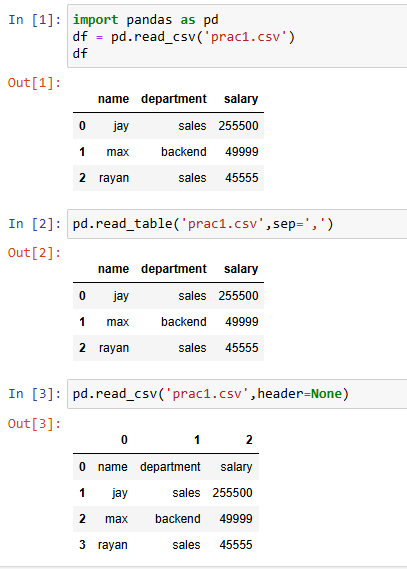
|  |  |  |  |
| --- | --- | --- | --- |
| **Course Code**  **PSDSP504** | **Course Title**  **Data Analysis and Visualization Practical** | **DATE** | **REMARK** |
| 1 | Implement Data Loading, Storage and File Formats. Read data  and store them in text format. | 20/07/24 |  |
| 2 | Implement the code to interact with Web APIs and to perform web  scrapping. | 27/07/24 |  |
| 3 | Demonstrate Data Cleaning and Preparation. | 03/08/24 |  |
| 4 | Implement Data wrangling on a data set. | 10/08/24 |  |
| 5 | Demonstrate the handling of missing data and string manipulation. | 17/08/24 |  |
| 6 | Create common charts with title, labels and descriptions using  Tableau. | 24/08/24 |  |
| 7 | Perform sorting and filtering using tableau, create visualizations  and publish it on Tableau Cloud. | 31/08/24 |  |
| 8 | Perform data visualization using Power BI. | 14/09/24 |  |
| 9 | Create reports using Power BI. | 28/09/24 |  |
| 10 | Create a data story in Tableau or power BI. | 05/10/24 |  |

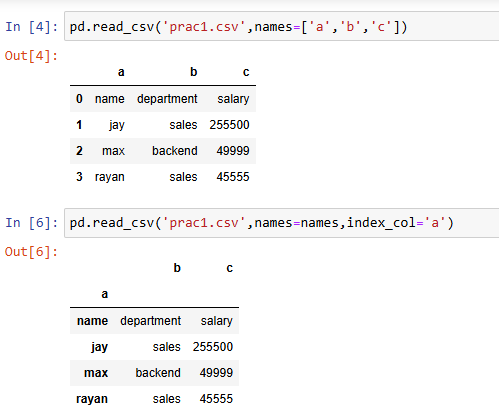
### PRACTICAL NO:01

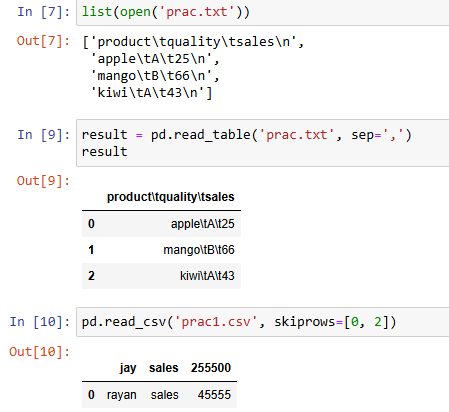
**AIM:** Implement Data Loading, Storage and File Formats. Read data and store them in

text format.

