मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)

University of Mumbai

DEPARTMENT OF COMPUTER SCIENCE

M. Sc. (Computer Science) (NEP) Semester-III

2025-2026

Data Visualization

Elective - II

Submitted by
Nayan Naresh Khuje

Seat No.

मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)

University of Mumbai

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the work entered in this journal was done in the University Department of Computer Science laboratory by Mr. Nayan Naresh Khuje Seat No. ____________ for the course of **M.Sc. (Computer Science) - Semester III (NEP 2020)** during the academic year **2025- 2026** in a satisfactory manner.

Subject In-charge
Department of Computer Science

Head
Department of Computer Science

External Examiner

INDEX

Practical 1

Aim: Create Charts and Reports in Power BI.

Dataset Used: The dataset used for this practical is the Superstore dataset containing fields like Order Date, Sales, Profit, Category, Sub-Category, and more. It spans data from 2016 to 2019.

Chart 1: Stacked Bar Chart

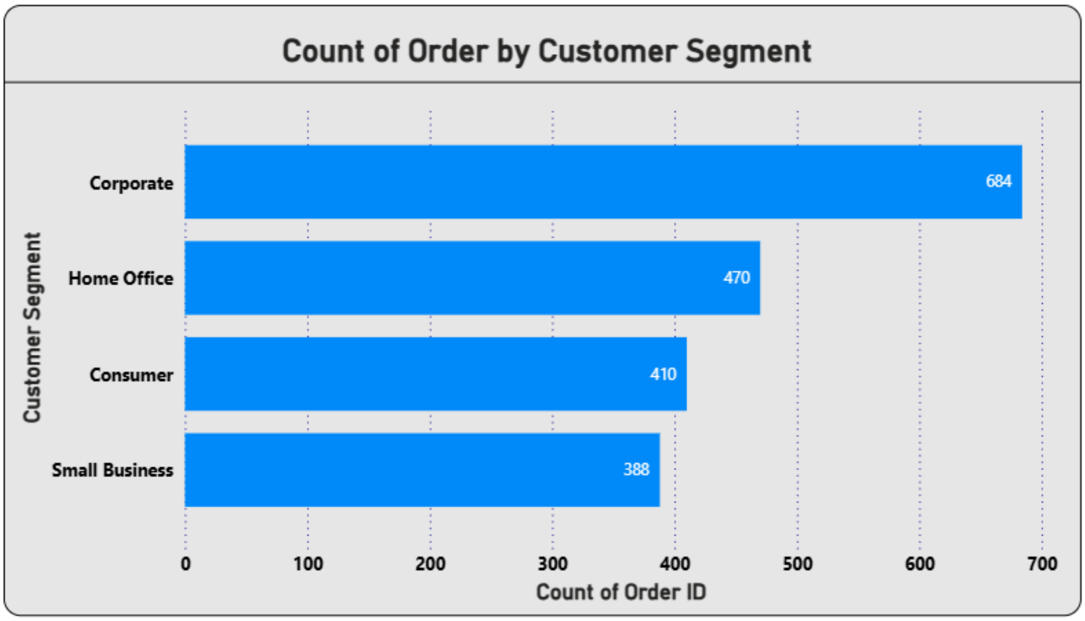


Chart 2: Funnel Chart

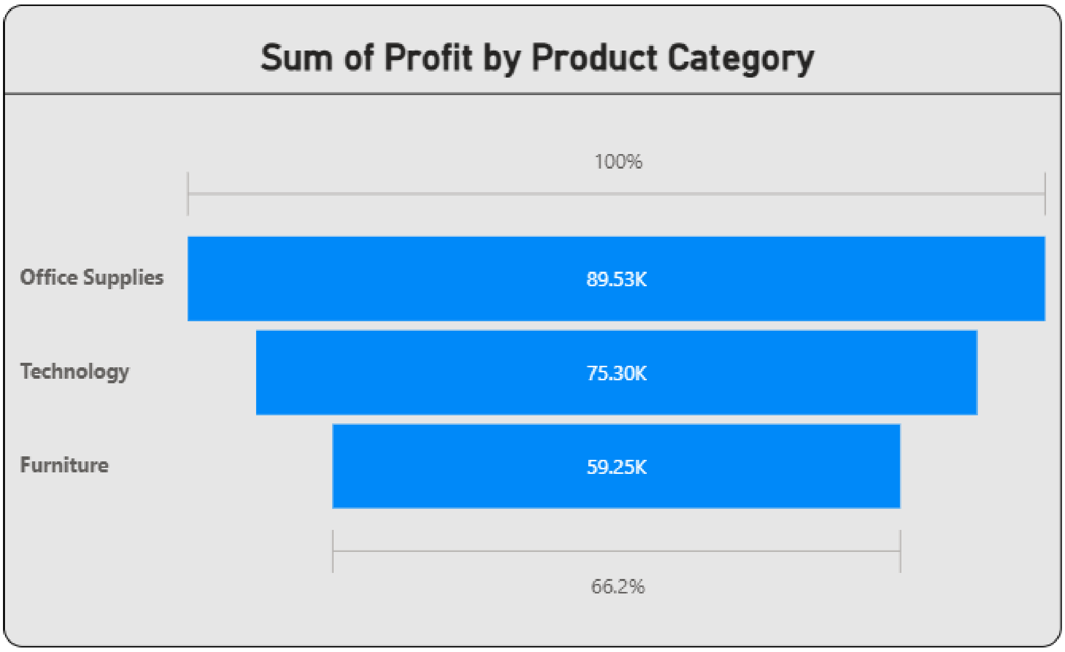


Chart 2: Donut Chart

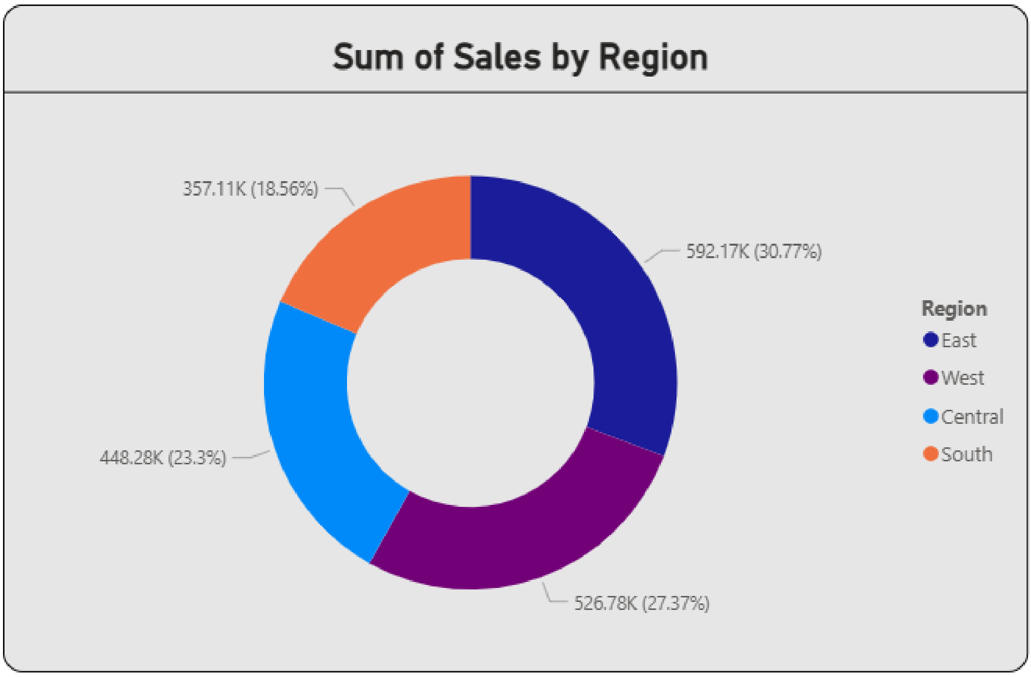
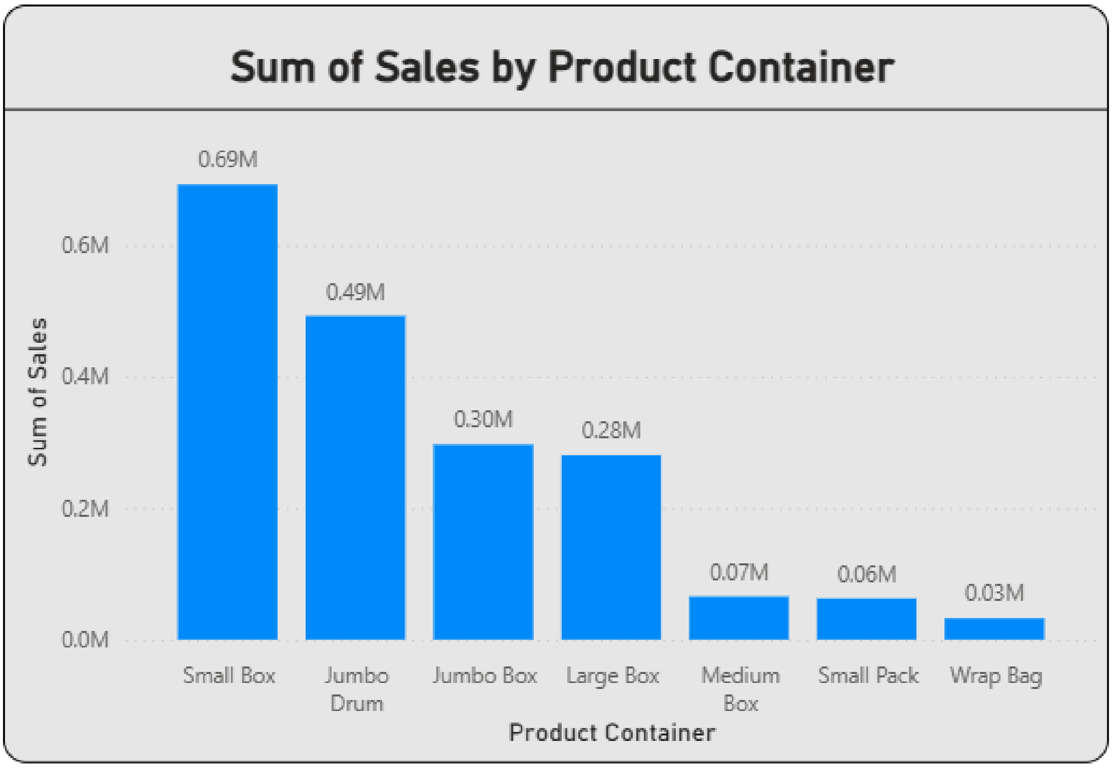


Chart 4: Stacked Column Chart



Practical 2

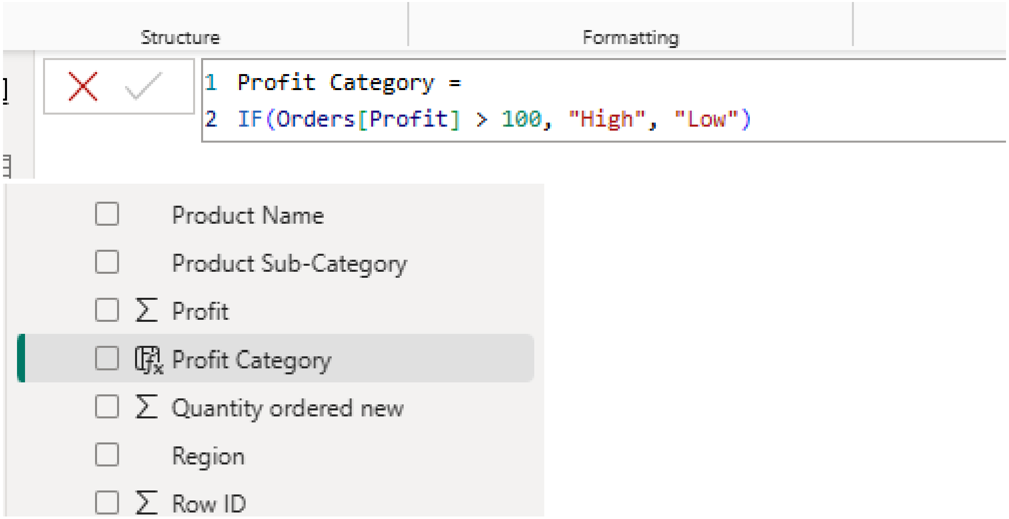
Aim: Time Intelligence and data analysis Functions with DAX

Dataset Used: The dataset used for this practical is the Superstore dataset containing fields like Order Date, Sales, Profit, Category, Sub-Category, and more. It spans data from 2016 to 2019.

1. Profit Category Classification

Objective: To classify profits into High and Low categories based on a threshold.

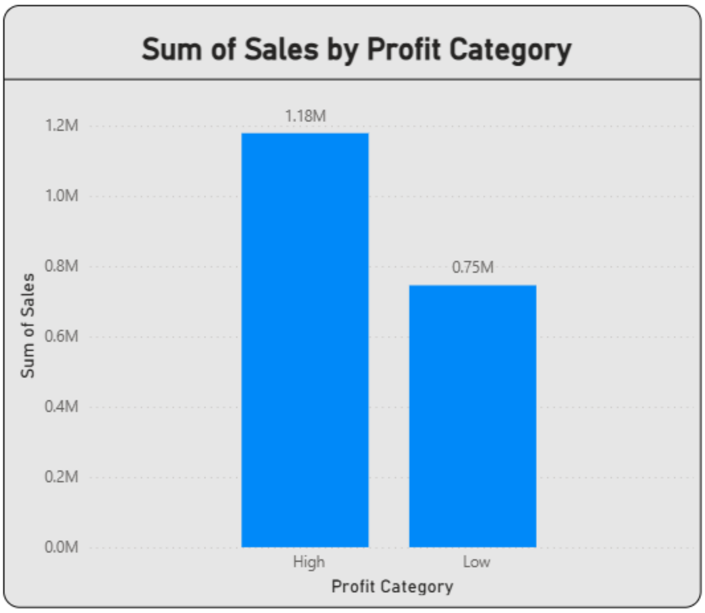
```
Profit Category = IF(Orders[Profit] > 100, "High", "Low")
```



```
Profit Category =
IF(Orders[Profit] > 100, "High", "Low")
```

☐ Product Name
☐ Product Sub-Category
☐ Σ Profit
☐ Profit Category
☐ Σ Quantity ordered new
☐ Region
☐ Σ Row ID

Column Chart: Profit Category:

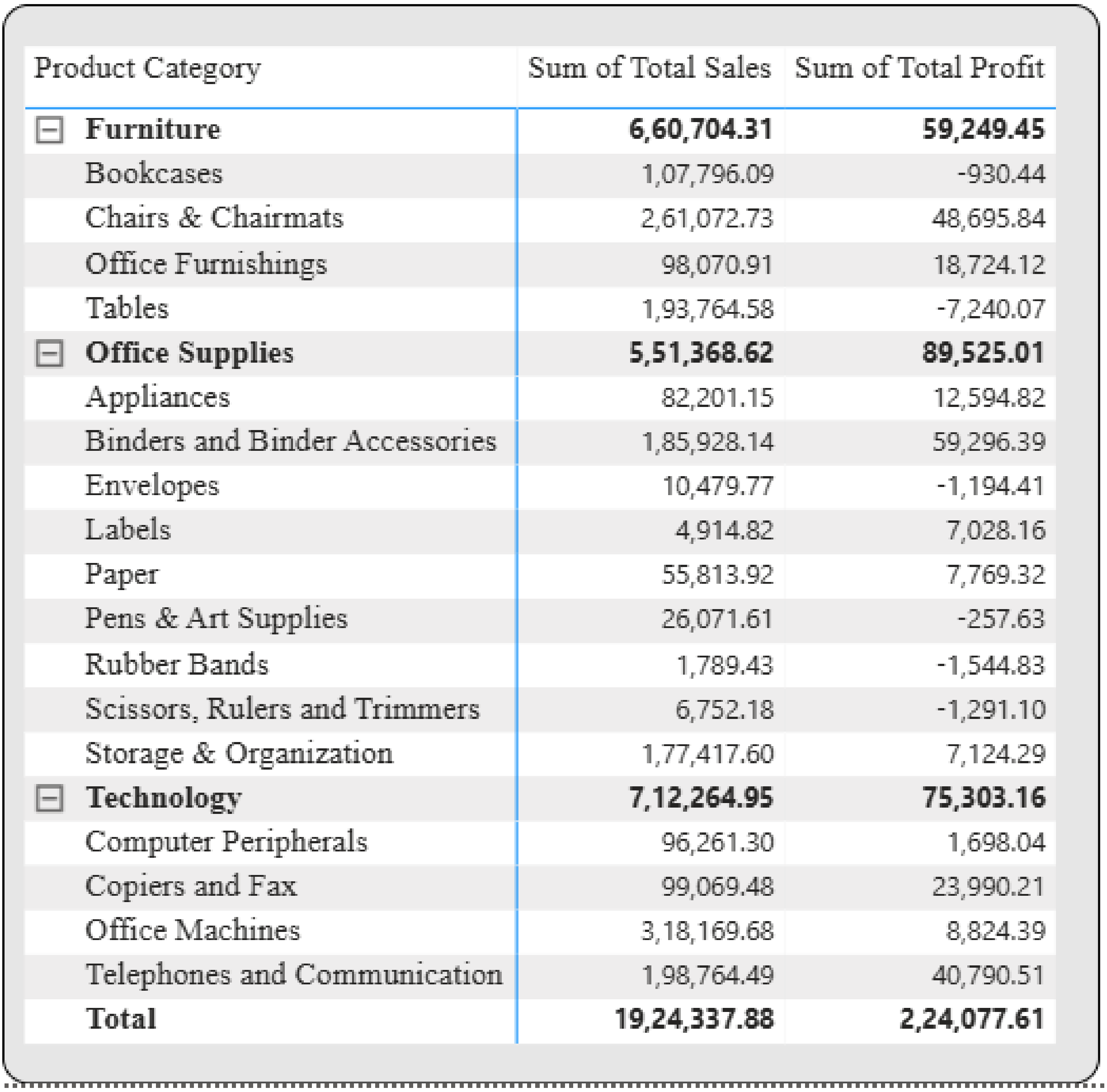


2. Group By and Summarize

Objective: To group data by Category and Sub-Category and calculate total sales and profit.

```
Category Summary =
SUMMARIZE(
            Orders, Orders[Category], Orders[Sub-Category], "Total Sales",
SUM(Orders[Sales]), "Total Profit", SUM(Orders[Profit]) )
```

