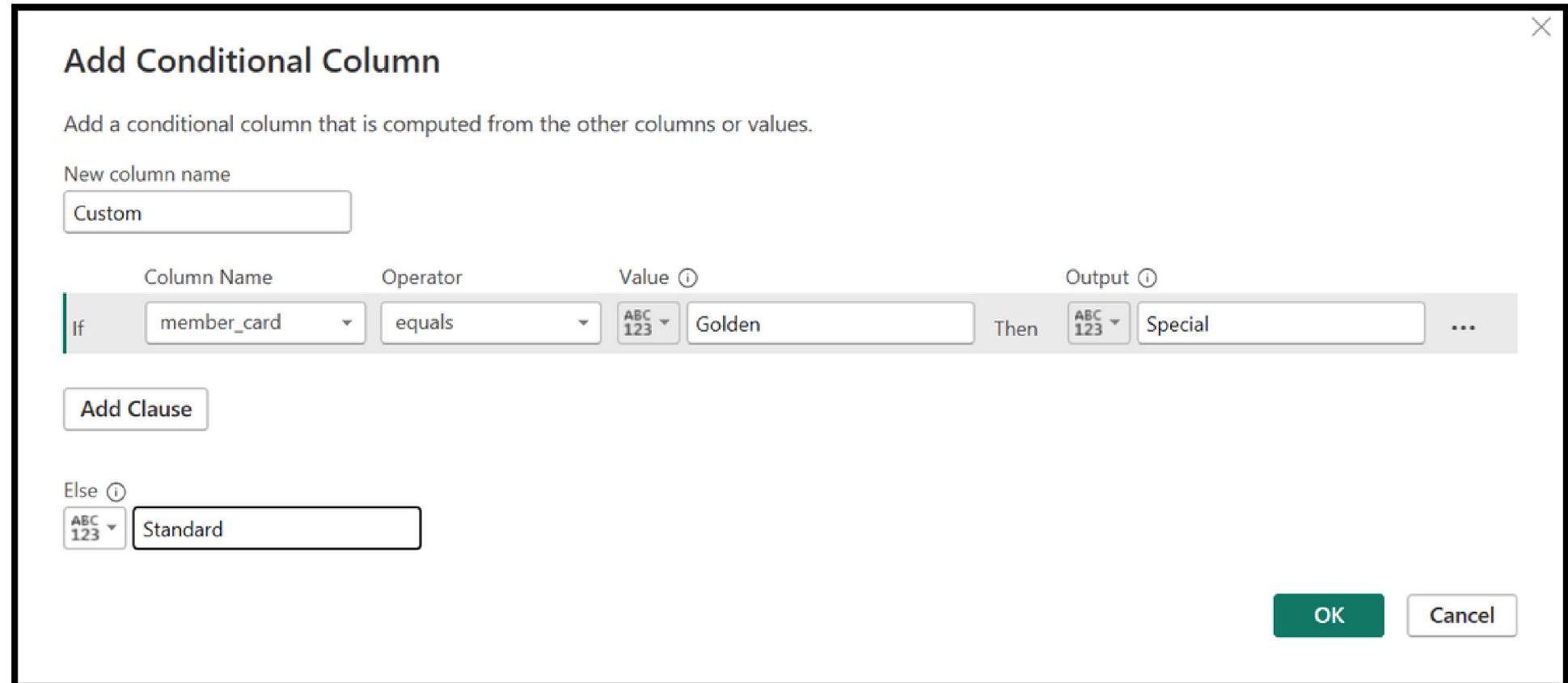
customer_id customer_acct_num first_name last_name customer_address customer_city customer_state_province customer_postal_code customer_country birthdate