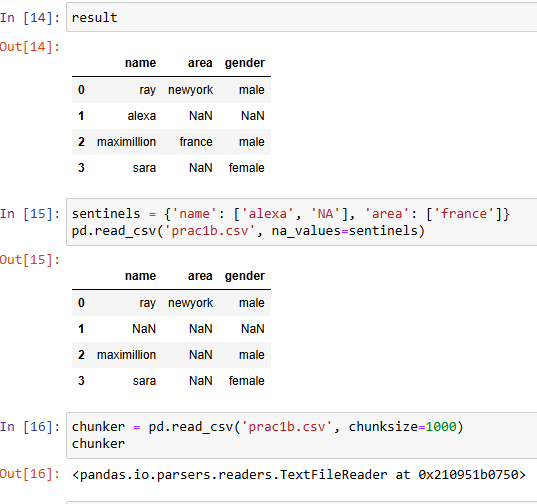
**CODE:**

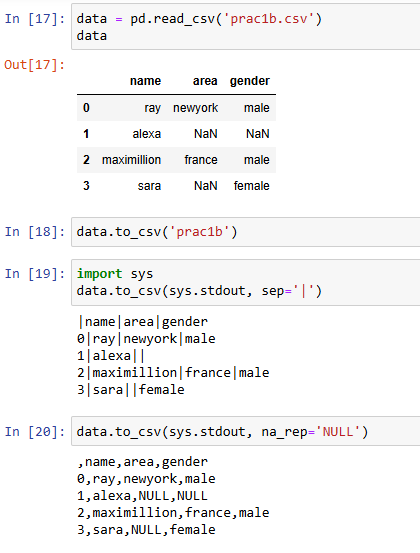
****

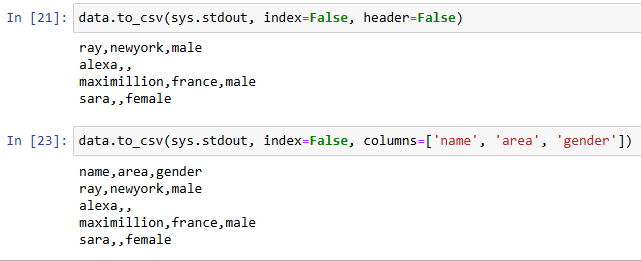
****

****

****

****

****

****

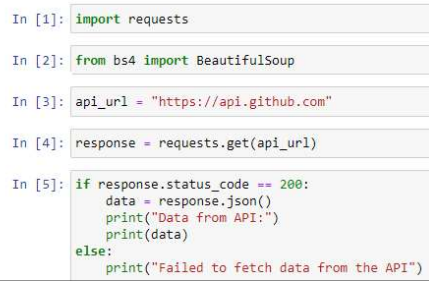
### PRACTICAL NO:02

**AIM:** Implement the code to interact with Web APIs and to perform web scrapping.

**CODE:**

**A.Interacting with a Web API**

1. Send an HTTP GET request to the API endpoint
2. Parse the JSON response

****

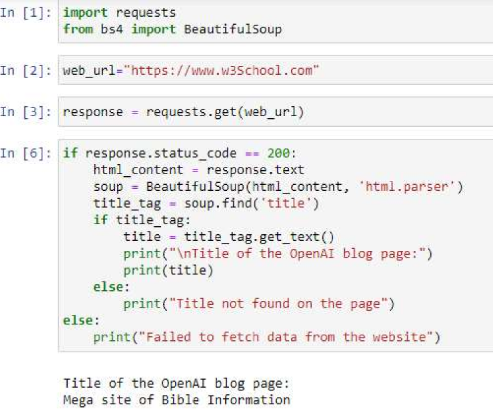
**OUTPUT:**

****

**B.** **Perform web scrapping**

1. Send an HTTP GET request to the website
2. Check if the request was successful
3. Parse the HTML content of the page
4. Find and print the title of the page

**CODE:**

****

### PRACTICAL NO:03

**AIM:** Demonstrate Data Cleaning and Preparation.

**CODE:**

import pandas as pd

data={'name':['jack','jay','tom','alice','sally'],'age':[None,34,32,None,33],'salary':[40000,None,600000,

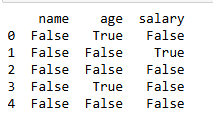
46777,80000]}

df=pd.DataFrame(data)

print("Missing Values:")



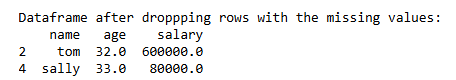
print(df.isnull())



df\_cleaned=df.dropna()

print("\nDataframe after droppping rows with the missing values:")

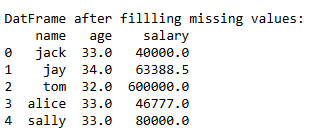
print(df\_cleaned)



df\_filled=df.fillna({'age':df['age'].mean(),'salary':df['salary'].median()})

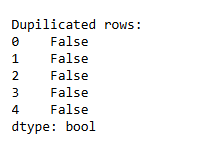
print("\nDatFrame after fillling missing values:")

print(df\_filled)



print("\nDupilicated rows:")

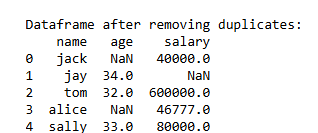
print(df.duplicated())



df\_no\_duplicates=df.drop\_duplicates()