Matrix: Sales and Profit by Category

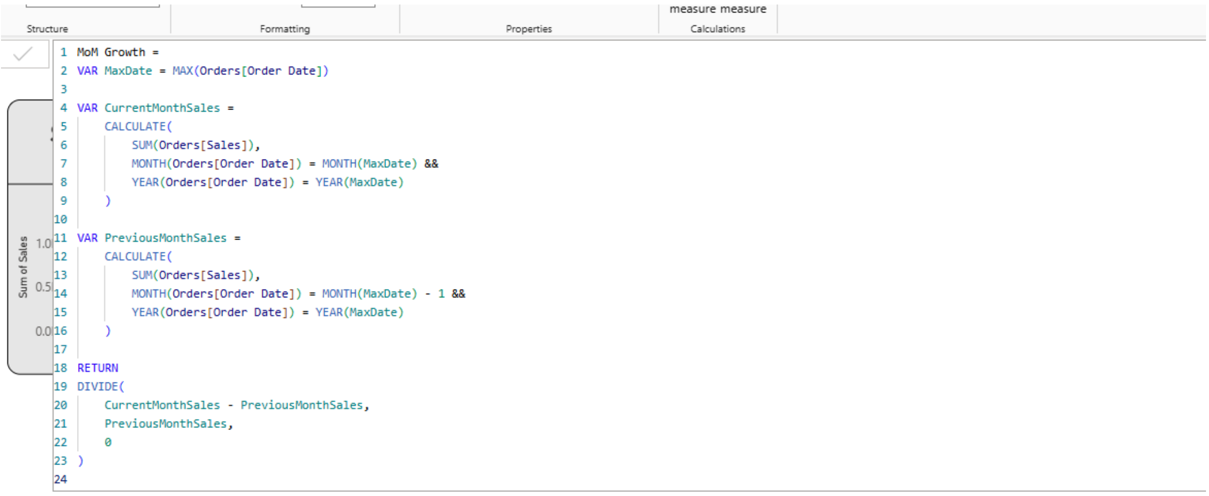


Product Category	Sum of Total Sales	Sum of Total Profit
⊟ **Furniture**	**6,60,704.31**	**59,249.45**
Bookcases	1,07,796.09	-930.44
Chairs & Chairmats	2,61,072.73	48,695.84
Office Furnishings	98,070.91	18,724.12
Tables	1,93,764.58	-7,240.07
⊟ **Office Supplies**	**5,51,368.62**	**89,525.01**
Appliances	82,201.15	12,594.82
Binders and Binder Accessories	1,85,928.14	59,296.39
Envelopes	10,479.77	-1,194.41
Labels	4,914.82	7,028.16
Paper	55,813.92	7,769.32
Pens & Art Supplies	26,071.61	-257.63
Rubber Bands	1,789.43	-1,544.83
Scissors, Rulers and Trimmers	6,752.18	-1,291.10
Storage & Organization	1,77,417.60	7,124.29
⊟ **Technology**	**7,12,264.95**	**75,303.16**
Computer Peripherals	96,261.30	1,698.04
Copiers and Fax	99,069.48	23,990.21
Office Machines	3,18,169.68	8,824.39
Telephones and Communication	1,98,764.49	40,790.51
Total	**19,24,337.88**	**2,24,077.61**

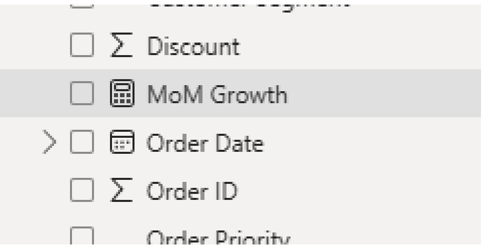
3. To analyze sales and profit performance over specific time periods.

Objective: Month-to-Date (MTD), and Quarter-to-Date (QTD) metrics.

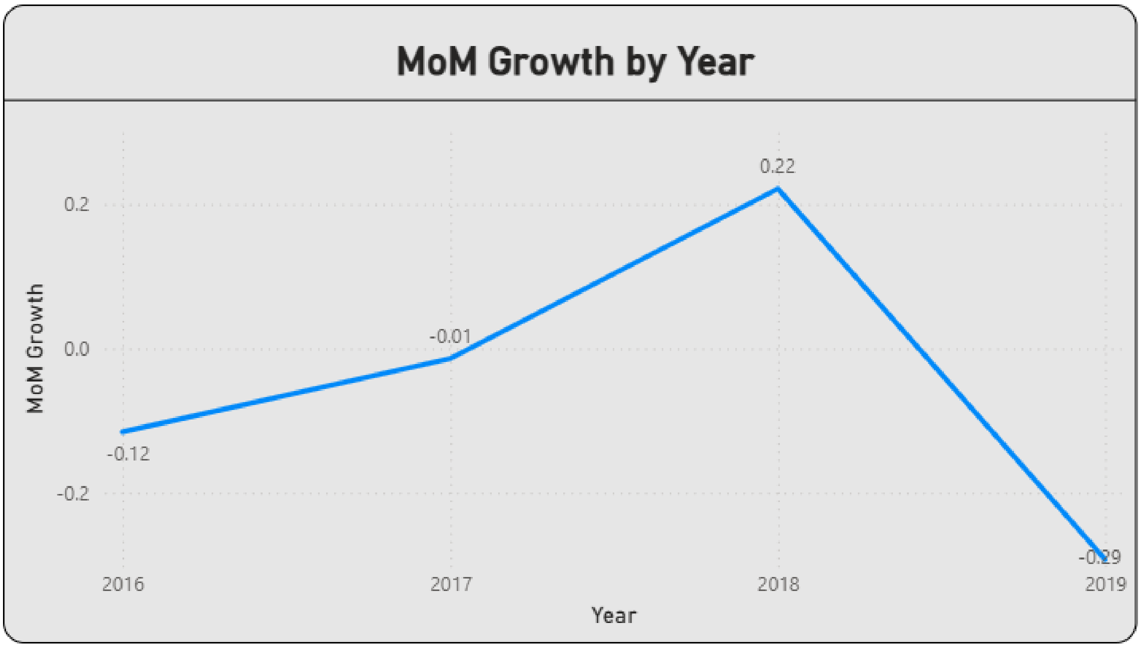
```
MoM Growth =
VAR MaxDate = MAX(Orders[Order Date])
VAR CurrentMonthSales =
            CALCULATE(
                        SUM(Orders[Sales]),
                        MONTH(Orders[Order Date]) = MONTH(MaxDate) && YEAR(Orders[Order
                        Date]) = YEAR(MaxDate)
                         )
VAR PreviousMonthSales =
            CALCULATE(
                         SUM(Orders[Sales]),
                        MONTH(Orders[Order Date]) = MONTH(MaxDate) - 1 && YEAR(Orders[Order
                        Date]) = YEAR(MaxDate)
                        )
 RETURN
DIVIDE(
            CurrentMonthSales - PreviousMonthSales,
            PreviousMonthSales,
            0
            )
```



```
MoM Growth =
VAR MaxDate = MAX(Orders[Order Date])

VAR CurrentMonthSales =
    CALCULATE(
        SUM(Orders[Sales]),
        MONTH(Orders[Order Date]) = MONTH(MaxDate) &&
        YEAR(Orders[Order Date]) = YEAR(MaxDate)
    )

VAR PreviousMonthSales =
    CALCULATE(
        SUM(Orders[Sales]),
        MONTH(Orders[Order Date]) = MONTH(MaxDate) - 1 &&
        YEAR(Orders[Order Date]) = YEAR(MaxDate)
    )

RETURN
DIVIDE(
    CurrentMonthSales - PreviousMonthSales,
    PreviousMonthSales,
    0
)
```



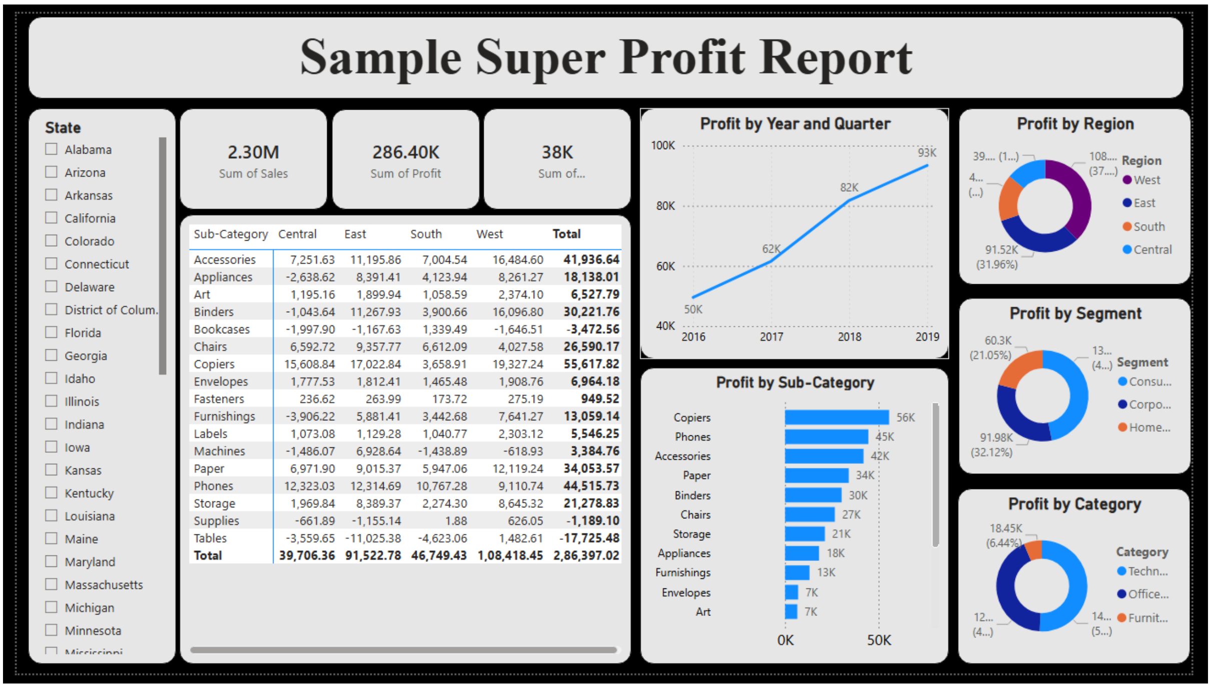
Line Chart: Month-over-Month Growth



Practical 3

Aim: Operations on Pinned Reports and Visuals using Power BI.

Sample superstore report:



Sub-Category	Central	East	South	West	**Total**
Accessories	7,251.63	11,195.86	7,004.54	16,484.60	**41,936.64**
Appliances	-2,638.62	8,391.41	4,123.94	8,261.27	**18,138.01**
Art	1,195.16	1,899.94	1,058.59	2,374.10	**6,527.79**
Binders	-1,043.64	11,267.93	3,900.66	16,096.80	**30,221.76**
Bookcases	-1,997.90	-1,167.63	1,339.49	-1,646.51	**-3,472.56**
Chairs	6,592.72	9,357.77	6,612.09	4,027.58	**26,590.17**
Copiers	15,608.84	17,022.84	3,658.91	19,327.24	**55,617.82**
Envelopes	1,777.53	1,812.41	1,465.48	1,908.76	**6,964.18**
Fasteners	236.62	263.99	173.72	275.19	**949.52**
Furnishings	-3,906.22	5,881.41	3,442.68	7,641.27	**13,059.14**
Labels	1,073.08	1,129.28	1,040.77	2,303.12	**5,546.25**
Machines	-1,486.07	6,928.64	-1,438.89	-618.93	**3,384.76**
Paper	6,971.90	9,015.37	5,947.06	12,119.24	**34,053.57**
Phones	12,323.03	12,314.69	10,767.28	9,110.74	**44,515.73**
Storage	1,969.84	8,389.37	2,274.30	8,645.32	**21,278.83**
Supplies	-661.89	-1,155.14	1.88	626.05	**-1,189.10**
Tables	-3,559.65	-11,025.38	-4,623.06	1,482.61	**-17,725.48**
Total	**39,706.36**	**91,522.78**	**46,749.43**	**1,08,418.45**	**2,86,397.02**

Dataset Used: The dataset utilized for this practical is the Superstore Sample Dataset, which is used to design and analyze the Sample Super Profit Report dashboard.
The dataset includes the following fields:

- Order Information: Order ID, Order Date, Ship Date, Ship Mode
- Customer Information: Customer ID, Customer Name, Segment
- Geographical Information: Country/Region, City, State, Postal Code, Region
- Product Information: Product ID, Category, Sub-Category, Product Name
- Sales Metrics: Sales, Quantity, Discount, Profit

Dashboard-Based Steps and Visualizations

1. KPI Cards (Used in Dashboard)

Fields Used: Sales, Profit, Quantity

Purpose: To display key performance indicators at the top of the dashboard for quick overview.

Outcome (From My Dashboard)

- Sales: 2.30M, Profit: 286.40K, Quantity: 38K

2. Profit by Sub-Category

Visualization Type: Bar Chart

Steps: 1. Drag Sub-Category to the Axis field.
2. Drag Profit to the Values field.

Purpose: To compare profit performance across different product sub-categories.

3. Profit by Year and Quarter

Visualization Type: Line Chart

Steps: 1. Drag Order Date to the Axis field.
2. Set the date hierarchy to Year and Quarter.
3. Drag Profit to the Values field.

Observation (From Dashboard): Profits were highest in specific quarters of 2019, indicating peak business periods.

4. Profit by Region

Visualization Type: Donut Chart

Steps: 1. Drag Region to the Legend field.
2. Drag Profit to the Values field.

Observation: West Region contributed the most profit (37.84%), Central Region had the least contribution (13.9%)

5. Profit by Segment

Visualization Type: Donut Chart

Steps: 1. Drag Segment to the Legend field.
2. Drag Profit to the Values field.

Observation: Consumer Segment performed best with 46.83% contribution, Home Office Segment contributed the least (21.05%)

6. Profit by Category

Visualization Type: Donut Chart

Steps: 1. Drag Category to the Legend field.
2. Drag Profit to the Values field.

Observation: Technology Category showed the highest profit (50.79%), Furniture Category had the lowest profit share (6.35%)

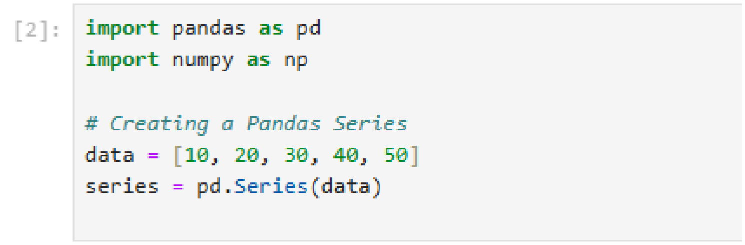
Practical No 4

Aim: Create one-dimensional data using series and perform various operations on it

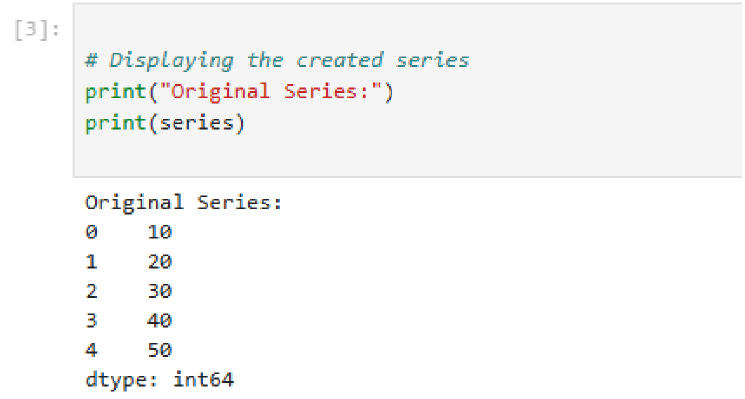
To create a one-dimensional data structure using a Pandas Series and perform various operations on it such as: Mathematical operations, Aggregation functions, Indexing and slicing, Conditional selection, Applying custom functions

Software / Tools Used: Python, Pandas library, NumPy library

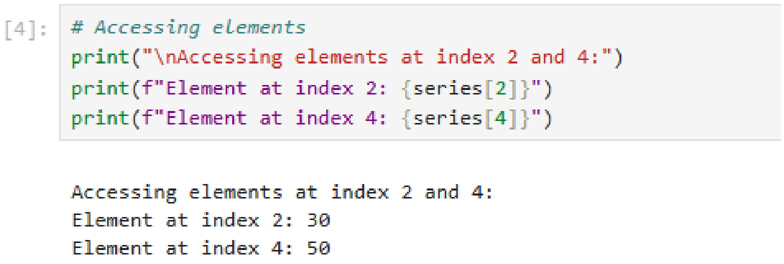
Theory: A Pandas Series is a one-dimensional labelled array capable of holding data of any type such as integers, floats, or strings. It supports various data manipulation operations which make it suitable for data analysis tasks.