customer_id	customer_acct_num	first_name	last_name	customer_address	customer_city	customer_state_province	customer_postal_code	customer_country	birthdate
213	89905924797	David	Watson	8157 W. Buchanan	Woodland Hills	CA	25400	USA	12/19/1915
227	90065194368	Michael	Suggs	40 Ellis St.	Burbank	CA	17728	USA	10/10/1963
246	90350758433	Rita	Santry	207 Barquentine Court	Lincoln Acres	CA	72730	USA	8/18/1924
255	90442248582	Jacky	Camille	3919 El Pintado Road	Torrance	CA	71696	USA	3/15/1965
258	90463679432	John	Minker	1061 Buskrik Avenue	Coronado	CA	95593	USA	1/12/1912
273	90642099976	Jennifer	Confetti	2775 Robinson Ave.	Long Beach	CA	89348	USA	8/2/1925
289	90872421300	Bertha	Jameson	3029 Heather Leaf Ln.	Los Angeles	CA	31309	USA	6/9/1948
294	90917686200	Michelle	Adams	9558 Orchard View Ave.	Imperial Beach	CA	32723	USA	4/23/1935
311	91123164892	Janice	Vrins	1055 Horseshoe Road	Pomona	CA	43683	USA	7/19/1965
313	91158838272	Ole	Weldon	5754 Glenhaven Ave	Palo Alto	CA	30993	USA	6/21/1931
331	91323767304	Lanna	Slaven	5118 Boxwood Dr.	Mill Valley	CA	71789	USA	2/12/1956
390	92005987200	Paul	Alcorn	4822 Center Street	Fremont	CA	18956	USA	3/26/1973
394	92050951719	Jared	Bustamante	4222 San Jose Dr.	Concord	CA	44256	USA	2/17/1910
425	92441642284	Margaret	Adams	8452 Dewing Avenue	San Carlos	CA	64065	USA	5/2/1979
457	92920920848	Ethel	Porter	2863 Polk Street	Torrance	CA	35114	USA	11/24/1929
482	93154197931	Vanessa	Tench	6621 Polaris Dr.	Imperial Beach	CA	69245	USA	1/10/1930
513	93673430681	Lenore	Yasi	1245 West Hookston Road	San Jose	CA	95838	USA	4/7/1932
574	94377177953	Catherine	Whitney	1239 Linnet Court	La Jolla	CA	76380	USA	5/18/1966
577	94402787700	Stacey	Cereghino	852 Santa Maria	Coronado	CA	47561	USA	8/27/1943
601	94717700500	Marlin	Coriell	4824 Discovery Bay	Beverly Hills	CA	75442	USA	3/15/1933
755	96895353794	Deanna	Sabella	8942 Sierra Road	Woodland Hills	CA	23993	USA	6/13/1916
761	96979832649	Kimberly	Malmendier	6206 Heavenly Drive	West Covina	CA	41635	USA	4/18/1925
796	97392292022	Richard	Young	5634 Blue Ridge Drive	La Jolla	CA	78376	USA	5/17/1941
859	98214603700	Joan	Bitler	2775 Delta View Ln.	Mill Valley	CA	73498	USA	11/10/1942
868	98372167348	Nellie	Orando	7942 Palms	Lakewood	CA	25757	USA	8/17/1963
884	98581272363	Joseph	Thompson	2099 San Jose	Mill Valley	CA	68331	USA	6/13/1968
885	98606104254	Roberta	Sturgel	3086 Indigo Ct	El Cajon	CA	24720	USA	9/6/1919
960	99577841758	Kari	Sloper	5283 Rishell Ct.	Richmond	CA	71842	USA	4/23/1945
966	99645835757	Lois	Knobel	9636 Palisade Court	Colma	CA	77752	USA	7/25/1974
1090	10705184936	Paula	Tomlinson	8885 Partridge Dr.	Woodland Hills	CA	65219	USA	9/17/1948
1122	10950652600	T	U	1016 W. Linda Ln.	San Bruno	CA	73665	USA	11/10/1952

Table: Customers (10,281 rows)