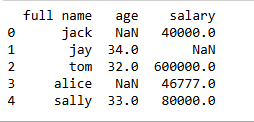
print("\nDataframe after removing duplicates:")

print(df\_no\_duplicates)



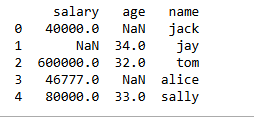
df\_renamed=df.rename(columns={'name':'full name'})

print(df\_renamed)



df\_reordered=df[['salary','age','name']]

print(df\_reordered)



### PRACTICAL NO:04

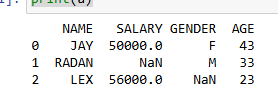
**AIM:** Implement Data wrangling on a data set.

**CODE:**

import pandas as pd

a=pd.read\_csv('SALARY.csv')

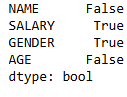
print(a)



a.isnull()



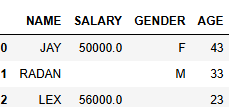
a.isnull().any()



a.dropna()



a.fillna("")



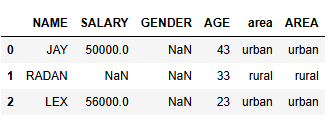
a['AREA']=['urban','rural','urban']

a



a['GENDER']=a['GENDER'].map({'M':0,'F':1,}).astype(float)

a



a.groupby('NAME').GENDER.value\_counts()



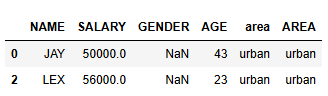
a.GENDER.unique()



a[a['SALARY']>50000]

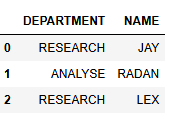


a[a['AREA']=='urban']



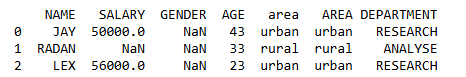
b=pd.DataFrame({'DEPARTMENT':['RESEARCH','ANALYSE','RESEARCH'],'NAME':['JAY','RADAN','LEX']})

b

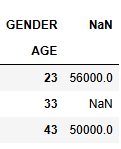


c=print(pd.merge(a,b,on='NAME'))

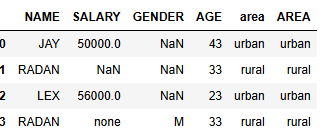
c



a.pivot(index='AGE',columns='GENDER',values='SALARY')



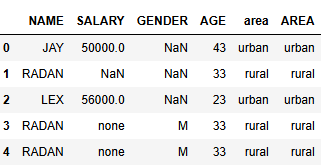
A



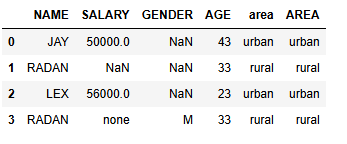
cow=pd.DataFrame([{'NAME':'RADAN','SALARY': 'none','GENDER':'M','AGE':33,'area':'rural','AREA':'rural'}])

a=pd.concat([a,cow],ignore\_index=True)

a



a.drop\_duplicates()



### PRACTICAL NO-05

**AIM:** Demonstrate the handling of missing data and string manipulation.

**CODE:**

In [1] : val = ‘a,b, guido’

In [2] : val.split(‘,’)



In [3] : pieces = [x.strip() for x in val.split(‘,’)]

In [4] : pieces



In [5] : first, second, third = pieces

In [6] : first :: + second + :: + third



In [7] : ‘::’.join(pieces)



In [8] : ‘guido’in val



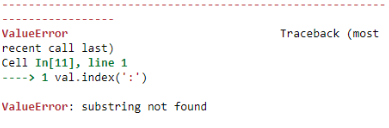
In [9] : val.index(‘,’)



In [10] : val.find(‘:’)



In [11] : val.index(‘:’)



In [12] : val.count(‘,’)



In [13] : val.replace(‘,’, ‘::’)



In [14] : val.replace(‘,’, “”)