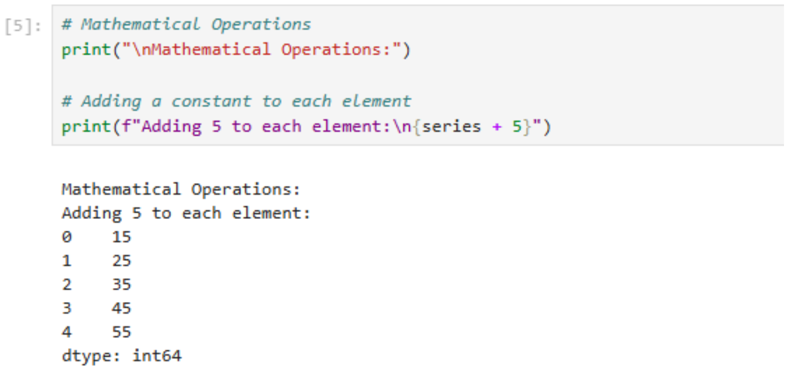
```
[2]: import pandas as pd
     import numpy as np

     # Creating a Pandas Series
     data = [10, 20, 30, 40, 50]
     series = pd.Series(data)
```



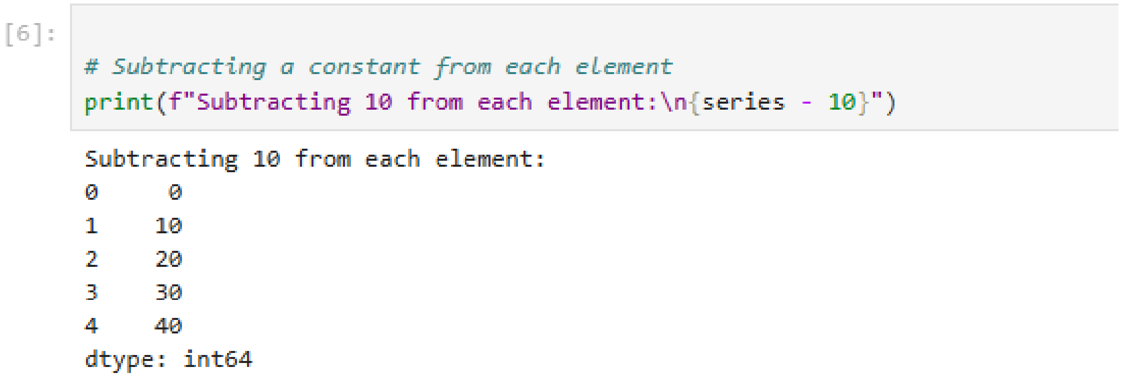
```
[3]:
     # Displaying the created series
     print("Original Series:")
     print(series)

     Original Series:
     0    10
     1    20
     2    30
     3    40
     4    50
     dtype: int64
```



```
[4]: # Accessing elements
     print("\nAccessing elements at index 2 and 4:")
     print(f"Element at index 2: {series[2]}")
     print(f"Element at index 4: {series[4]}")

     Accessing elements at index 2 and 4:
     Element at index 2: 30
     Element at index 4: 50
```

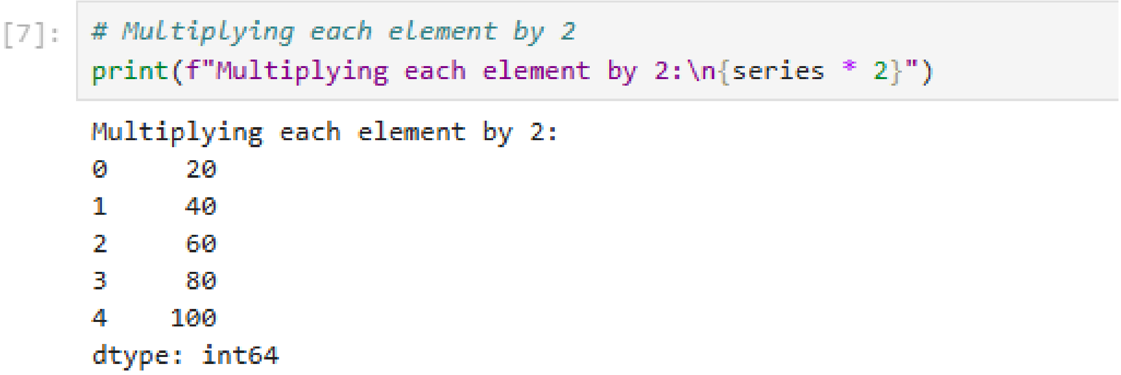


```
[5]: # Mathematical Operations
     print("\nMathematical Operations:")

     # Adding a constant to each element
     print(f"Adding 5 to each element:\n{series + 5}")

     Mathematical Operations:
     Adding 5 to each element:
     0    15
     1    25
     2    35
     3    45
     4    55
     dtype: int64
```



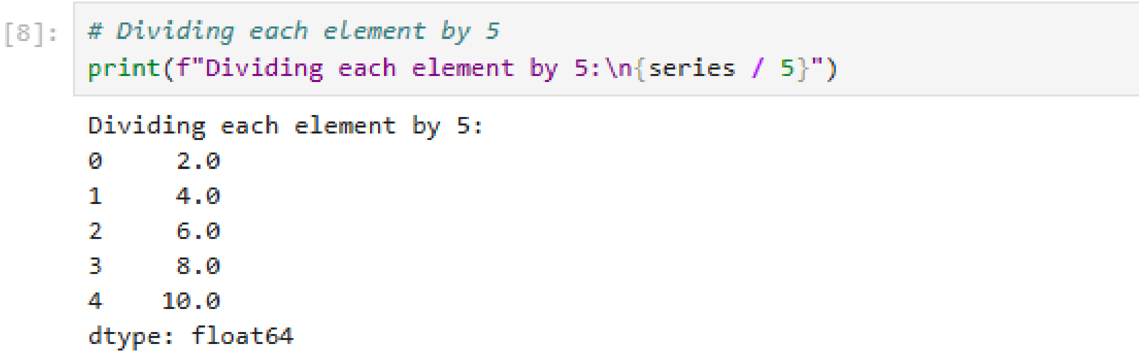
```
[6]:
# Subtracting a constant from each element
print(f"Subtracting 10 from each element:\n{series - 10}")
```

```
Subtracting 10 from each element:
0     0
1    10
2    20
3    30
4    40
dtype: int64
```



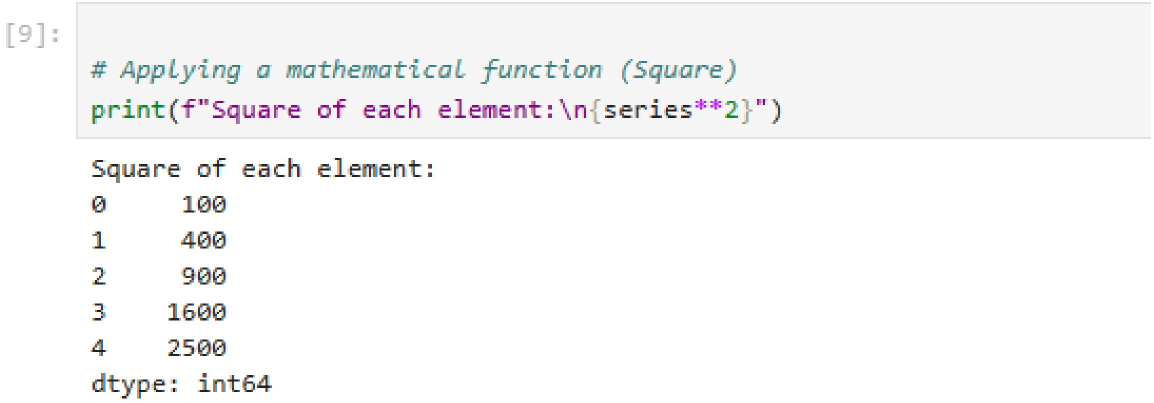
```
[7]:
# Multiplying each element by 2
print(f"Multiplying each element by 2:\n{series * 2}")
```

```
Multiplying each element by 2:
0     20
1     40
2     60
3     80
4    100
dtype: int64
```



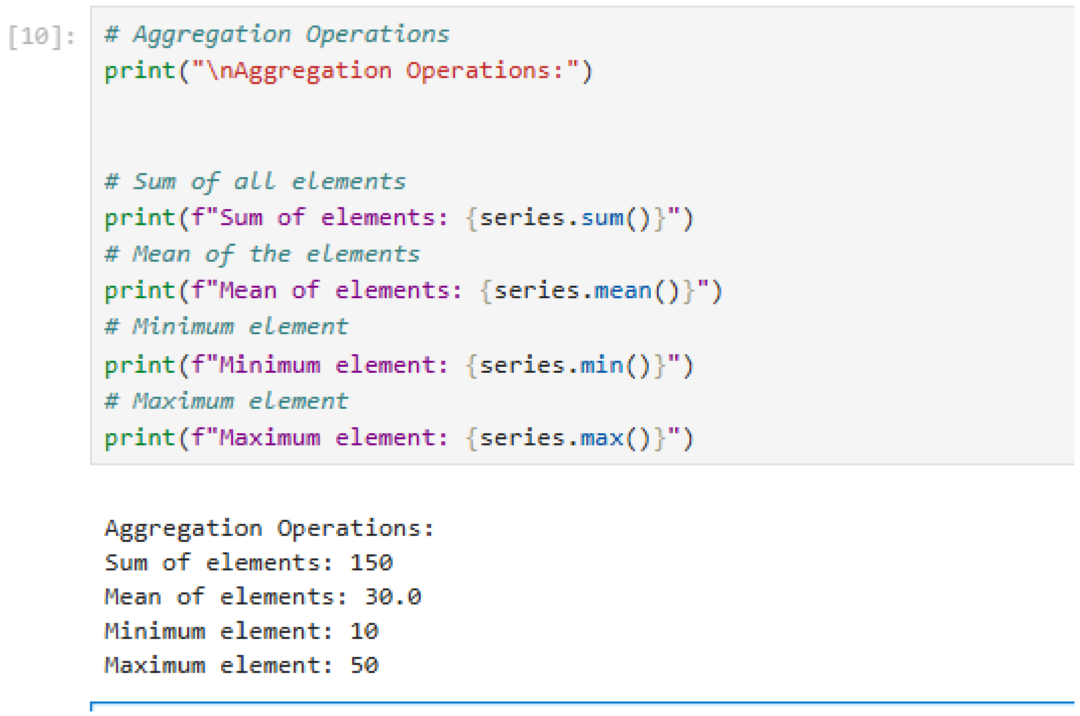
```
[8]:
# Dividing each element by 5
print(f"Dividing each element by 5:\n{series / 5}")
```

```
Dividing each element by 5:
0     2.0
1     4.0
2     6.0
3     8.0
4    10.0
dtype: float64
```



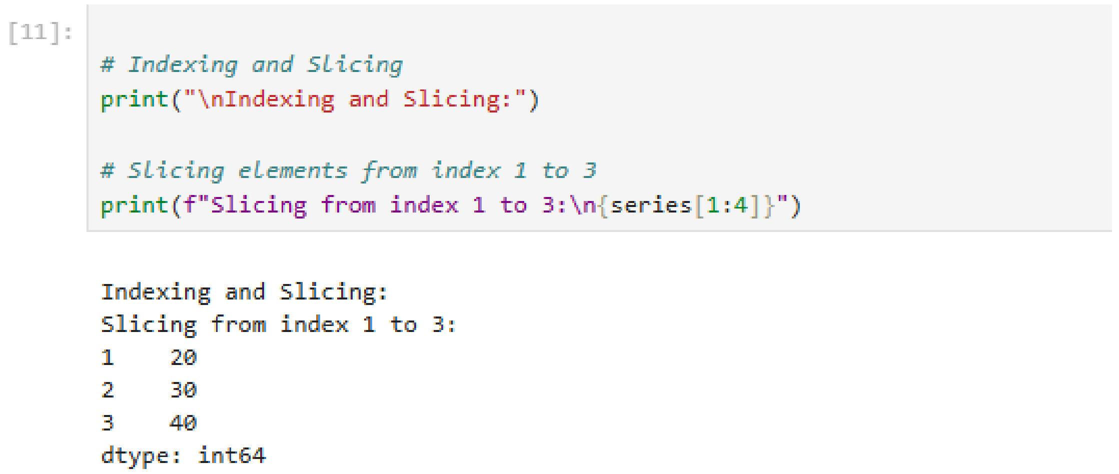
```
[9]:
# Applying a mathematical function (Square)
print(f"Square of each element:\n{series**2}")
```

```
Square of each element:
0     100
1     400
2     900
3    1600
4    2500
dtype: int64
```



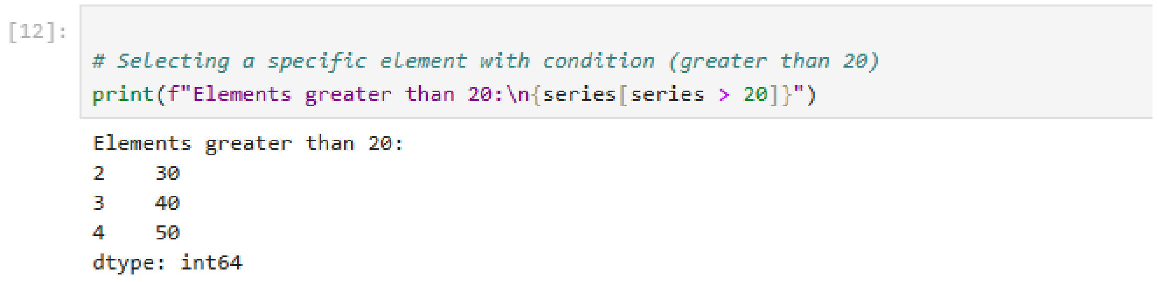
```
[10]: # Aggregation Operations
      print("\nAggregation Operations:")


      # Sum of all elements
      print(f"Sum of elements: {series.sum()}")
      # Mean of the elements
      print(f"Mean of elements: {series.mean()}")
      # Minimum element
      print(f"Minimum element: {series.min()}")
      # Maximum element
      print(f"Maximum element: {series.max()}")
```

```
Aggregation Operations:
Sum of elements: 150
Mean of elements: 30.0
Minimum element: 10
Maximum element: 50
```



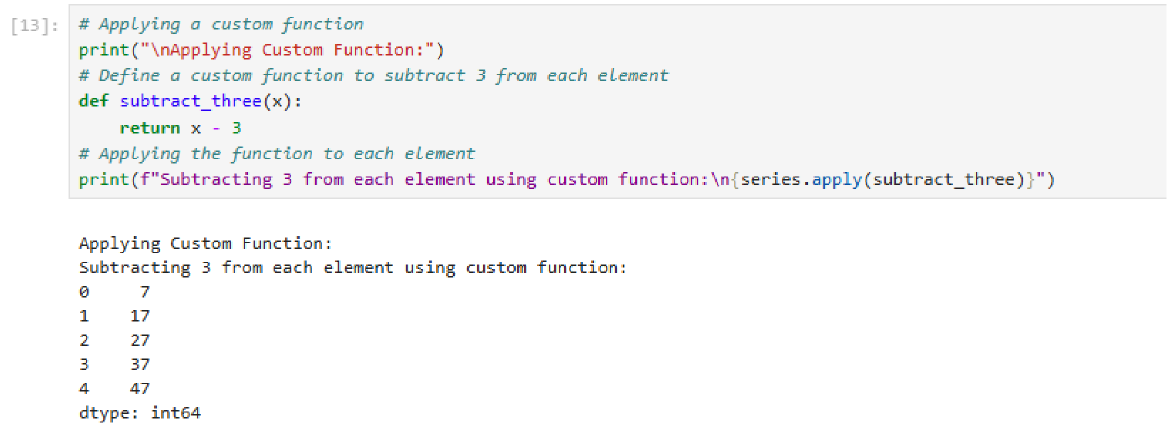
```
[11]:
      # Indexing and Slicing
      print("\nIndexing and Slicing:")

      # Slicing elements from index 1 to 3
      print(f"Slicing from index 1 to 3:\n{series[1:4]}")
```

```
Indexing and Slicing:
Slicing from index 1 to 3:
1    20
2    30
3    40
dtype: int64
```



```
[12]:
      # Selecting a specific element with condition (greater than 20)
      print(f"Elements greater than 20:\n{series[series > 20]}")
```

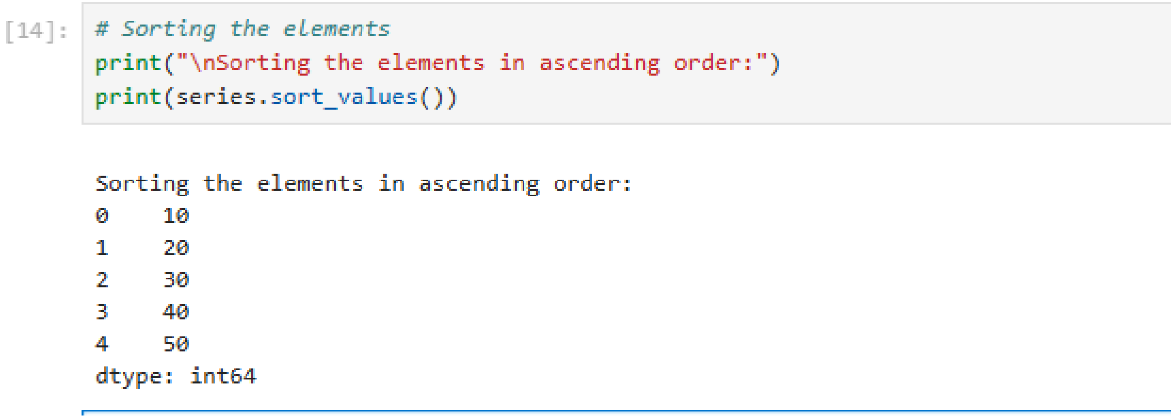
```
Elements greater than 20:
2    30
3    40
4    50
dtype: int64
```



[13]:

```
# Applying a custom function
print("\nApplying Custom Function:")
# Define a custom function to subtract 3 from each element
def subtract_three(x):
    return x - 3
# Applying the function to each element
print(f"Subtracting 3 from each element using custom function:\n{series.apply(subtract_three)}")
```

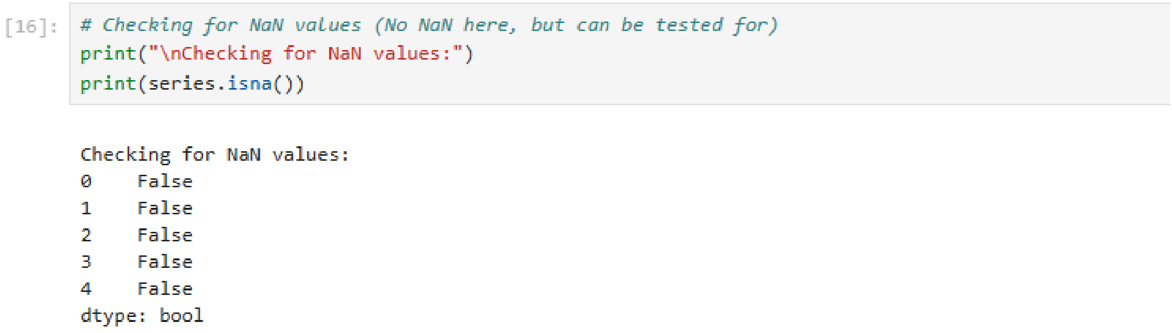
```
Applying Custom Function:
Subtracting 3 from each element using custom function:
0     7
1    17
2    27
3    37
4    47
dtype: int64
```



[14]:

```
# Sorting the elements
print("\nSorting the elements in ascending order:")
print(series.sort_values())
```

```

Sorting the elements in ascending order:
0    10
1    20
2    30
3    40
4    50
dtype: int64
```



[16]:

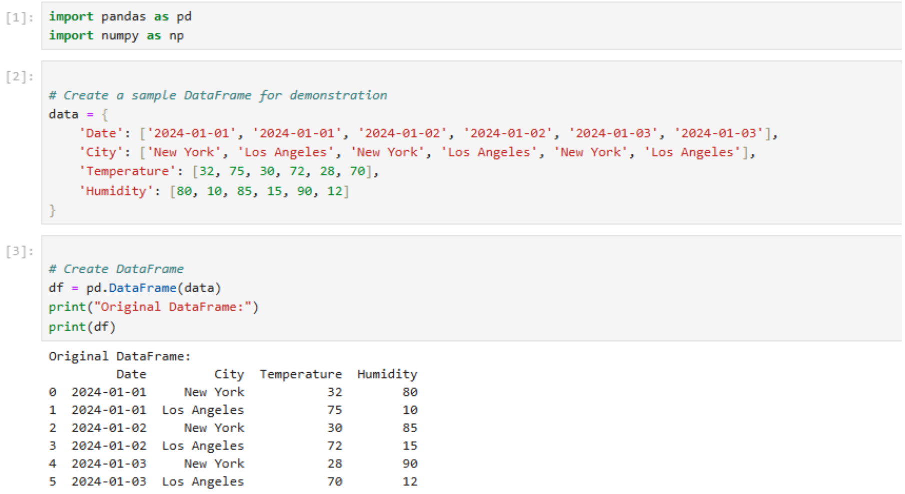
```
# Checking for NaN values (No NaN here, but can be tested for)
print("\nChecking for NaN values:")
print(series.isna())
```

```

Checking for NaN values:
0    False
1    False
2    False
3    False
4    False
dtype: bool
```