Manipulating and Editing Data using Power Query

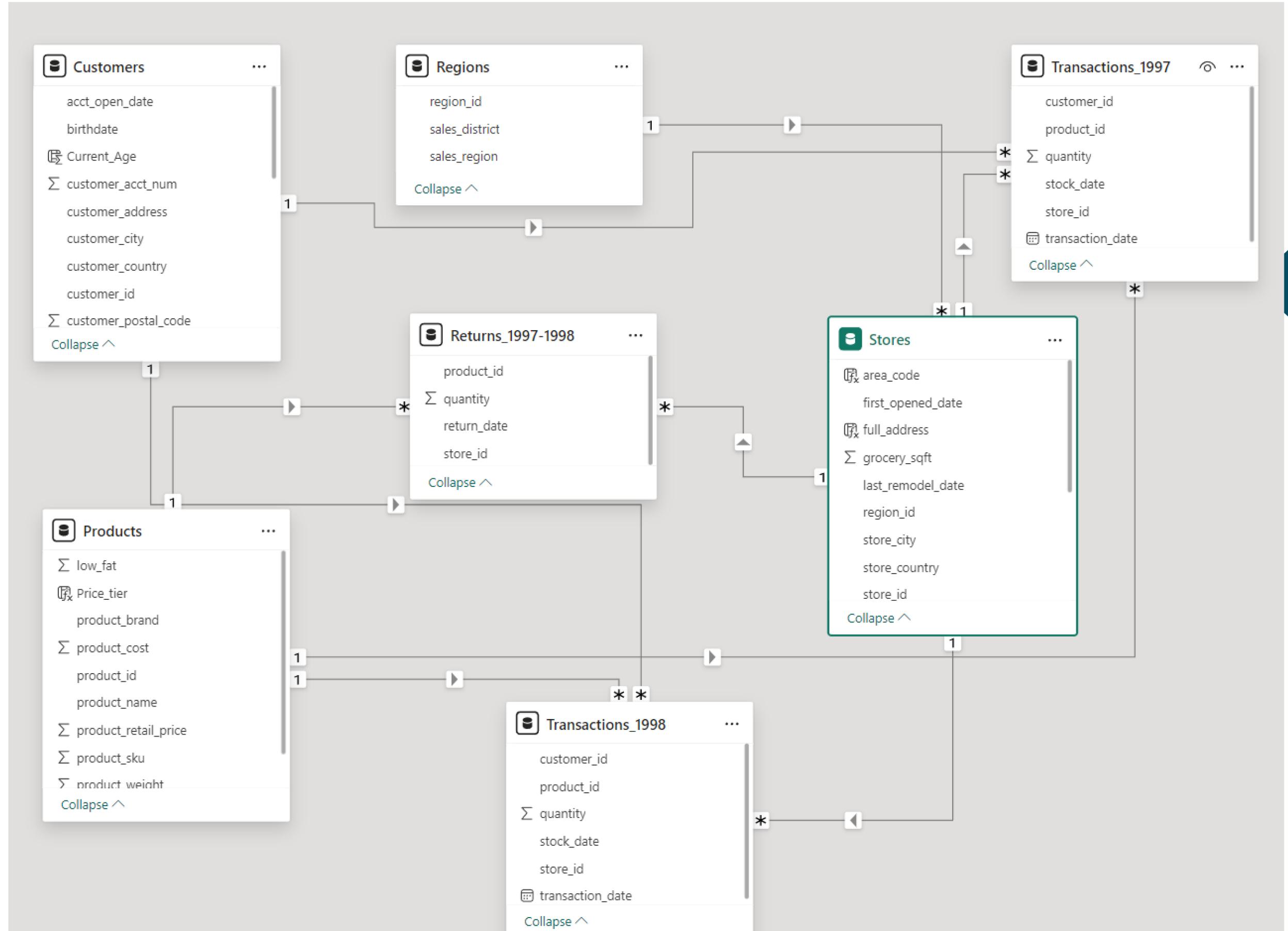


The screenshot shows the Power Query editor interface. The formula bar at the top contains the formula: `= Table.RenameColumns(#"Reordered Columns",{{"Custom", "Membership"}})`. The main area displays a table with 12 rows of data. The columns are labeled: 'id', 'home', 'education', 'acct_open_date', 'member_card', 'Membership', 'occupation', and 'homeowner'. The data includes various combinations of education levels (Partial High School, Bachelor's Degree, etc.) and membership types (Bronze, Silver, Normal, Golden, Standard, Special). The 'Membership' column is highlighted in red, indicating it is the target for the conditional column addition.