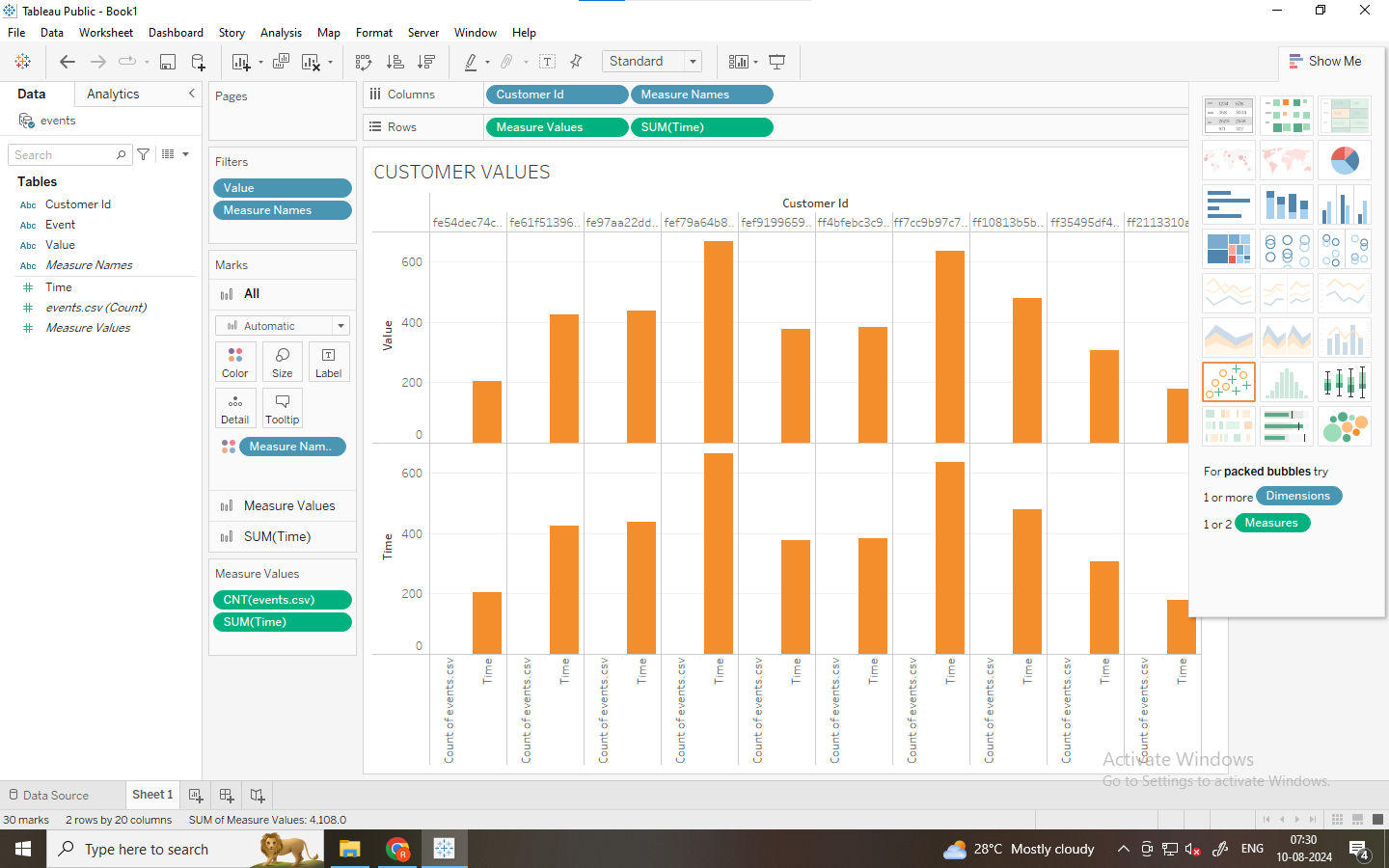


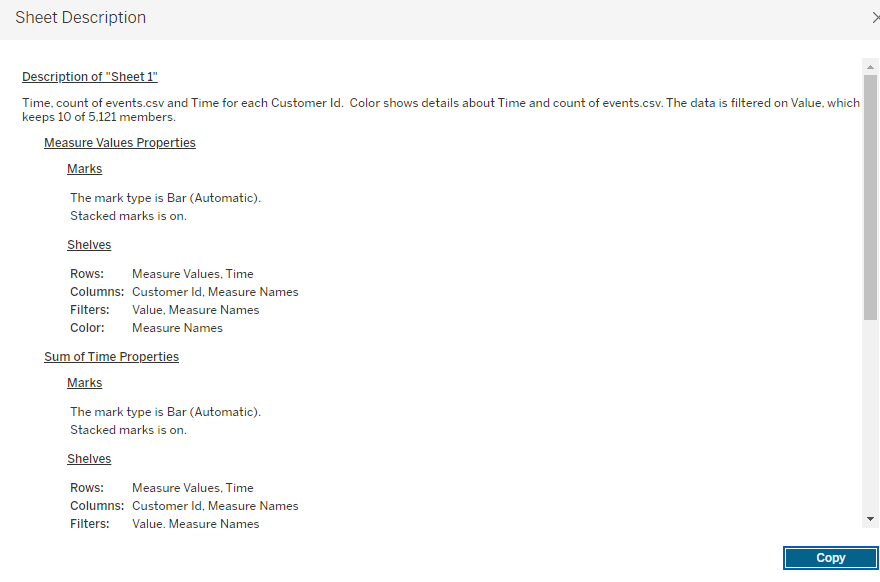
### PRACTICAL NO :06