Practical 5

Aim: Perform Reshaping of the hierarchical data and pivoting data frame data

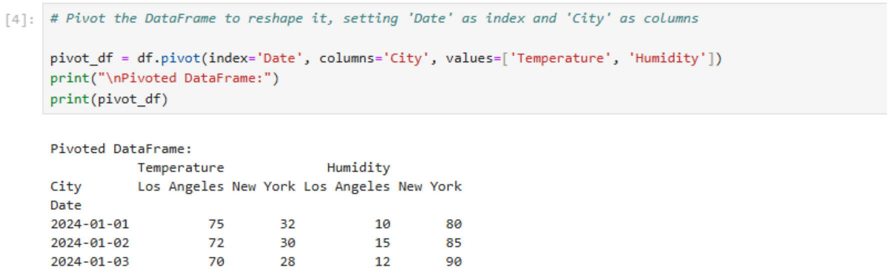
To perform reshaping operations such as pivoting, melting, stacking, and unstacking using a hierarchical Data Frame structure in Pandas.



```
[1]: import pandas as pd
     import numpy as np

[2]:
     # Create a sample DataFrame for demonstration
     data = {
         'Date': ['2024-01-01', '2024-01-01', '2024-01-02', '2024-01-02', '2024-01-03', '2024-01-03'],
         'City': ['New York', 'Los Angeles', 'New York', 'Los Angeles', 'New York', 'Los Angeles'],
         'Temperature': [32, 75, 30, 72, 28, 70],
         'Humidity': [80, 10, 85, 15, 90, 12]
     }

[3]:
     # Create DataFrame
     df = pd.DataFrame(data)
     print("Original DataFrame:")
     print(df)
```

```
Original DataFrame:
         Date         City  Temperature  Humidity
0  2024-01-01     New York           32        80
1  2024-01-01  Los Angeles           75        10
2  2024-01-02     New York           30        85
3  2024-01-02  Los Angeles           72        15
4  2024-01-03     New York           28        90
5  2024-01-03  Los Angeles           70        12
```

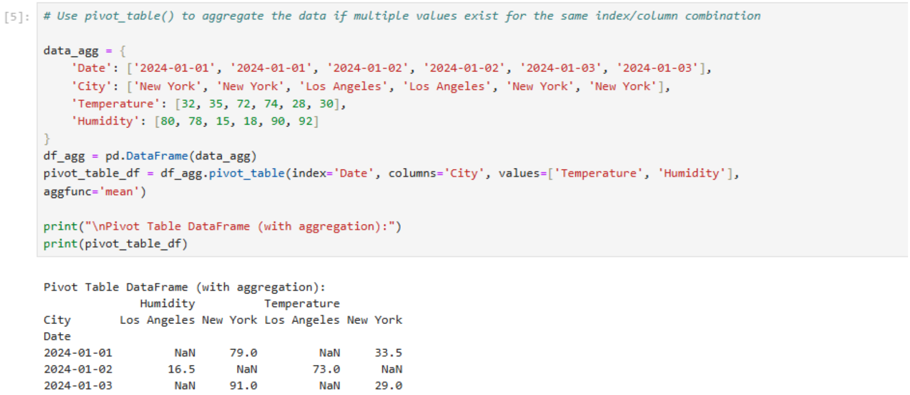
1. Pivoting Data Frame using pivot()



```
[4]: # Pivot the DataFrame to reshape it, setting 'Date' as index and 'City' as columns

     pivot_df = df.pivot(index='Date', columns='City', values=['Temperature', 'Humidity'])
     print("\nPivoted DataFrame:")
     print(pivot_df)
```

```
Pivoted DataFrame:
           Temperature              Humidity         
City       Los Angeles New York Los Angeles New York
Date                                                
2024-01-01          75       32          10       80
2024-01-02          72       30          15       85
2024-01-03          70       28          12       90
```

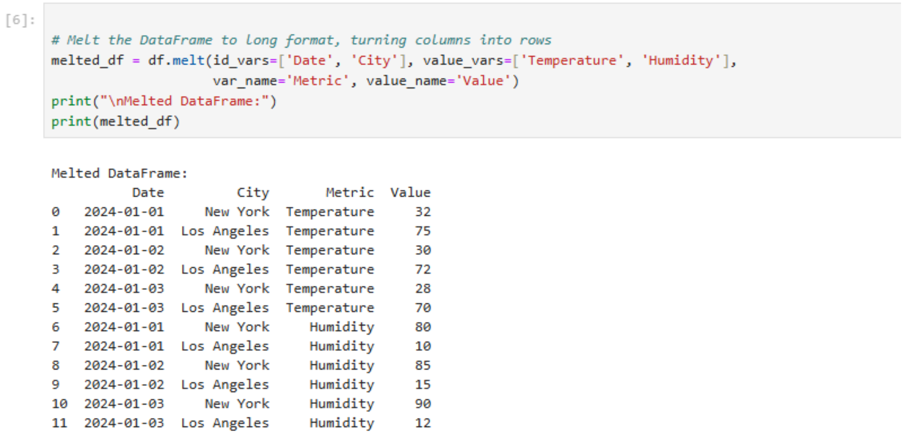
2. Pivoting DataFrame using pivot_table() with aggregation



```
[5]: # Use pivot_table() to aggregate the data if multiple values exist for the same index/column combination

     data_agg = {
         'Date': ['2024-01-01', '2024-01-01', '2024-01-02', '2024-01-02', '2024-01-03', '2024-01-03'],
         'City': ['New York', 'New York', 'Los Angeles', 'Los Angeles', 'New York', 'New York'],
         'Temperature': [32, 35, 72, 74, 28, 30],
         'Humidity': [80, 78, 15, 18, 90, 92]
     }
     df_agg = pd.DataFrame(data_agg)
     pivot_table_df = df_agg.pivot_table(index='Date', columns='City', values=['Temperature', 'Humidity'],
     aggfunc='mean')

     print("\nPivot Table DataFrame (with aggregation):")
     print(pivot_table_df)
```

```
Pivot Table DataFrame (with aggregation):
              Humidity              Temperature         
City       Los Angeles New York Los Angeles New York
Date                                                
2024-01-01         NaN     79.0         NaN     33.5
2024-01-02        16.5      NaN        73.0      NaN
2024-01-03         NaN     91.0         NaN     29.0
```

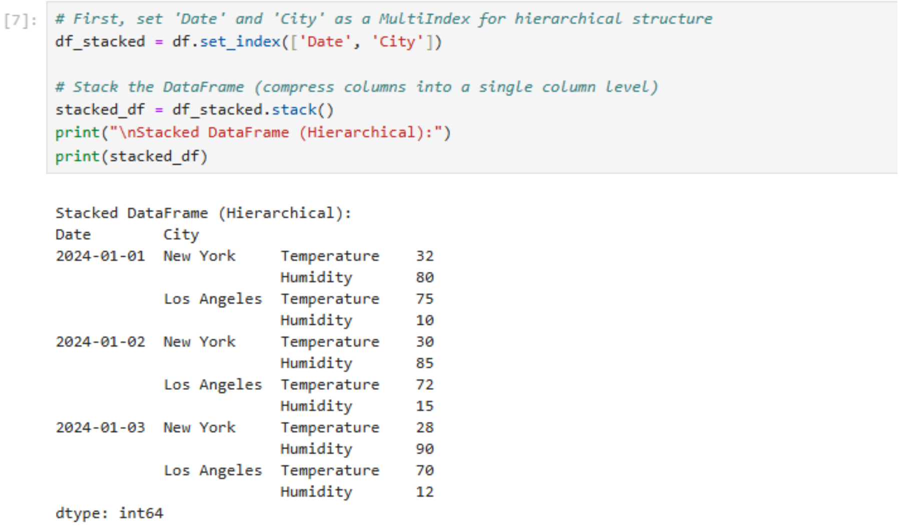
3. Melting DataFrame (Unpivoting)



```
[6]:
# Melt the DataFrame to long format, turning columns into rows
melted_df = df.melt(id_vars=['Date', 'City'], value_vars=['Temperature', 'Humidity'],
                    var_name='Metric', value_name='Value')
print("\nMelted DataFrame:")
print(melted_df)
```

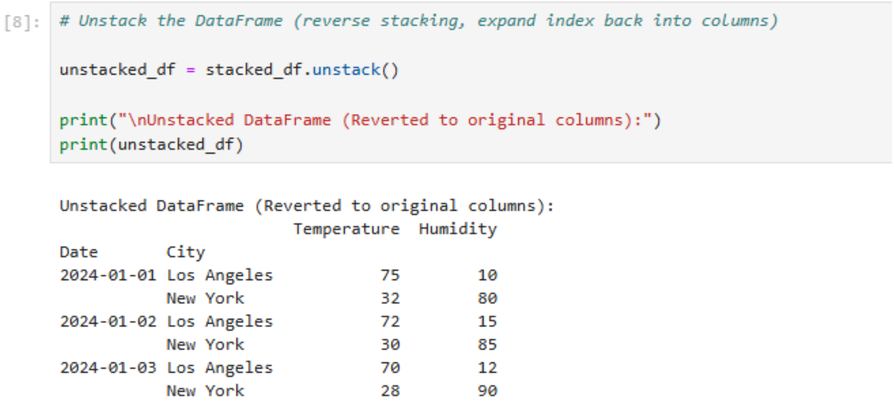
```
Melted DataFrame:
          Date         City       Metric  Value
0   2024-01-01     New York  Temperature     32
1   2024-01-01  Los Angeles  Temperature     75
2   2024-01-02     New York  Temperature     30
3   2024-01-02  Los Angeles  Temperature     72
4   2024-01-03     New York  Temperature     28
5   2024-01-03  Los Angeles  Temperature     70
6   2024-01-01     New York     Humidity     80
7   2024-01-01  Los Angeles     Humidity     10
8   2024-01-02     New York     Humidity     85
9   2024-01-02  Los Angeles     Humidity     15
10  2024-01-03     New York     Humidity     90
11  2024-01-03  Los Angeles     Humidity     12
```

1. Stacking and Unstacking DataFrame (Hierarchical Reshaping)



```
[7]:
# First, set 'Date' and 'City' as a MultiIndex for hierarchical structure
df_stacked = df.set_index(['Date', 'City'])

# Stack the DataFrame (compress columns into a single column level)
stacked_df = df_stacked.stack()
print("\nStacked DataFrame (Hierarchical):")
print(stacked_df)
```

```
Stacked DataFrame (Hierarchical):
Date        City
2024-01-01  New York     Temperature    32
                         Humidity       80
            Los Angeles  Temperature    75
                         Humidity       10
2024-01-02  New York     Temperature    30
                         Humidity       85
            Los Angeles  Temperature    72
                         Humidity       15
2024-01-03  New York     Temperature    28
                         Humidity       90
            Los Angeles  Temperature    70
                         Humidity       12
dtype: int64
```



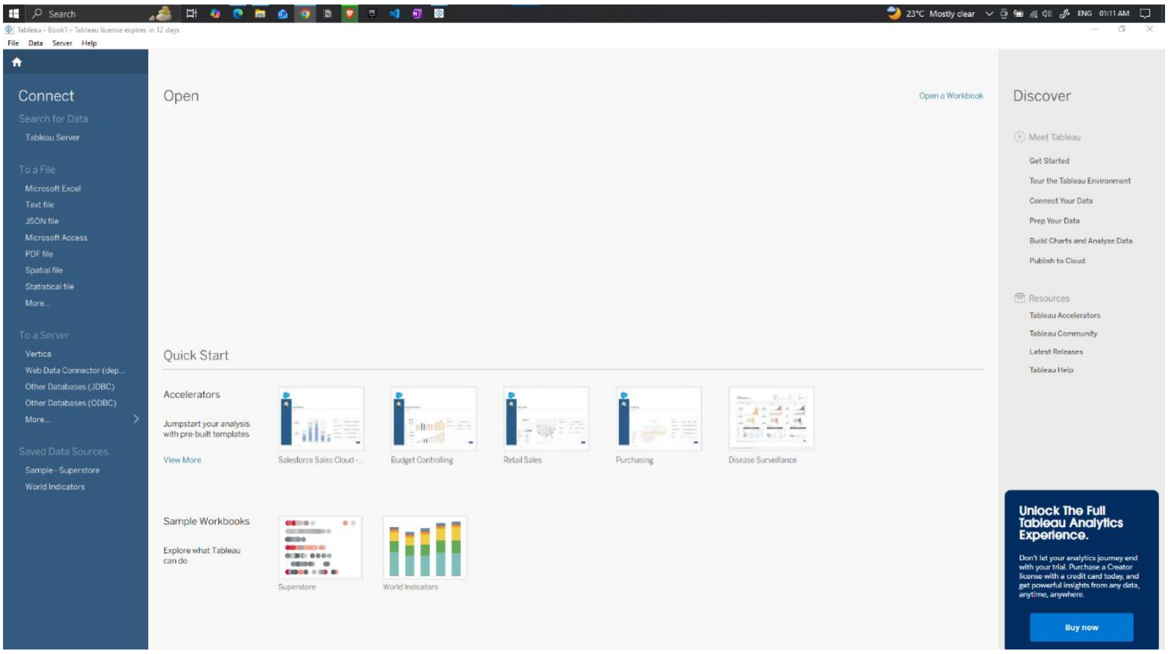
```
[8]:
# Unstack the DataFrame (reverse stacking, expand index back into columns)

unstacked_df = stacked_df.unstack()

print("\nUnstacked DataFrame (Reverted to original columns):")
print(unstacked_df)
```

```
Unstacked DataFrame (Reverted to original columns):
                        Temperature  Humidity
Date       City
2024-01-01 Los Angeles           75        10
           New York              32        80
2024-01-02 Los Angeles           72        15
           New York              30        85
2024-01-03 Los Angeles           70        12
           New York              28        90
```

Practical 6

Aim: Connecting and extracting with various data resources in tableau and Perform calculations and creating parameters in Tableau.

Step 1: Connect to Data Source 1. Open Tableau Desktop.
2. Click on "Connect" on the left-hand side and choose the desired data source:
For a local file: Choose Excel, CSV, or other file types. For a database: Choose from MySQL, PostgreSQL, or other supported databases.
3. Once connected, drag the required table or sheet into the Data Pane.
4. Preview the data to ensure the table contains relevant columns (e.g., Category, Sub-Category, Sales, etc.).



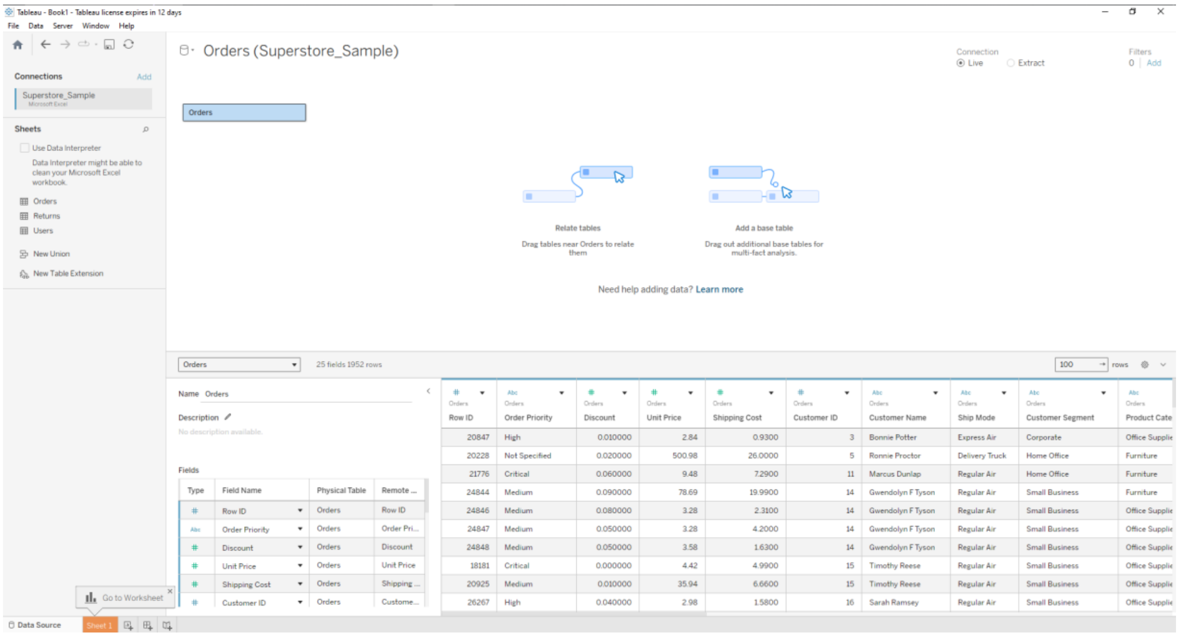


Tableau - Book1 - Tableau license expires in 12 days
File Data Server Window Help
Orders (Superstore_Sample)
Connection
Live
Extract
Filters
0 | Add
Connections Add
Superstore_Sample
Microsoft Excel
Sheets
Use Data Interpreter
Data Interpreter might be able to clean your Microsoft Excel workbook.
Orders
Returns
Users
New Union
New Table Extension
Orders
Relate tables
Drag tables near Orders to relate them
Add a base table
Drag out additional base tables for multi-fact analysis.
Need help adding data? Learn more
Go to Worksheet
Data Source
Sheet 1

Orders ▾ 25 fields 1952 rows

Name Orders

Description

No description available.

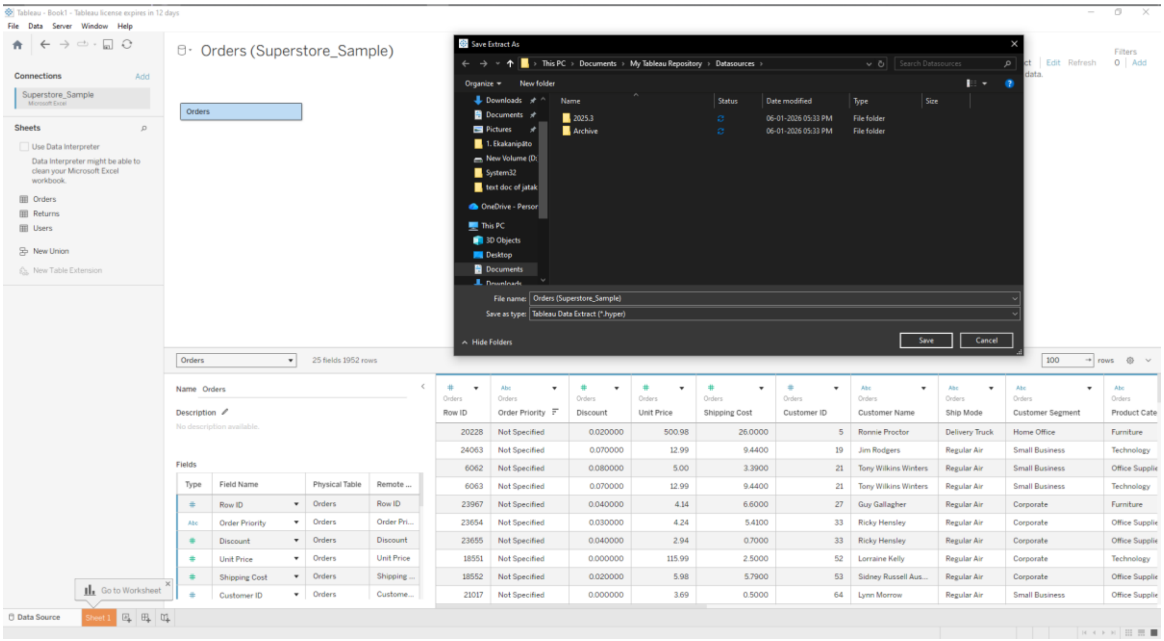
Fields

Type	Field Name	Physical Table	Remote ...
#	Row ID	Orders	Row ID
Abc	Order Priority	Orders	Order Pri...
#	Discount	Orders	Discount
#	Unit Price	Orders	Unit Price
#	Shipping Cost	Orders	Shipping ...
#	Customer ID	Orders	Custome...