	id	home	education	acct_open_date	member_card	Membership	occupation	homeowner
1	2	Partial High School	9/10/1991	Bronze	Standard	Skilled Manual	Y	
2	0	Partial High School	3/11/1993	Bronze	Standard	Professional	N	
3	1	Bachelors Degree	6/11/1991	Bronze	Standard	Professional	Y	
4	4	Partial High School	5/21/1994	Normal	Standard	Skilled Manual	N	
5	0	Partial College	8/21/1992	Silver	Standard	Manual	N	
6	0	Bachelors Degree	4/5/1992	Bronze	Standard	Professional	Y	
7	1	Partial High School	5/1/1991	Bronze	Standard	Manual	Y	
8	2	Bachelors Degree	6/11/1992	Bronze	Standard	Professional	Y	
9	3	Partial High School	11/14/1993	Normal	Standard	Skilled Manual	Y	
10	0	Bachelors Degree	5/4/1993	Golden	Special	Management	N	
11	0	High School Degree	4/23/1992	Bronze	Standard	Manual	N	
12	0	High School Degree	7/6/1991	Bronze	Standard	Skilled Manual	N	



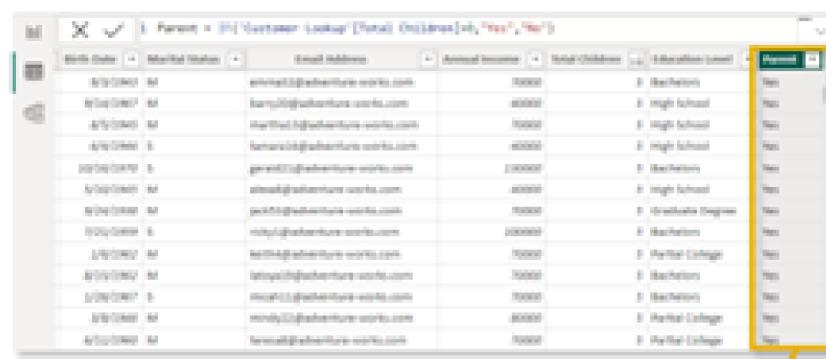
Model View of the Dataset



Calculating Measures using DAX Queries

CALCULATED COLUMNS

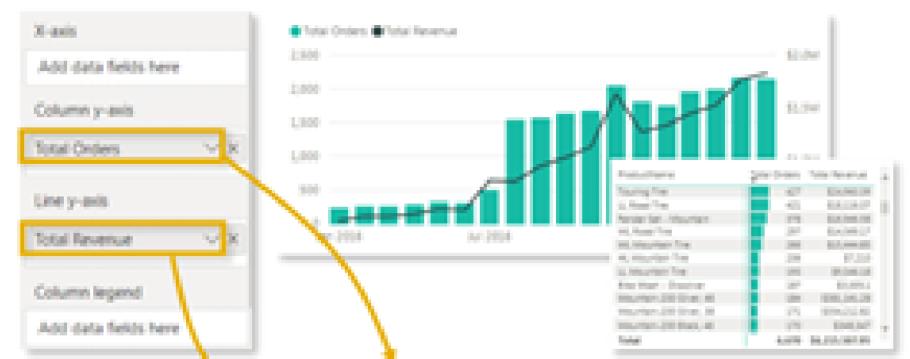
- Values are calculated based on information from each row of a table (**row context**)
- Appends static values to each row in a table and stores them in the model (*which increases file size*)
- Recalculate on data source refresh or when changes are made to component columns
- Primarily used for **filtering** data in reports



A screenshot of the Power BI Data View interface. It shows a table named 'Customer_Lookup [Total_Orders, City, PostCode, Name]'. The table contains several rows of data. A yellow box highlights the 'Total_Orders' column header. Below the table, the text 'Calculated columns "live" in tables' is written.