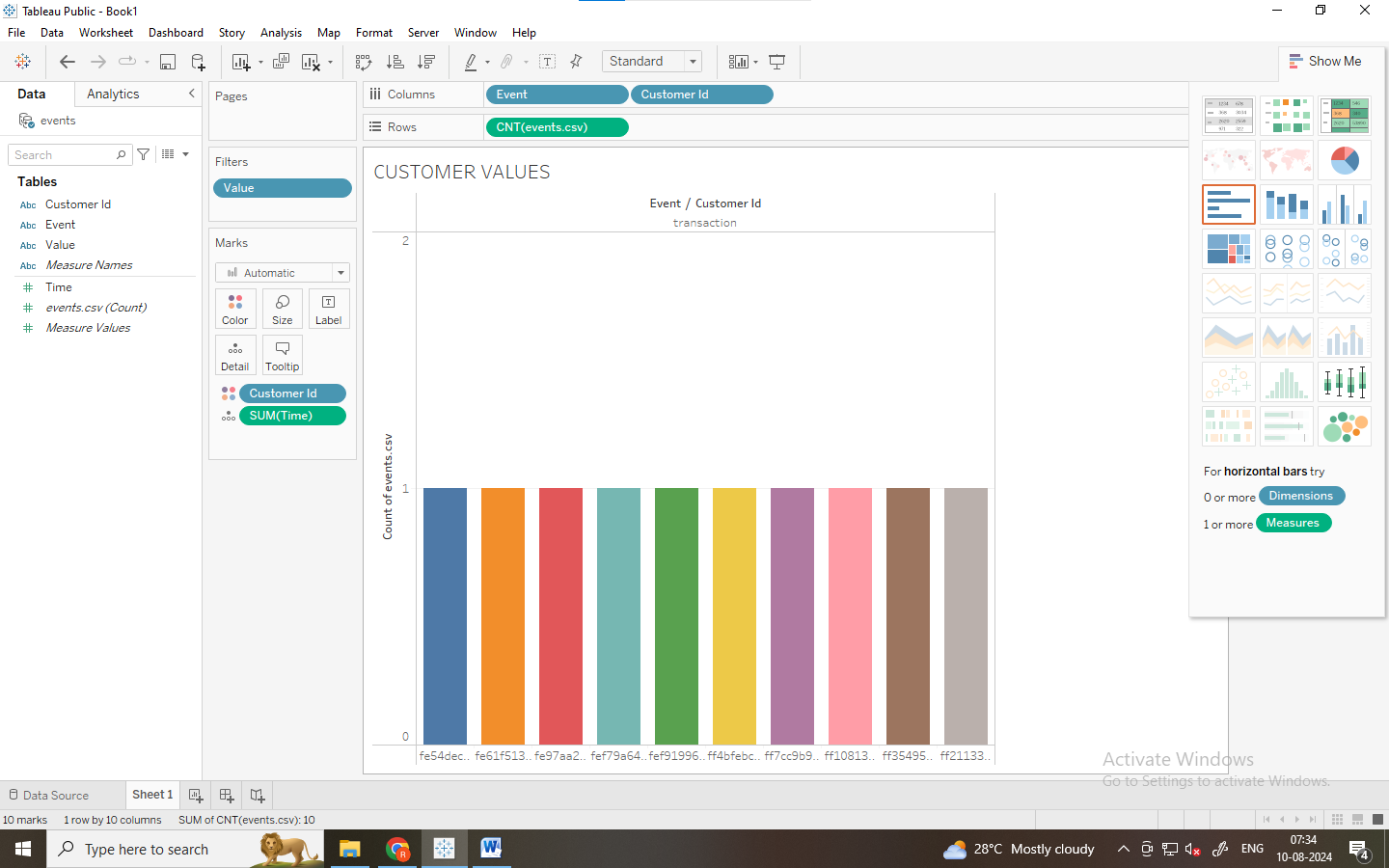
**AIM:** Create common charts with title, labels and descriptions using Tableau.