100 rows

Orders Row ID	Orders Order Priority	Orders Discount	Orders Unit Price	Orders Shipping Cost	Orders Customer ID	Orders Customer Name	Orders Ship Mode	Orders Customer Segment	Orders Product Cate
20847	High	0.010000	2.84	0.9300	3	Bonnie Potter	Express Air	Corporate	Office Supplie
20228	Not Specified	0.020000	500.98	26.0000	5	Ronnie Proctor	Delivery Truck	Home Office	Furniture
21776	Critical	0.060000	9.48	7.2900	11	Marcus Dunlap	Regular Air	Home Office	Furniture
24844	Medium	0.090000	78.69	19.9900	14	Gwendolyn F Tyson	Regular Air	Small Business	Furniture
24846	Medium	0.080000	3.28	2.3100	14	Gwendolyn F Tyson	Regular Air	Small Business	Office Supplie
24847	Medium	0.050000	3.28	4.2000	14	Gwendolyn F Tyson	Regular Air	Small Business	Office Supplie
24848	Medium	0.050000	3.58	1.6300	14	Gwendolyn F Tyson	Regular Air	Small Business	Office Supplie
18181	Critical	0.000000	4.42	4.9900	15	Timothy Reese	Regular Air	Small Business	Office Supplie
20925	Medium	0.010000	35.94	6.6600	15	Timothy Reese	Regular Air	Small Business	Office Supplie
26267	High	0.040000	2.98	1.5800	16	Sarah Ramsey	Regular Air	Small Business	Office Supplie

Step 2: Extract Data 1. Go to the Data Source tab.
2. Click on the Extract button in the top-right corner.
3. Select the required columns to optimize performance (e.g., Category, Sub-Category, Sales, Profit). Save the extract file (.hyper) locally by clicking Extract.



Name	Status	Date modified	Type	Size
2025.3		06-01-2026 05:33 PM	File folder	
Archive		06-01-2026 05:33 PM	File folder	

File name: Orders (Superstore_Sample)

Save as type: Tableau Data Extract (*.hyper)

Orders — 25 fields 1952 rows — 100 rows

Name Orders

Description

No description available.

Fields

Type	Field Name	Physical Table	Remote ...
#	Row ID	Orders	Row ID
Abc	Order Priority	Orders	Order Pri...
#	Discount	Orders	Discount
#	Unit Price	Orders	Unit Price
#	Shipping Cost	Orders	Shipping ...
#	Customer ID	Orders	Custome...

Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	Customer Name	Ship Mode	Customer Segment	Product Cate
20228	Not Specified	0.020000	500.98	26.0000	5	Ronnie Proctor	Delivery Truck	Home Office	Furniture
24063	Not Specified	0.070000	12.99	9.4400	19	Jim Rodgers	Regular Air	Small Business	Technology
6062	Not Specified	0.080000	5.00	3.3900	21	Tony Wilkins Winters	Regular Air	Small Business	Office Supplie
6063	Not Specified	0.070000	12.99	9.4400	21	Tony Wilkins Winters	Regular Air	Small Business	Technology
23967	Not Specified	0.040000	4.14	6.6000	27	Guy Gallagher	Regular Air	Corporate	Furniture
23654	Not Specified	0.030000	4.24	5.4100	33	Ricky Hensley	Regular Air	Corporate	Office Supplie
23655	Not Specified	0.040000	2.94	0.7000	33	Ricky Hensley	Regular Air	Corporate	Office Supplie
18551	Not Specified	0.000000	115.99	2.5000	52	Lorraine Kelly	Regular Air	Corporate	Technology
18552	Not Specified	0.020000	5.98	5.7900	53	Sidney Russell Aus...	Regular Air	Corporate	Office Supplie
21017	Not Specified	0.000000	3.69	0.5000	64	Lynn Morrow	Regular Air	Small Business	Office Supplie

Step 3: Top N subcategories dynamic

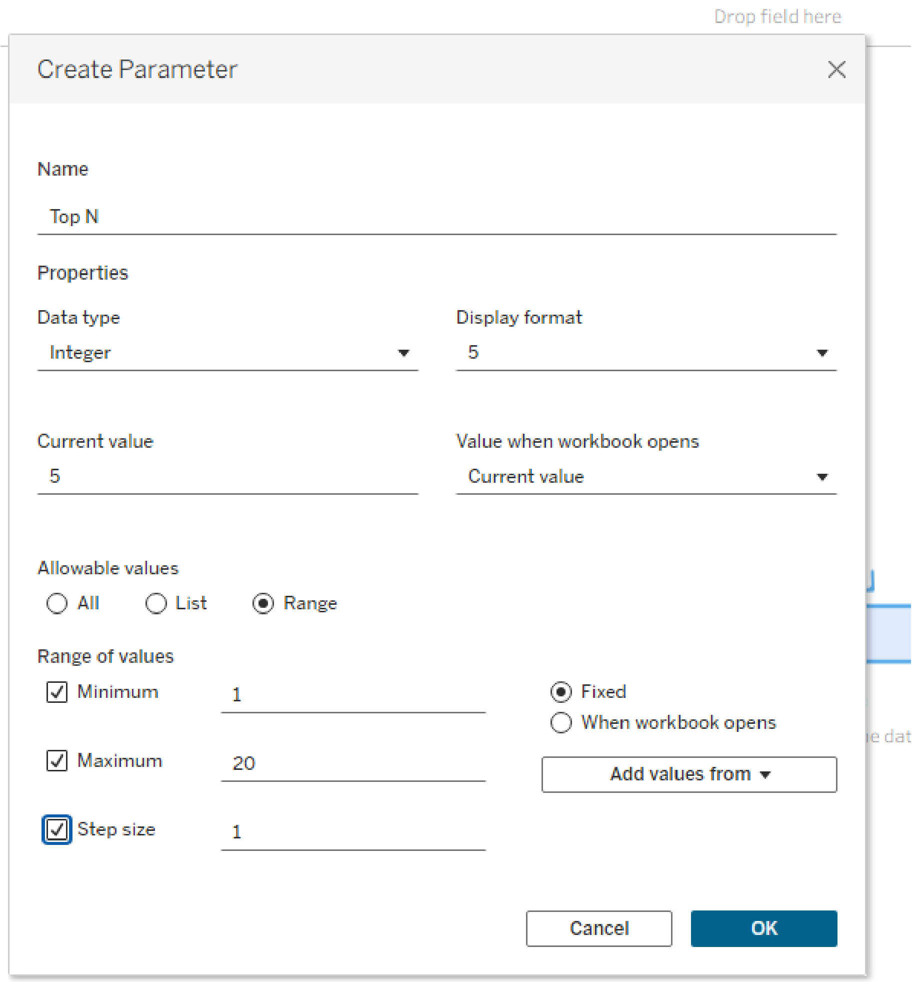
1. Create a parameter for N with current value 5:

2. Create a calculated field for ranking:

```
RANK(SUM([Sales]), 'desc')
```

2. Create another calculated field to filter the top N:

```
[Sales Rank] <= [Top N]
```



Create Parameter

Name: Top N

Properties

Data type: Integer

Display format: 5

Current value: 5

Value when workbook opens: Current value

Allowable values: ○ All ○ List ● Range

Range of values

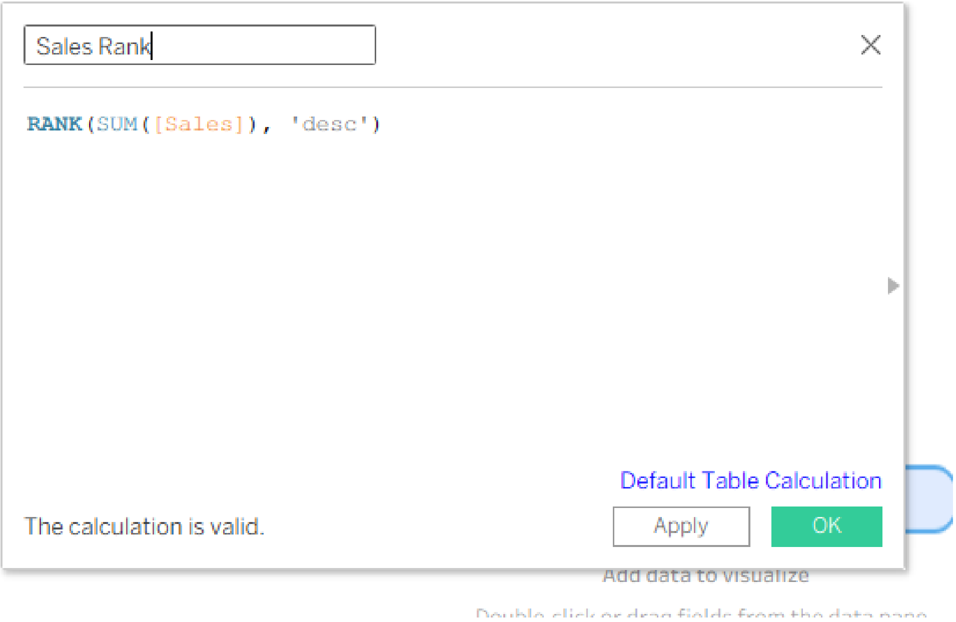
☑ Minimum 1

☑ Maximum 20

☑ Step size 1

● Fixed

○ When workbook opens



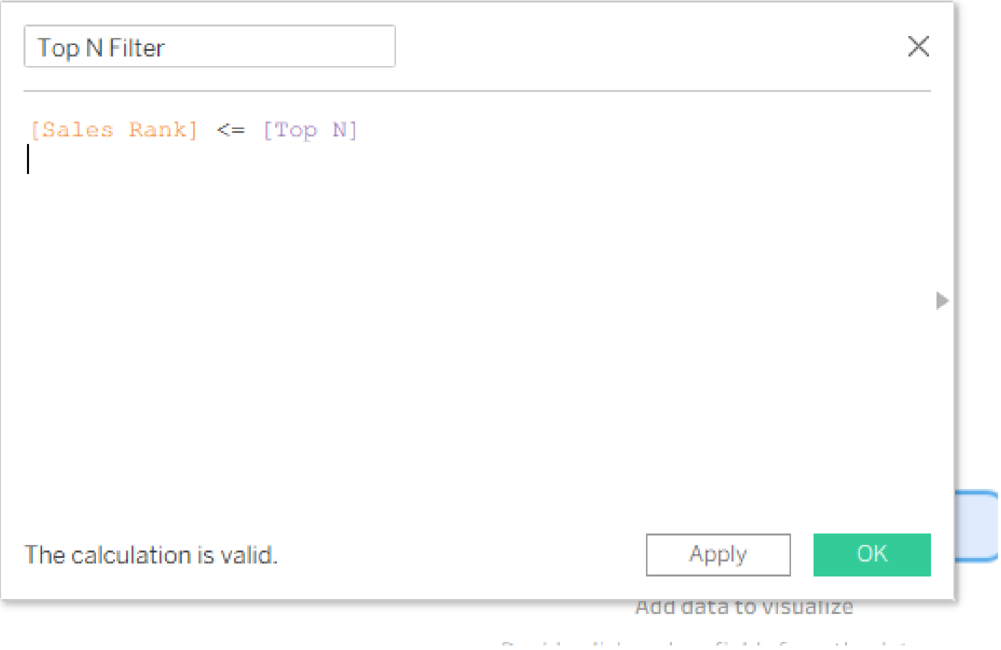
Sales Rank

```
RANK(SUM([Sales]), 'desc')
```

The calculation is valid.

Default Table Calculation
Apply
OK
Add data to visualize
Double click or drag fields from the data pane

Step 3.: Create Calculated Field to Filter Top N

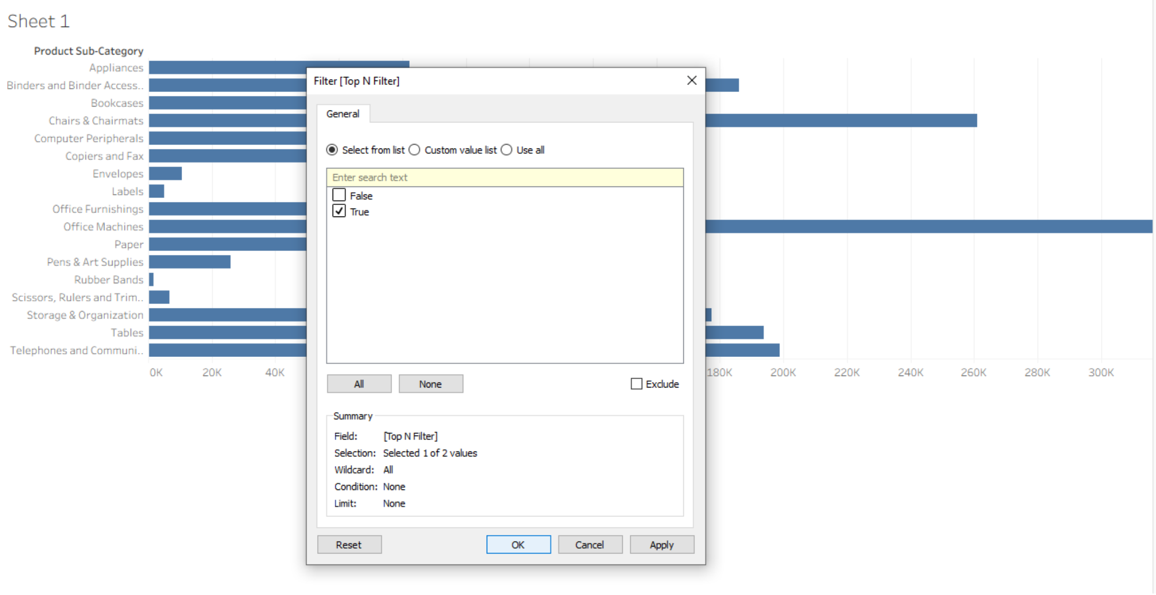


Top N Filter

```
[Sales Rank] <= [Top N]
```

The calculation is valid.

Step 3: Apply Top N Filter to Visualization



Filter [Top N Filter]

General

● Select from list ○ Custom value list ○ Use all

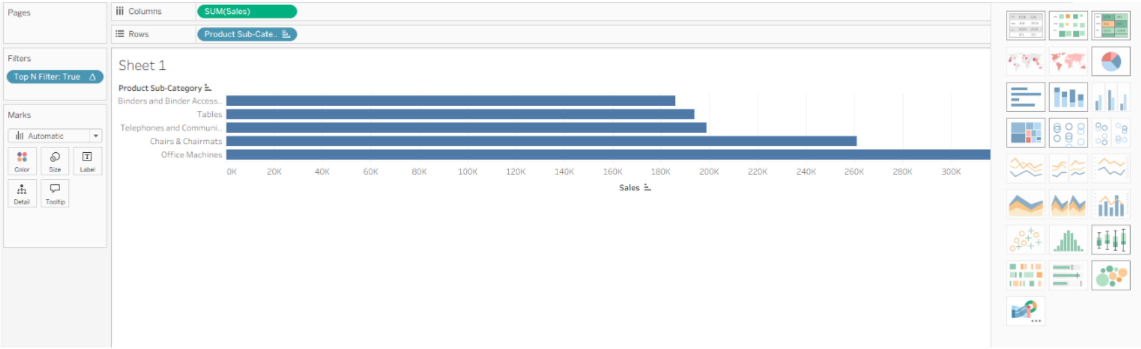
Enter search text

□ False
☑ True

□ Exclude

Summary

Field: [Top N Filter]
Selection: Selected 1 of 2 values
Wildcard: All
Condition: None
Limit: None

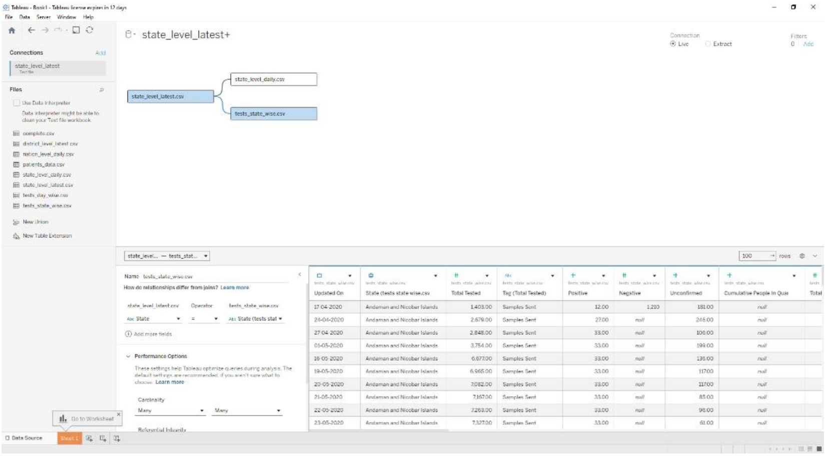


Columns: SUM(Sales)
Rows: Product Sub-Cate..
Filters: Top N Filter: True

Practical 7

Aim: Designing Tableau Dashboards for different displays and devices.