MEASURES

- Values are calculated based on information from any filters in the report (**filter context**)
- Does not create new data in the tables themselves (*doesn't increase file size*)
- Recalculate in response to any change to filters within the report
- Primarily used for **aggregating values** in report visuals



Profit Margin_1997 = [Total Profit_1997] / [Total Revenue_1997]

Quantity Returned = SUM('Returns_1997-1998'[quantity])

Quantity Sold_1997 = SUM(Transactions_1997[quantity])

Return_Rate_1997 = [Quantity Returned] / [Quantity Sold_1997]

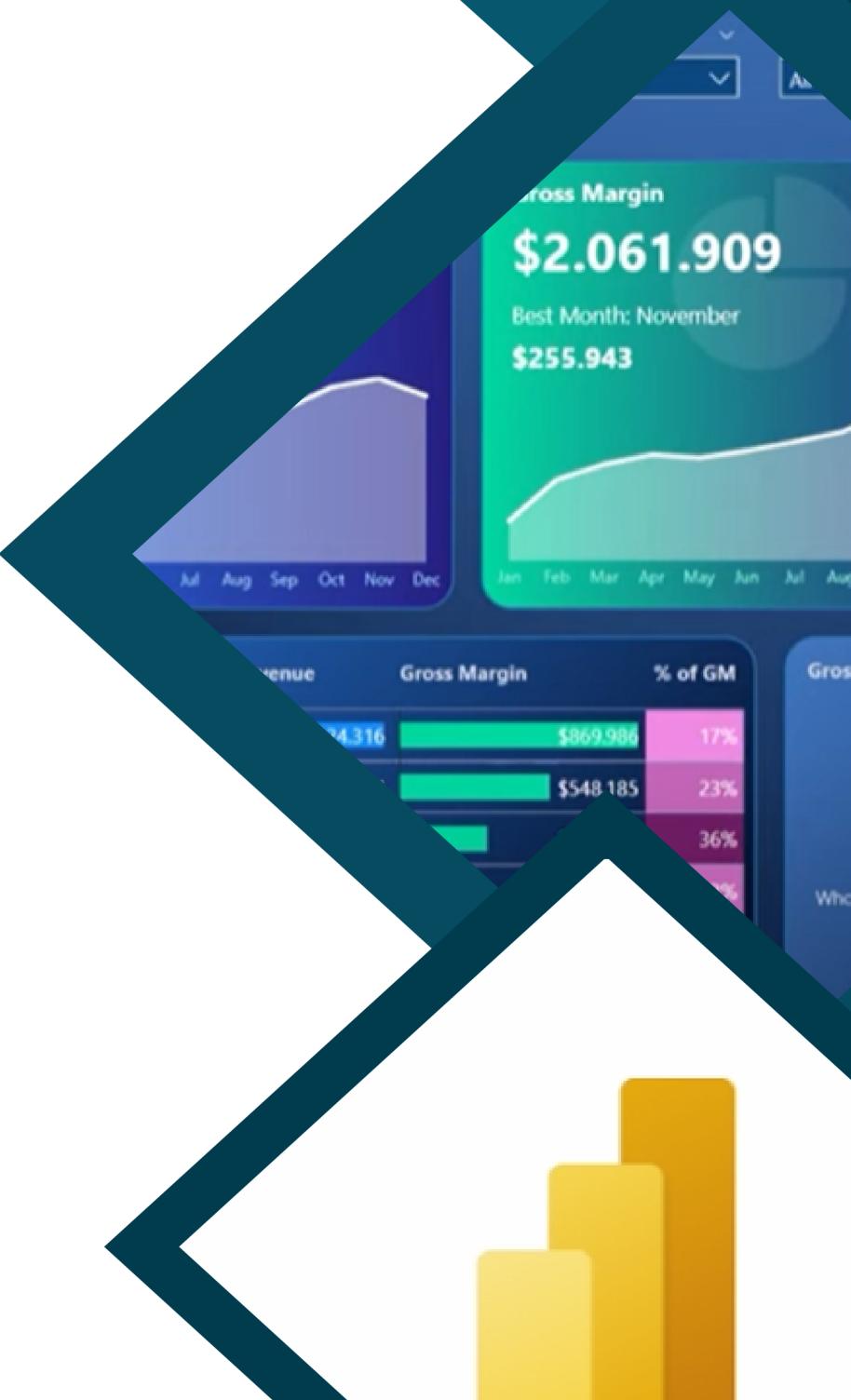
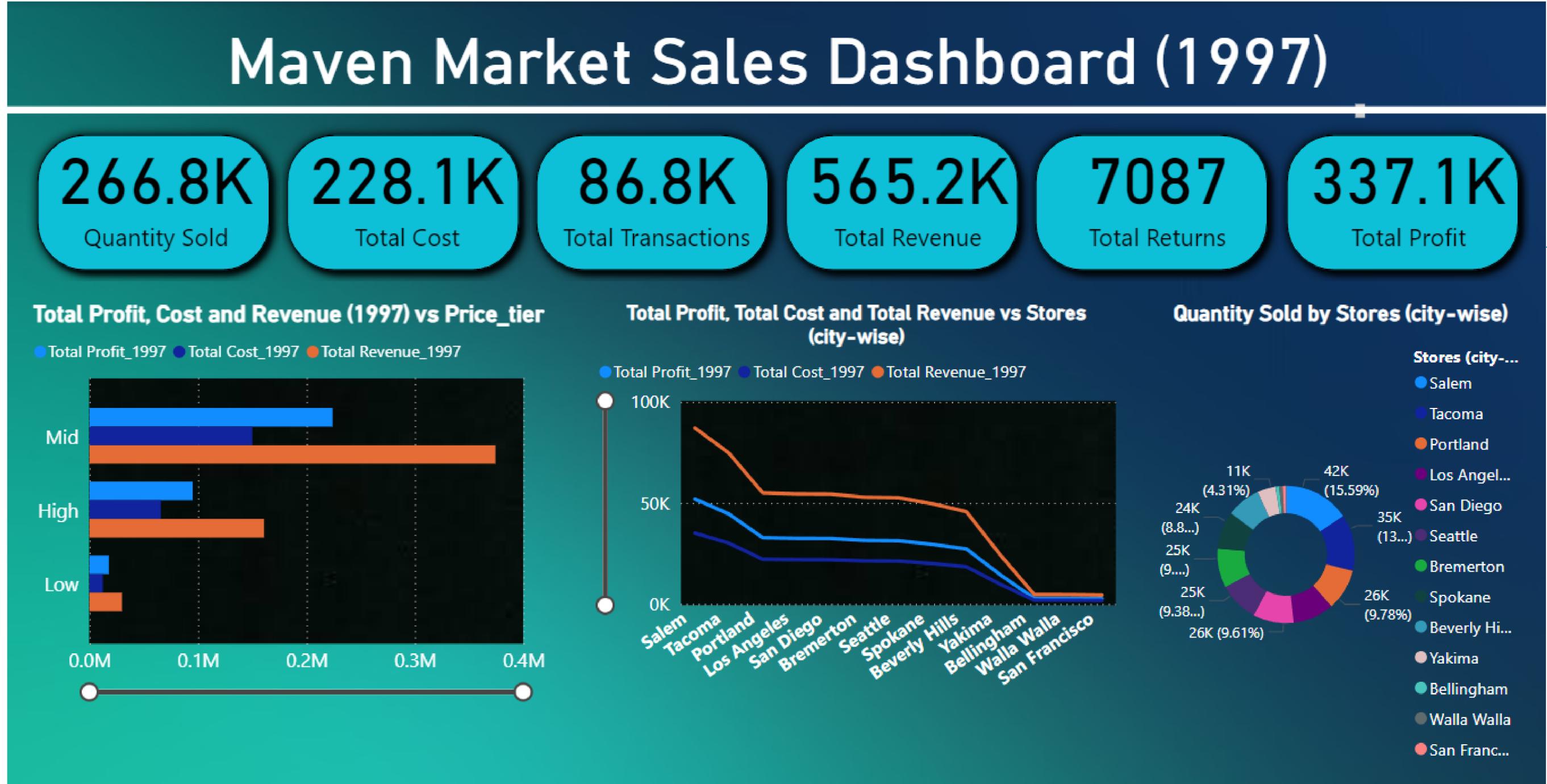
Total Cost_1997 = SUMX(Transactions_1997, Transactions_1997[quantity] * RELATED(Products[product_cost]))