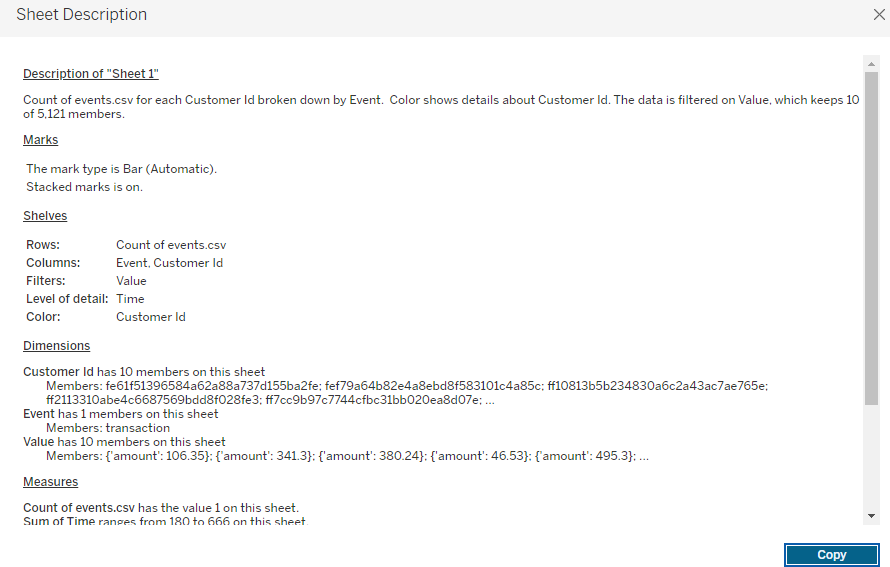
**Steps:**

1. Right click on the worksheet title.
2. Select "Edit title"
3. In the dialog box that appears create the title that you require, inserting any of the fields to customise the title from the "Insert" drop-down menu. In the above case, my title reads: ...
4. Hit "OK", and you are done.

**OUTPUT**: 

**DISCRIPTION**: 

**OUTPUT**: 

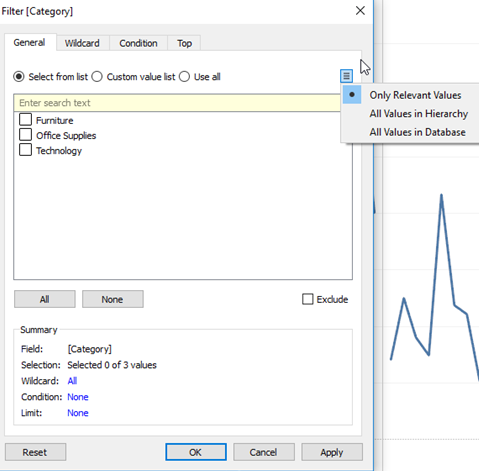
**DISCRIPTION**: 

### PRACTICAL NO:07

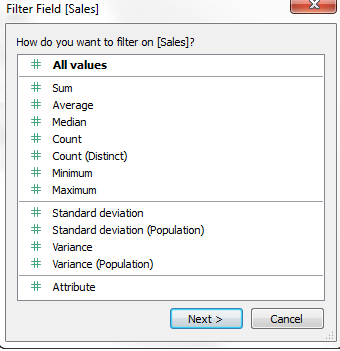
**AIM:** Perform sorting and filtering using tableau, create visualizations and publish it on Tableau Cloud.