Integrated different dataset for making dashboard

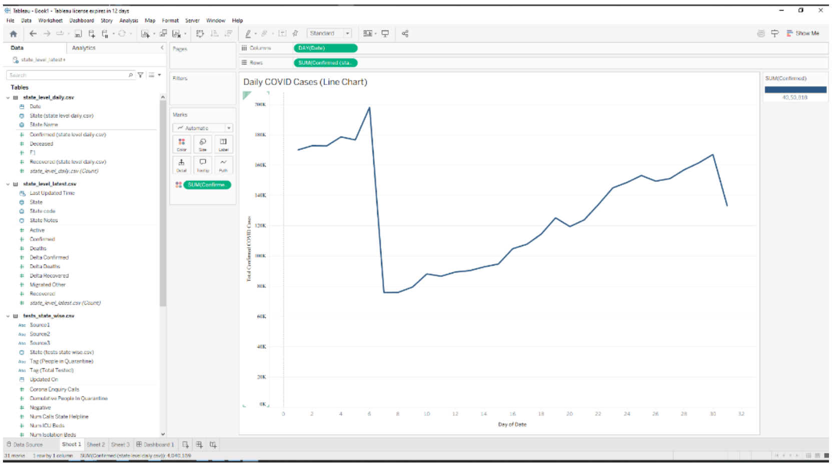


Updated On	State (tests state wise.csv	Total Tested	Tag (Total Tested)	Positive	Negative	Unconfirmed	Cumulative People In Quai	Total
17-04-2020	Andaman and Nicobar Islands	1,403.00	Samples Sent	12.00	1,210	181.00	null	
24-04-2020	Andaman and Nicobar Islands	2,679.00	Samples Sent	27.00	null	246.00	null	
27-04-2020	Andaman and Nicobar Islands	2,848.00	Samples Sent	33.00	null	106.00	null	
01-05-2020	Andaman and Nicobar Islands	3,754.00	Samples Sent	33.00	null	199.00	null	
16-05-2020	Andaman and Nicobar Islands	6,677.00	Samples Sent	33.00	null	136.00	null	
19-05-2020	Andaman and Nicobar Islands	6,965.00	Samples Sent	33.00	null	117.00	null	
20-05-2020	Andaman and Nicobar Islands	7,082.00	Samples Sent	33.00	null	117.00	null	
21-05-2020	Andaman and Nicobar Islands	7,167.00	Samples Sent	33.00	null	85.00	null	
22-05-2020	Andaman and Nicobar Islands	7,263.00	Samples Sent	33.00	null	96.00	null	
23-05-2020	Andaman and Nicobar Islands	7,327.00	Samples Sent	33.00	null	61.00	null	

Create multiple sheets for dashboard:

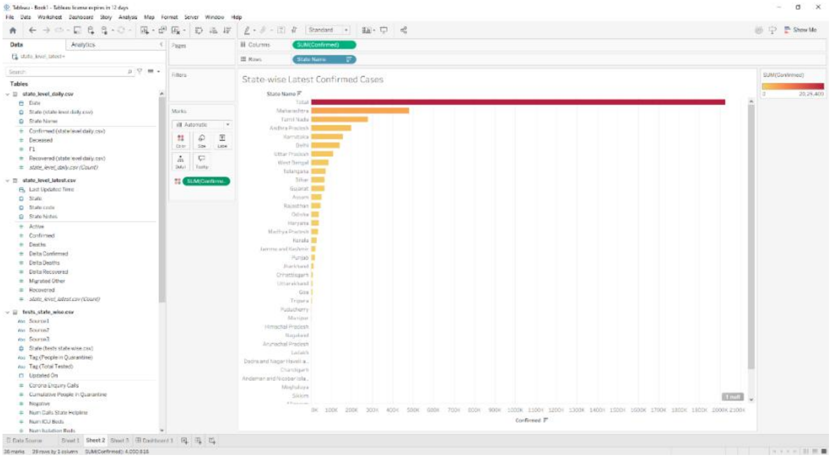
Sheet 1: Daily COVID Cases (Line Chart)

Purpose: To show Time-wise COVID case trend.

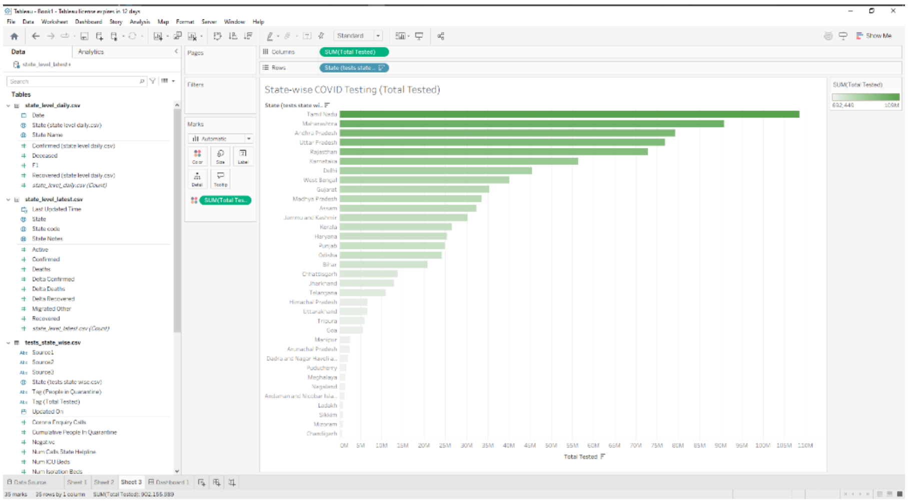


Columns: DAY(Date)
Rows: SUM(Confirmed (sta...
Daily COVID Cases (Line Chart)
Total Confirmed COVID Cases
Day of Date
SUM(Confirmed)
Sheet 1
Sheet 2
Sheet 3
Dashboard 1

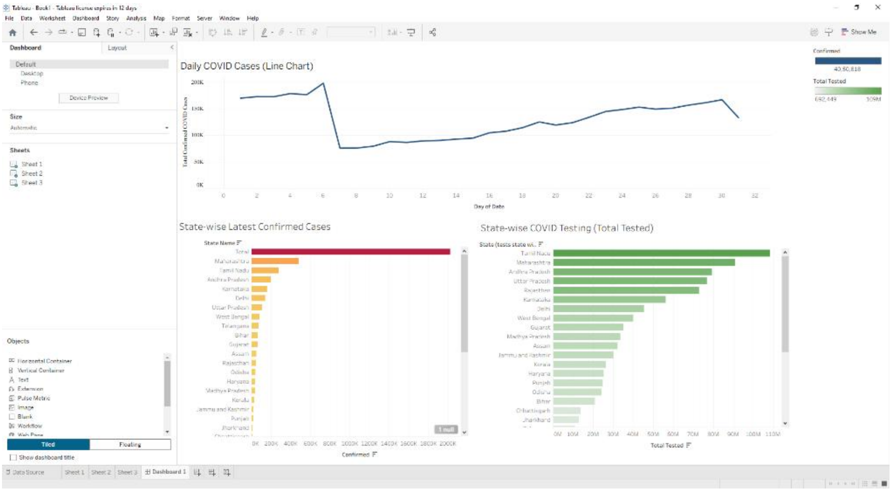
Sheet 2: State-wise Latest Confirmed Cases (Bar Chart)



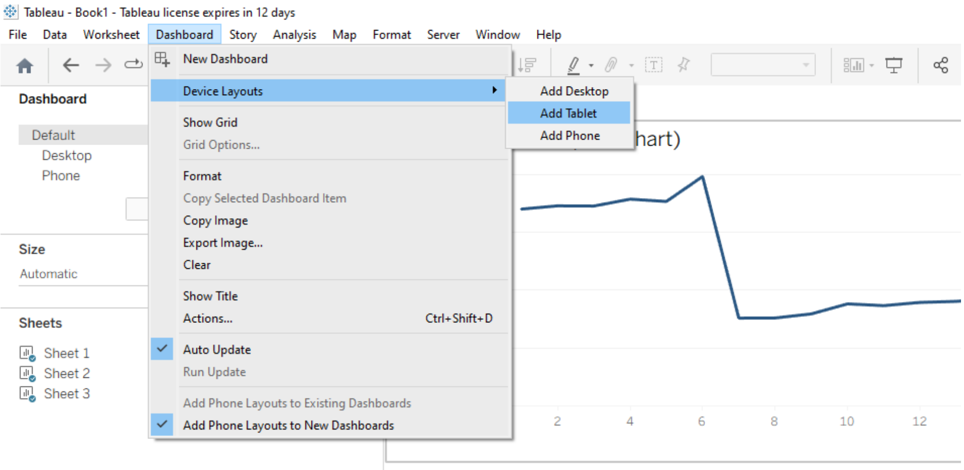
Sheet 3: State-wise COVID Testing (Bar Chart)



Combine all three sheets for dashboard


Daily COVID Cases (Line Chart)
State-wise Latest Confirmed Cases
State-wise COVID Testing (Total Tested)

For selecting the dashboard into different layout go to the dashboard option in the navigator bar and select device layout



Dashboard in Desktop layout

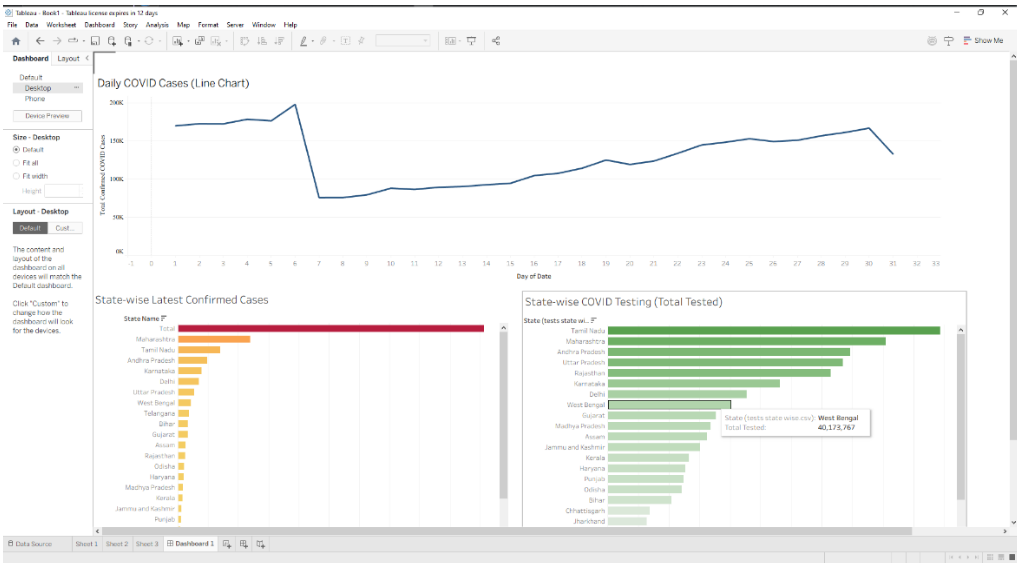


Tableau - Book1 - Tableau license expires in 12 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Dashboard Layout
Default
Desktop
Phone
Device Preview
Size - Desktop
Default
Fit all
Fit width
Layout - Desktop
Default Cust...
The content and layout of the dashboard on all devices will match the Default dashboard.
Click "Custom" to change how the dashboard will look for the devices.
Daily COVID Cases (Line Chart)
Total Confirmed COVID Cases
Day of Date
State-wise Latest Confirmed Cases
State Name
Total
Maharashtra
Tamil Nadu
Andhra Pradesh
Karnataka
Delhi
Uttar Pradesh
West Bengal
Telangana
Bihar
Gujarat
Assam
Rajasthan
Odisha
Haryana
Madhya Pradesh
Kerala
Jammu and Kashmir
Punjab
State-wise COVID Testing (Total Tested)
State (tests state wi...
State (tests state wise csv): West Bengal
Total Tested: 40,173,767
Data Source Sheet 1 Sheet 2 Sheet 3 Dashboard 1
Show Me

Dashboard in Tablet layout

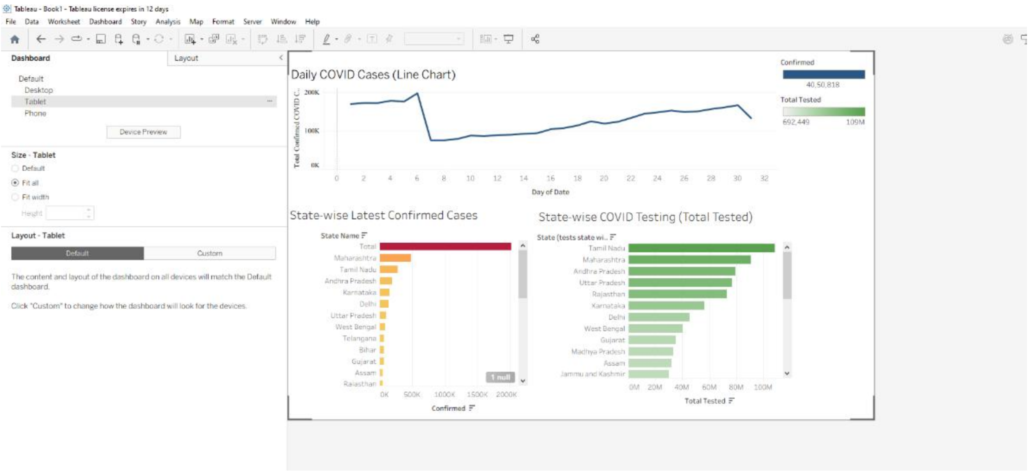
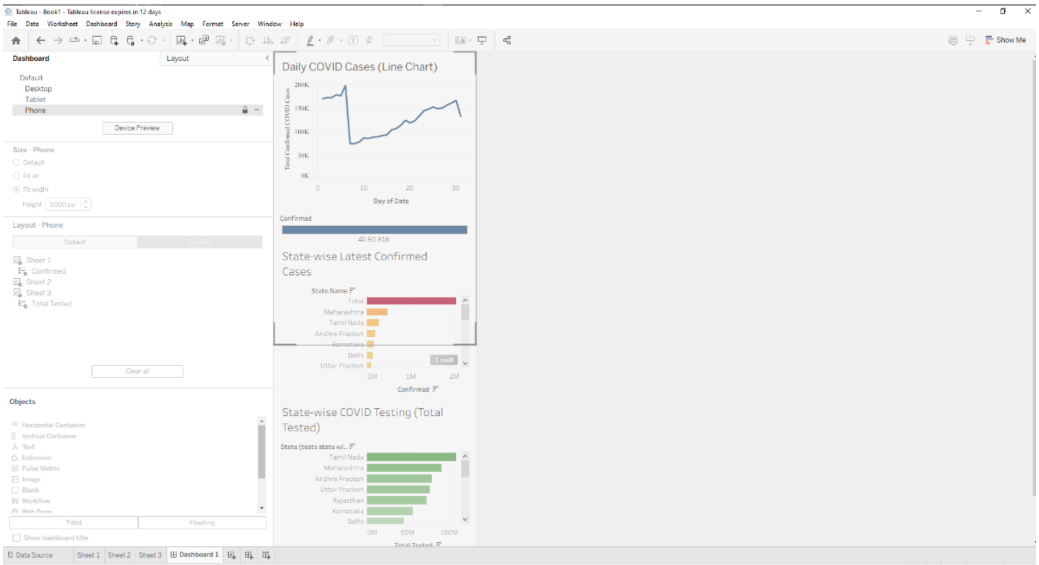


Tableau - Book1 - Tableau license expires in 12 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Dashboard Layout
Default
Desktop
Tablet
Phone
Device Preview
Size - Tablet
Default
Fit all
Fit width
Height
Layout - Tablet
Default Custom
The content and layout of the dashboard on all devices will match the Default dashboard.
Click "Custom" to change how the dashboard will look for the devices.
Daily COVID Cases (Line Chart)
Confirmed
40,50,818
Total Tested
692,449 109M
Day of Date
State-wise Latest Confirmed Cases
State Name
Confirmed
State-wise COVID Testing (Total Tested)
State (tests state wi...
Total Tested

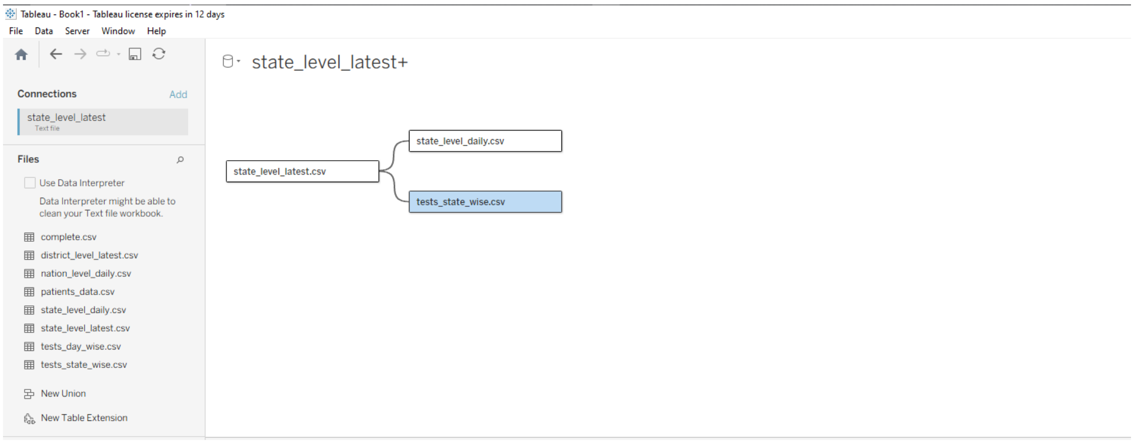
Dashboard in phone layout



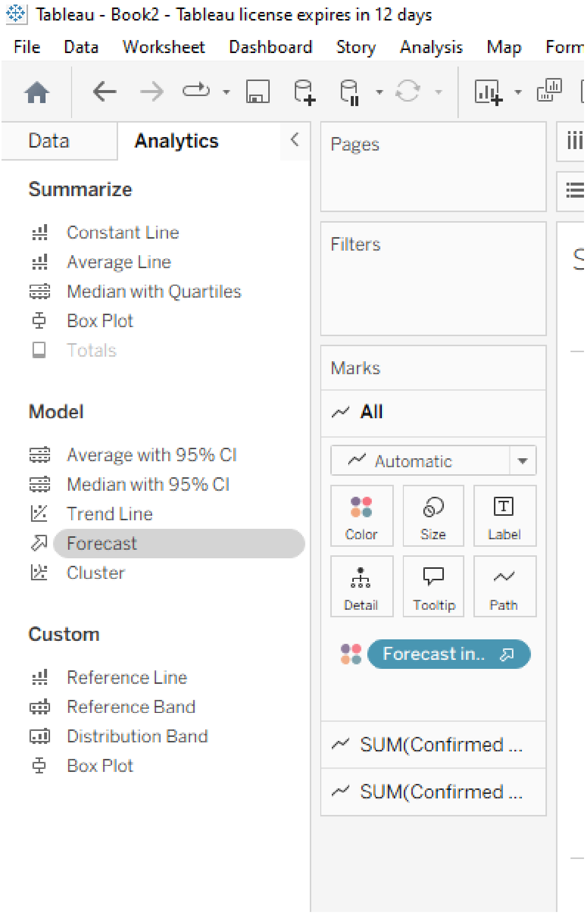
Practical 8

Aim: Create a Trend model using data, Analyse-it and use it for forecasting.