Total Profit_1997 = [Total Revenue_1997] - [Total Cost_1997]

Total Revenue_1997 = SUMX(Transactions_1997, Transactions_1997[quantity] * RELATED(Products[product_retail_price]))



Visualization of Dataset by designing a Dashboard



References

- Simplilearn. (2021, April 16). Data Analyst: Job Description, Responsibilities and Skills Required. Simplilearn.com. <https://www.simplilearn.com/data-analyst-job-description-article>
- Data Analyst vs. Data Scientist: What's the Difference? (2023, November 29). Coursera. <https://www.coursera.org/in/articles/data-analyst-vs-data-scientist-whats-the-difference>
- The Types of Data Science Roles Explained. (2022, October 7). 365 Data Science. <https://365datascience.com/career-advice/types-of-data-science-roles-explained/>
- Python Software Foundation. (n.d.). What Is Python? Executive Summary. Python. <https://www.python.org/doc/essays/blurb/>
- What is Python? (2023, June 4). GeeksforGeeks. <https://www.geeksforgeeks.org/what-is-python/>
- VanderPlas, J. (2019). Python Data Science Handbook | Python Data Science Handbook. Github.io. <https://jakevdp.github.io/PythonDataScienceHandbook/>
- Python. (2020). The Python Standard Library — Python 3.8.1 documentation. Python.org. <https://docs.python.org/3/library/>
- SQL Tutorial. (n.d.). <https://www.w3schools.com/sql/>
- Groff, J., Weinberg, P., & Oppel, A. (2009, August 12). SQL The Complete Reference, 3rd Edition. McGraw-hill.
- Learn MySQL: A Comprehensive Guide to Using and Managing Databases. (n.d.). <https://www.w3schools.in/mysql/>
- SQL Tutorial, Tutorials SQL. (n.d.). <https://beginner-sql-tutorial.com/sql.htm>
- Introduction to MS-Excel. (2021, May 29). GeeksforGeeks. <https://www.geeksforgeeks.org/introduction-to-ms-excel/>
- What is MS Excel? | An Overview, Features & History. (n.d.). BYJUS. <https://byjus.com/govt-exams/ms-excel-basics/>
- Create a simple formula in Excel. (n.d.). Support.microsoft.com. <https://support.microsoft.com/en-us/office/create-a-simple-formula-in-excel-11a5f0e5-38a3-4115-85bc-f4a465f64a8a>
- The 15 Basic Excel Formulas Everyone Needs to Know. (2023, December) <https://www.datacamp.com/tutorial/basic-excel-formulas-for-everyone>
- Power BI Tutorial - Tutorialspoint. (n.d.). [Www.tutorialspoint.com. https://www.tutorialspoint.com/power_bi/index.htm](https://www.tutorialspoint.com/power_bi/index.htm)
- Microsoft. (2023, February 22). What Is Power BI? - Power BI. Learn.microsoft.com. <https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview>
- Microsoft. (2024). Power BI - Data Visualization | Microsoft Power Platform. [Www.microsoft.com. https://www.microsoft.com/en-us/power-platform/products/power-bi](https://www.microsoft.com/en-us/power-platform/products/power-bi)
- Power BI vs. Excel: Three Major Differences. (n.d.). Intellipaat.com. <https://intellipaat.com/blog/power-bi-vs-excel/>

Thank You

For Your Attention