Open Tableau and integrate the dataset of covid 19 cases and create sheet for Forecasting of Positive cases



Select all the filed from the data plan and for the network mode go to analytics and select furcating option



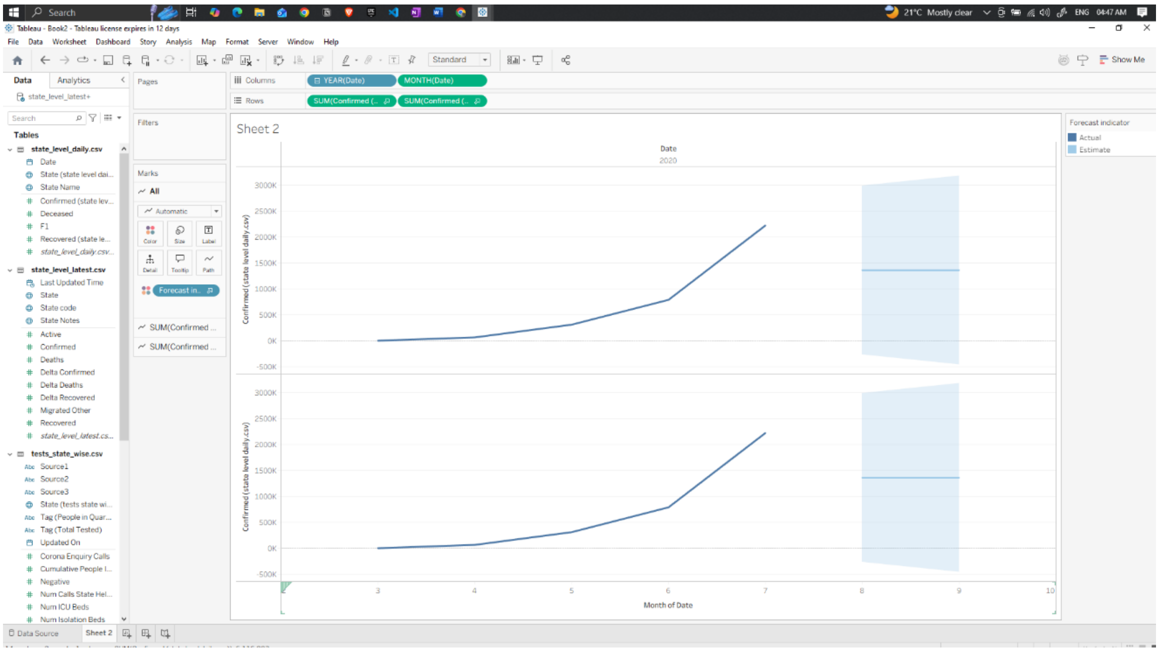


Tableau - Book2 - Tableau license expires in 12 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
state_level_latest+
Search
Tables
state_level_daily.csv
Date
State (state level dai...
State Name
Confirmed (state lev...
Deceased
F1
Recovered (state le...
state_level_daily.csv...
state_level_latest.csv
Last Updated Time
State
State code
State Notes
Active
Confirmed
Deaths
Delta Confirmed
Delta Deaths
Delta Recovered
Migrated Other
Recovered
state_level_latest.cs...
tests_state_wise.csv
Source1
Source2
Source3
State (tests state wi...
Tag (People in Quar...
Tag (Total Tested)
Updated On
Corona Enquiry Calls
Cumulative People I...
Negative
Num Calls State Hel...
Num ICU Beds
Num Isolation Beds
Pages
Filters
Marks
All
Automatic
Color
Size
Label
Detail
Tooltip
Path
Forecast in..
SUM(Confirmed ...
SUM(Confirmed ...
Columns
YEAR(Date)
MONTH(Date)
Rows
SUM(Confirmed (..
SUM(Confirmed (..
Sheet 2
Date
2020
Confirmed (state level daily.csv)
3000K
2500K
2000K
1500K
1000K
500K
0K
-500K
3
4
5
6
7
8
9
10
Month of Date
Forecast indicator
Actual
Estimate
Data Source
Sheet 2

Practical 9

Aim: Creating Geospatial feature maps in Tableau using Geospatial Data.

Open the tableau and integrate sample-superstore dataset into the data panel

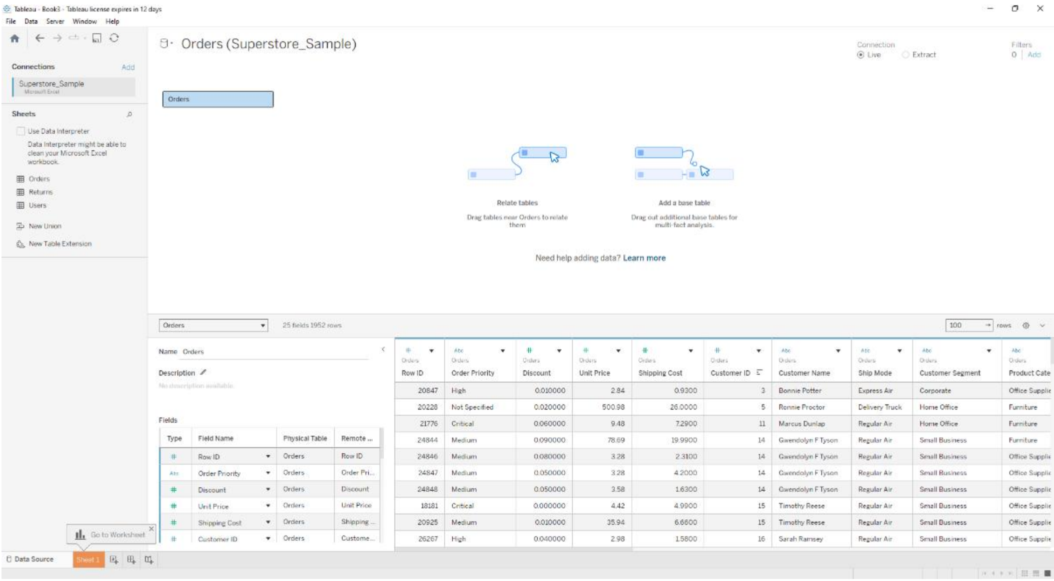


Tableau - Book3 - Tableau license expires in 12 days
File Data Server Window Help
Orders (Superstore_Sample)
Connections
Add
Superstore_Sample
Microsoft Excel
Sheets
Use Data Interpreter
Data Interpreter might be able to clean your Microsoft Excel workbook.
Orders
Returns
Users
New Union
New Table Extension
Orders
Relate tables
Drag tables near Orders to relate them
Add a base table
Drag out additional base tables for multi fact analysis.
Need help adding data? Learn more
Connection
Live
Extract
Filters
0 | Add
Go to Worksheet
Data Source
Sheet 1

Orders 25 fields 1952 rows

Name Orders

Description

Fields

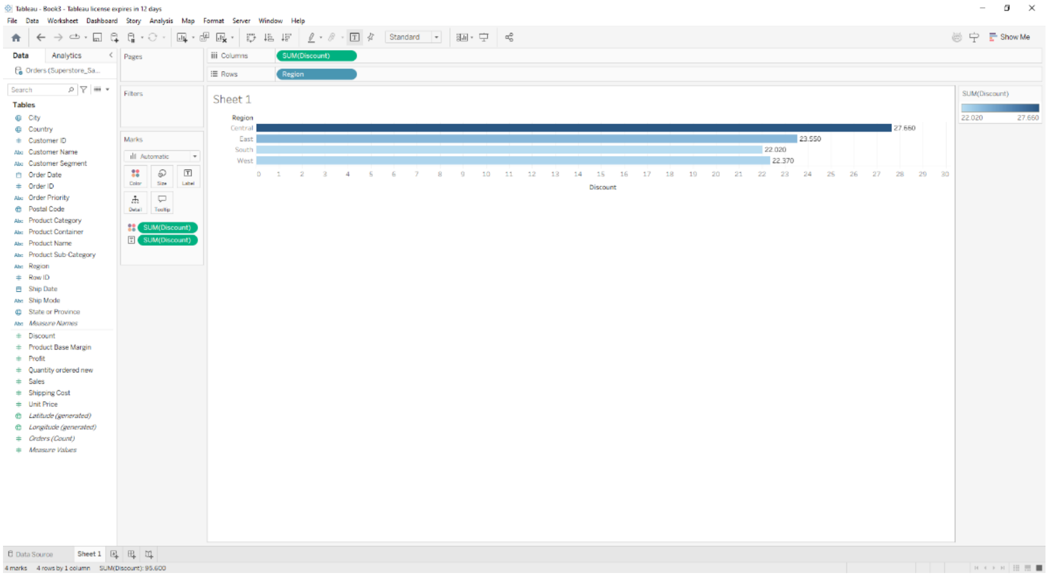
Type	Field Name	Physical Table	Remote ...
#	Row ID	Orders	Row ID
Abc	Order Priority	Orders	Order Pri...
#	Discount	Orders	Discount
#	Unit Price	Orders	Unit Price
#	Shipping Cost	Orders	Shipping ...
#	Customer ID	Orders	Custome...

100 rows

Row ID	Order Priority	Discount	Unit Price	Shipping Cost	Customer ID	Customer Name	Ship Mode	Customer Segment	Product Cate
20847	High	0.010000	2.84	0.9300	3	Bonnie Potter	Express Air	Corporate	Office Supplie
20228	Not Specified	0.020000	500.98	26.0000	5	Ronnie Proctor	Delivery Truck	Home Office	Furniture
21776	Critical	0.060000	9.48	7.2900	11	Marcus Dunlap	Regular Air	Home Office	Furniture
24844	Medium	0.090000	78.69	19.9900	14	Gwendolyn F Tyson	Regular Air	Small Business	Furniture
24846	Medium	0.080000	3.28	2.3100	14	Gwendolyn F Tyson	Regular Air	Small Business	Office Supplie
24847	Medium	0.050000	3.28	4.2000	14	Gwendolyn F Tyson	Regular Air	Small Business	Office Supplie
24848	Medium	0.050000	3.58	1.6300	14	Gwendolyn F Tyson	Regular Air	Small Business	Office Supplie
18181	Critical	0.000000	4.42	4.9900	15	Timothy Reese	Regular Air	Small Business	Office Supplie
20925	Medium	0.010000	35.94	6.6600	15	Timothy Reese	Regular Air	Small Business	Office Supplie
26267	High	0.040000	2.98	1.5800	16	Sarah Ramsey	Regular Air	Small Business	Office Supplie

Drag the 'Discount' field into the Columns section, the 'Region' field into the Rows section, and add the sum of 'Discount' into both the Color and Label sections.

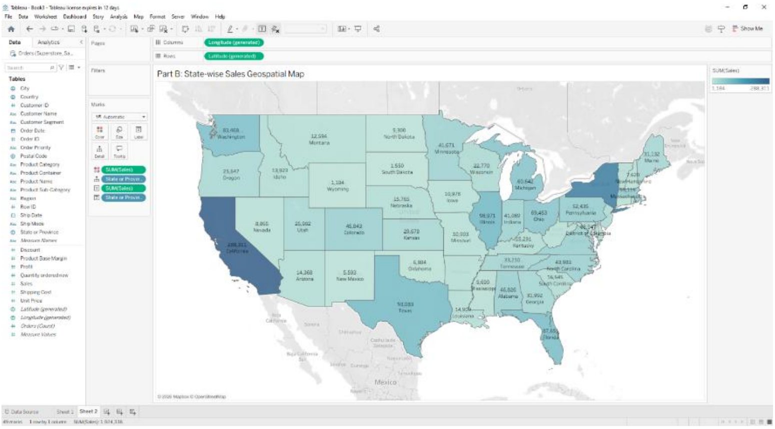
Part A: Region-wise Discount Visualization



Part B: State-wise Sales Geospatial Map

Open a new sheet. Create a state-wise sales map by following these steps:

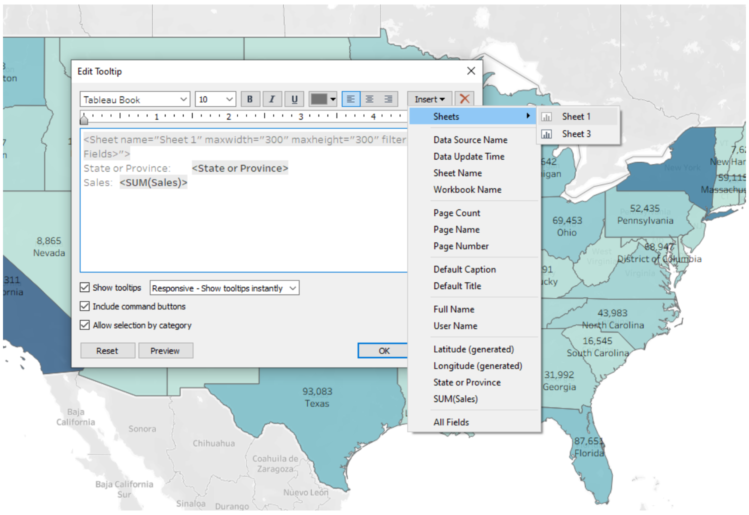
- Drag the 'Longitude' field into the Columns section.
- Drag the 'Latitude' field into the Rows section.
- Drag the sum of 'Sales' into both the Color and Label sections.
- Drag the 'State' field into both the Detail and Label sections.



Part C: Integrating Region-wise Discounts using Tooltips

To integrate region-wise discounts into the map graph using tooltips, follow these steps:

1. Select the Tooltips option in the Marks section.
2. Click on the Insert option.
3. Select the region-wise discount sheet for integration. This should help you display the region-wise discounts when you hover over the map.



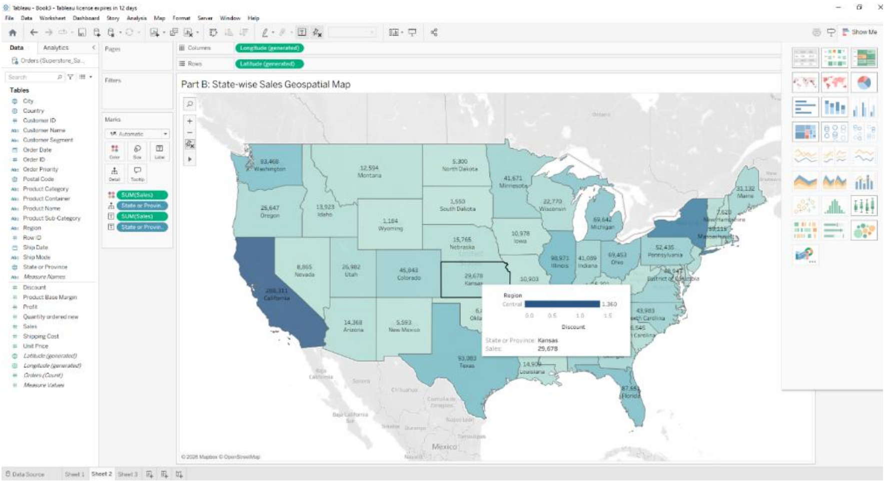
Edit Tooltip

```
<Sheet name="Sheet 1" maxwidth="300" maxheight="300" filter
Fields>">
State or Province:   <State or Province>
Sales:  <SUM(Sales)>
```

☑ Show tooltips   Responsive - Show tooltips instantly

☑ Include command buttons

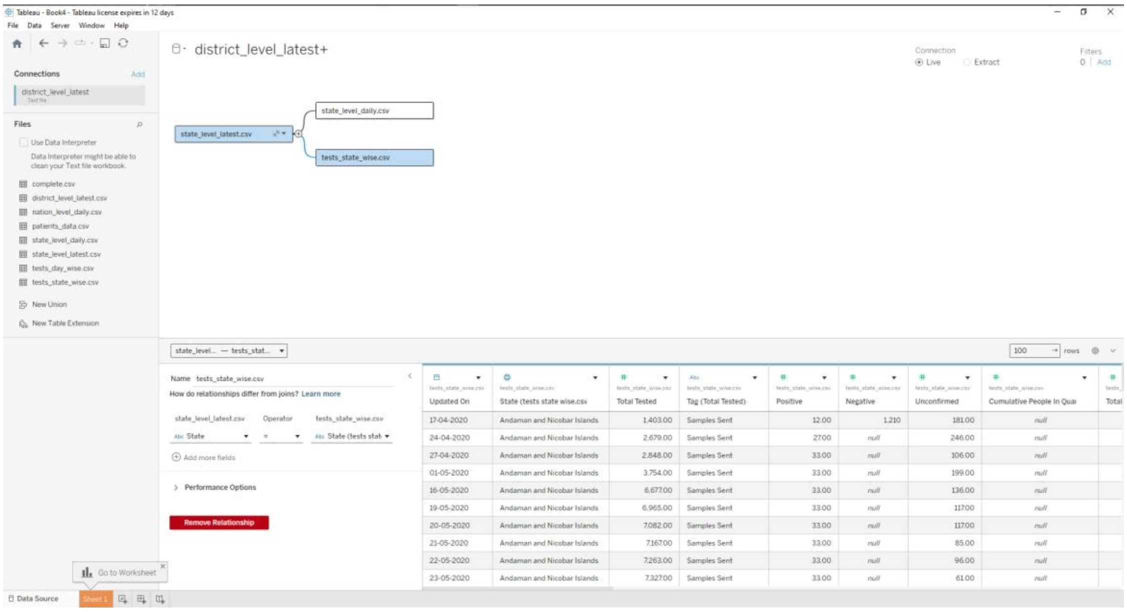
☑ Allow selection by category



Practical 10

Aim: Create Dashboard and Storytelling using tableau.

Open Tableau and integrate the dataset of covid 19 cases and create sheet for Forecasting of Positive cases



Name tests_state_wise.csv

How do relationships differ from joins? Learn more

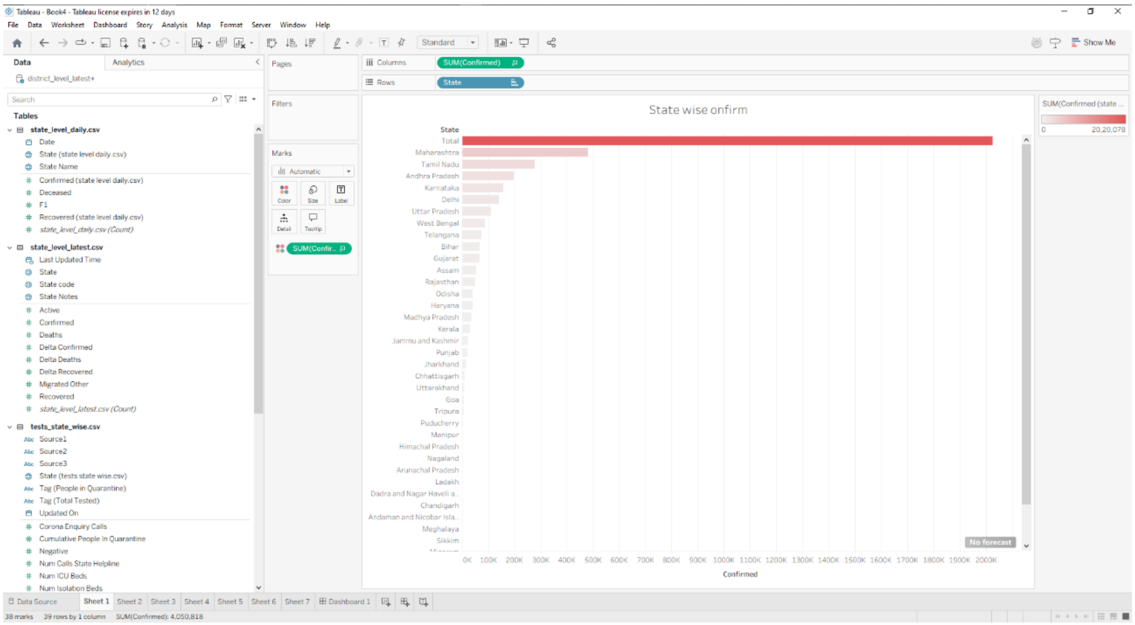
state_level_latest.csv	Operator	tests_state_wise.csv
State	=	State (tests stat

Add more fields

100 rows

Updated On	State (tests state wise.csv	Total Tested	Tag (Total Tested)	Positive	Negative	Unconfirmed	Cumulative People In Qua	Total
17-04-2020	Andaman and Nicobar Islands	1,403.00	Samples Sent	12.00	1,210	181.00	null	
24-04-2020	Andaman and Nicobar Islands	2,679.00	Samples Sent	27.00	null	246.00	null	
27-04-2020	Andaman and Nicobar Islands	2,848.00	Samples Sent	33.00	null	106.00	null	
01-05-2020	Andaman and Nicobar Islands	3,754.00	Samples Sent	33.00	null	199.00	null	
16-05-2020	Andaman and Nicobar Islands	6,677.00	Samples Sent	33.00	null	136.00	null	
19-05-2020	Andaman and Nicobar Islands	6,965.00	Samples Sent	33.00	null	117.00	null	
20-05-2020	Andaman and Nicobar Islands	7,082.00	Samples Sent	33.00	null	117.00	null	
21-05-2020	Andaman and Nicobar Islands	7,167.00	Samples Sent	33.00	null	85.00	null	
22-05-2020	Andaman and Nicobar Islands	7,263.00	Samples Sent	33.00	null	96.00	null	
23-05-2020	Andaman and Nicobar Islands	7,327.00	Samples Sent	33.00	null	61.00	null	

Create a multiple sheet reports like State wise confirmed covid cases and charts, Sate Wise Cured Covid Cases, Sate wise Covid Death cases, State wise Positive and Negative cases State wise positive Graph etc …



Columns: SUM(Confirmed)

Rows: State

38 marks 39 rows by 1 column SUM(Confirmed): 4,050,818

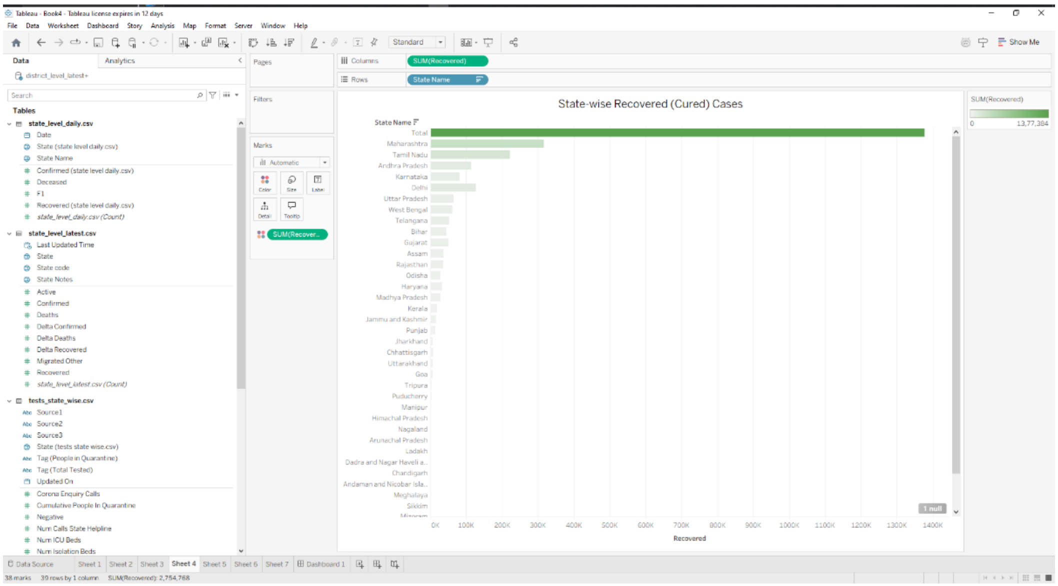


Tableau - Book4 - Tableau license expires in 12 days
Columns: SUM(Recovered)
Rows: State Name
State-wise Recovered (Cured) Cases
State Name
Total
Maharashtra
Tamil Nadu
Andhra Pradesh
Karnataka
Delhi
Uttar Pradesh
West Bengal
Telangana
Bihar
Gujarat
Assam
Rajasthan
Odisha
Haryana
Madhya Pradesh
Kerala
Jammu and Kashmir
Punjab
Jharkhand
Chhattisgarh
Uttarakhand
Goa
Tripura
Puducherry
Manipur
Himachal Pradesh
Nagaland
Arunachal Pradesh
Ladakh
Dadra and Nagar Haveli a..
Chandigarh
Andaman and Nicobar Isla..
Meghalaya
Sikkim
Mizoram
0K 100K 200K 300K 400K 500K 600K 700K 800K 900K 1000K 1100K 1200K 1300K 1400K
Recovered
SUM(Recovered)
0 13,77,384
1 null
38 marks 39 rows by 1 column SUM(Recovered): 2,754,768

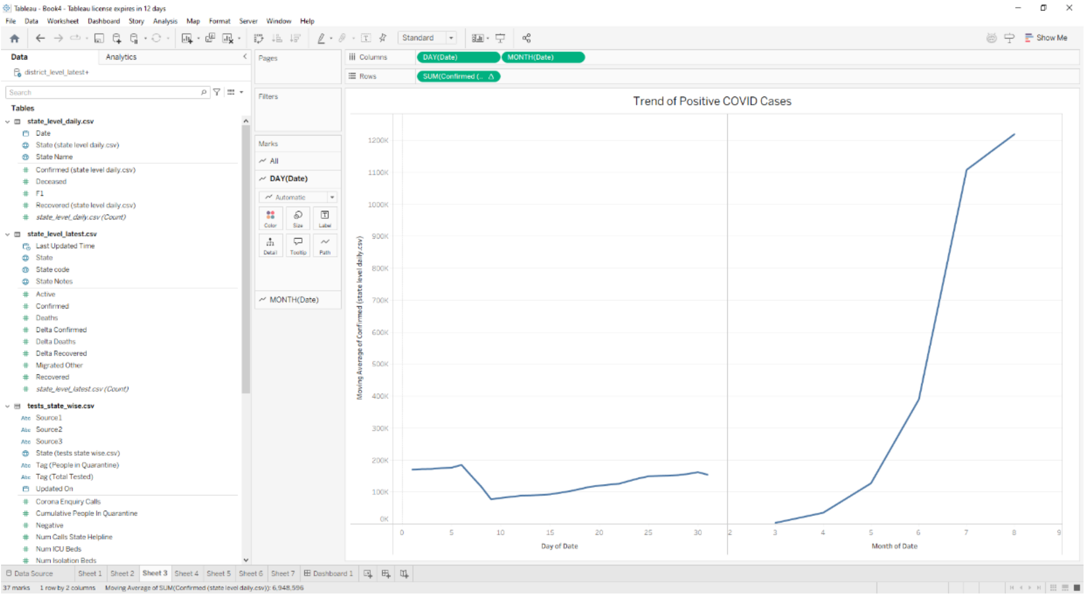


Tableau - Book4 - Tableau license expires in 12 days
Columns: DAY(Date) MONTH(Date)
Rows: SUM(Confirmed (... Δ
Trend of Positive COVID Cases
Moving Average of Confirmed (state level daily.csv)
0K 100K 200K 300K 400K 500K 600K 700K 800K 900K 1000K 1100K 1200K
0 5 10 15 20 25 30
Day of Date
2 3 4 5 6 7 8 9
Month of Date
37 marks 1 row by 2 columns Moving Average of SUM(Confirmed (state level daily.csv)): 6,948,996

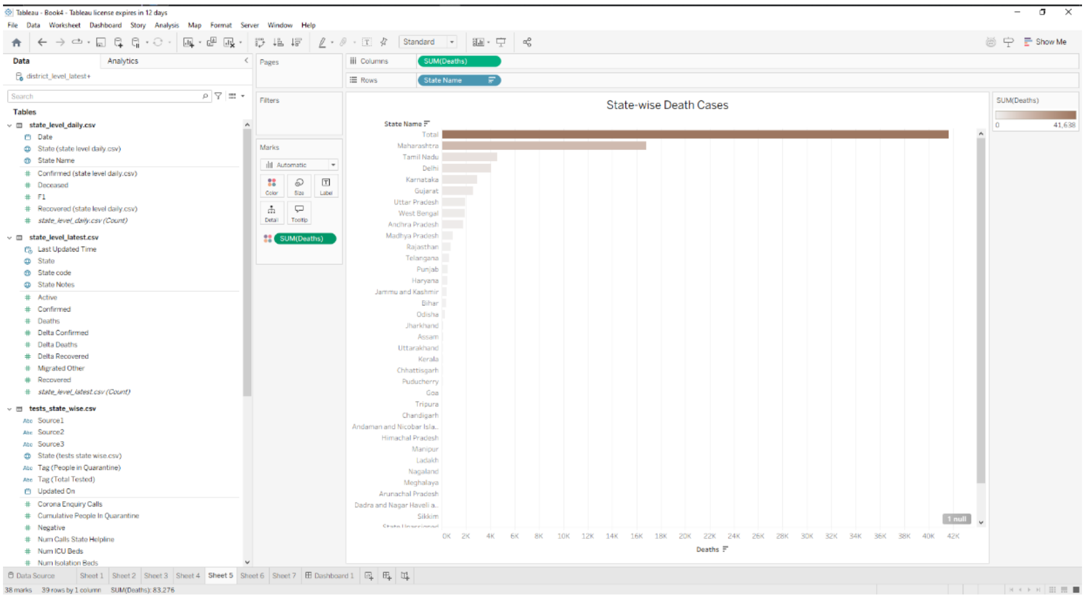


Tableau - Book4 - Tableau license expires in 12 days
Columns: SUM(Deaths)
Rows: State Name
State-wise Death Cases
State Name
Total
Maharashtra
Tamil Nadu
Delhi
Karnataka
Gujarat
Uttar Pradesh
West Bengal
Andhra Pradesh
Madhya Pradesh
Rajasthan
Telangana
Punjab
Haryana
Jammu and Kashmir
Bihar
Odisha
Jharkhand
Assam
Uttarakhand
Kerala
Chhattisgarh
Puducherry
Goa
Tripura
Chandigarh
Andaman and Nicobar Isla..
Himachal Pradesh
Manipur
Ladakh
Nagaland
Meghalaya
Arunachal Pradesh
Dadra and Nagar Haveli a..
Sikkim
State Unassigned
0K 2K 4K 6K 8K 10K 12K 14K 16K 18K 20K 22K 24K 26K 28K 30K 32K 34K 36K 38K 40K 42K
Deaths
SUM(Deaths)
0 41,638
1 null
38 marks 39 rows by 1 column SUM(Deaths): 83,276

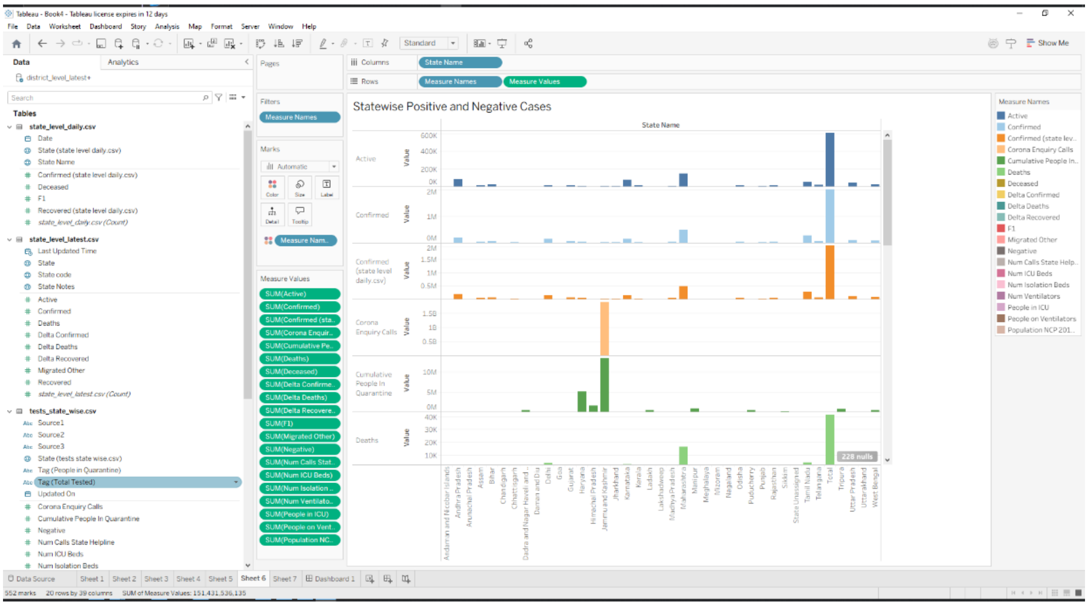


Tableau - Book4 - Tableau license expires in 12 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Data
Analytics
district_level_latest+
Tables
state_level_daily.csv
state_level_latest.csv
tests_state_wise.csv
Pages
Filters
Measure Names
Marks
Measure Values
Columns
State Name
Rows
Measure Names
Measure Values
Statewise Positive and Negative Cases
State Name
Active
Confirmed
Confirmed (state level daily.csv)
Corona Enquiry Calls
Cumulative People In Quarantine
Deaths
Value
228 nulls
Measure Names
Sheet 1 Sheet 2 Sheet 3 Sheet 4 Sheet 5 Sheet 6 Sheet 7 Dashboard 1
552 marks 20 rows by 39 columns SUM of Measure Values: 151,431,536,135

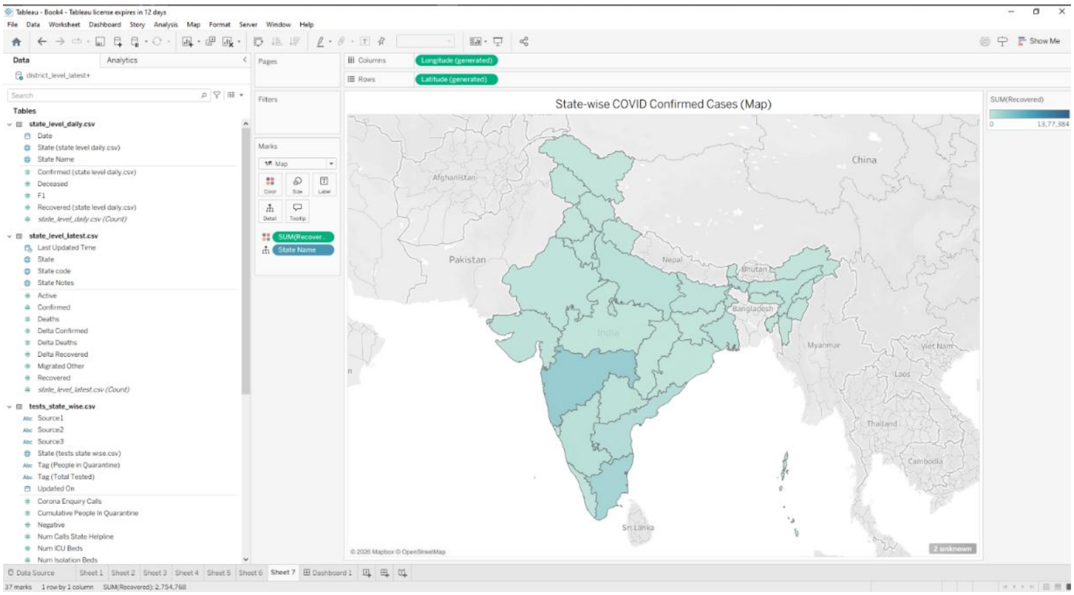


Tableau - Book4 - Tableau license expires in 12 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Columns
Longitude (generated)
Rows
Latitude (generated)
State-wise COVID Confirmed Cases (Map)
SUM(Recovered)
0
13,77,384
Afghanistan
China
Pakistan
Nepal
Bhutan
Bangladesh
India
Myanmar
Viet Nam
Laos
Thailand
Cambodia
Sri Lanka
© 2026 Mapbox © OpenStreetMap
2 unknown
SUM(Recovered)
State Name
Sheet 1 Sheet 2 Sheet 3 Sheet 4 Sheet 5 Sheet 6 Sheet 7 Dashboard 1
37 marks 1 row by 1 column SUM(Recovered): 2,754,768

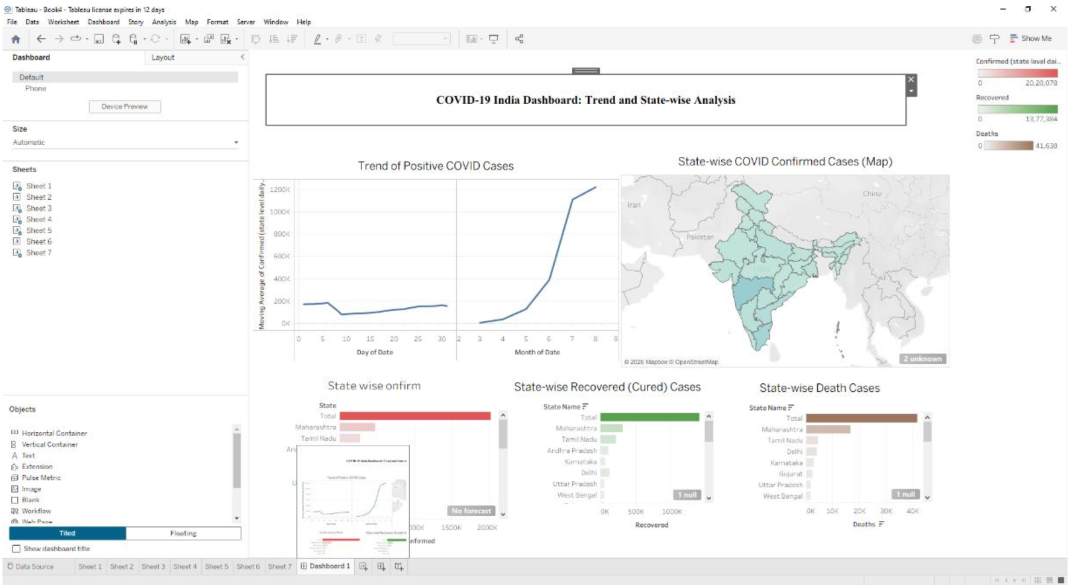


Tableau - Book4 - Tableau license expires in 12 days
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Dashboard
Layout
Default
Phone
Device Preview
Size
Automatic
Sheets
Sheet 1
Sheet 2
Sheet 3
Sheet 4
Sheet 5
Sheet 6
Sheet 7
COVID-19 India Dashboard: Trend and State-wise Analysis
Trend of Positive COVID Cases
Moving Average of Confirmed (state level daily...
Day of Date
Month of Date
State-wise COVID Confirmed Cases (Map)
© 2026 Mapbox © OpenStreetMap
2 unknown
State wise onfirm
State-wise Recovered (Cured) Cases
State-wise Death Cases
Total
Maharashtra
Tamil Nadu
Andhra Pradesh
Karnataka
Delhi
Uttar Pradesh
West Bengal
Recovered
Deaths
No forecast
1 null
Confirmed (state level dai...
0
20,20,078
Recovered
0
13,77,384
Deaths
0
41,638
Objects
Horizontal Container
Vertical Container
Text
Extension
Pulse Metric
Image
Blank
Workflow
Web Page
Tiled
Floating
Show dashboard title
Data Source Sheet 1 Sheet 2 Sheet 3 Sheet 4 Sheet 5 Sheet 6 Sheet 7 Dashboard 1