**Steps:**

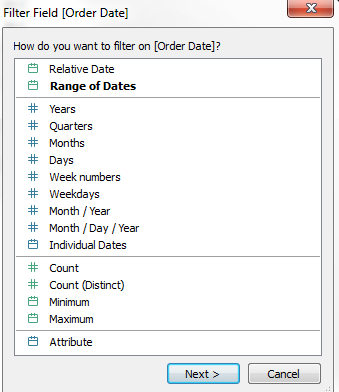
1. **Filter categorical data (dimensions)**:
2. Dimensions contain discrete categorical data, so filtering this type of field generally involves selecting the values to include or exclude.
3. When you drag a dimension from the Data pane to the Filters shelf in Tableau Desktop, the following Filter dialog box appears:



1. **Filter Measures are the filters applied on the measure fieldS:**
2. Measures contain quantitative data, so filtering this type of field generally involves selecting a range of values that you want to include.
3. When you drag a measure from the Data pane to the Filters shelf in Tableau Desktop, the following dialog box appears:

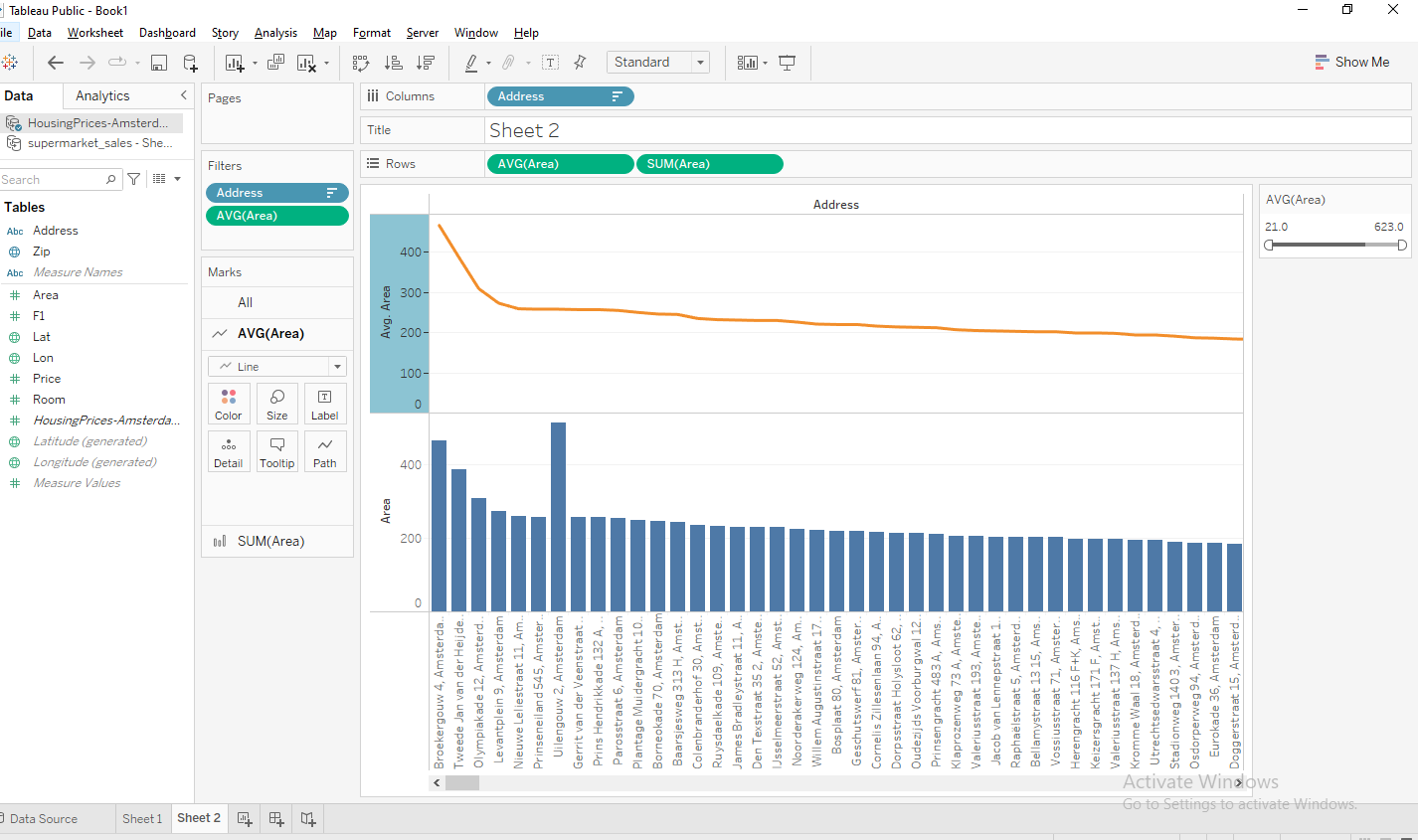


1. **Filter Dates are the filters applied on the date fields.**
2. When you drag a date field from the Data pane to the Filters shelf in Tableau Desktop, the following Filter Field dialog box appears:

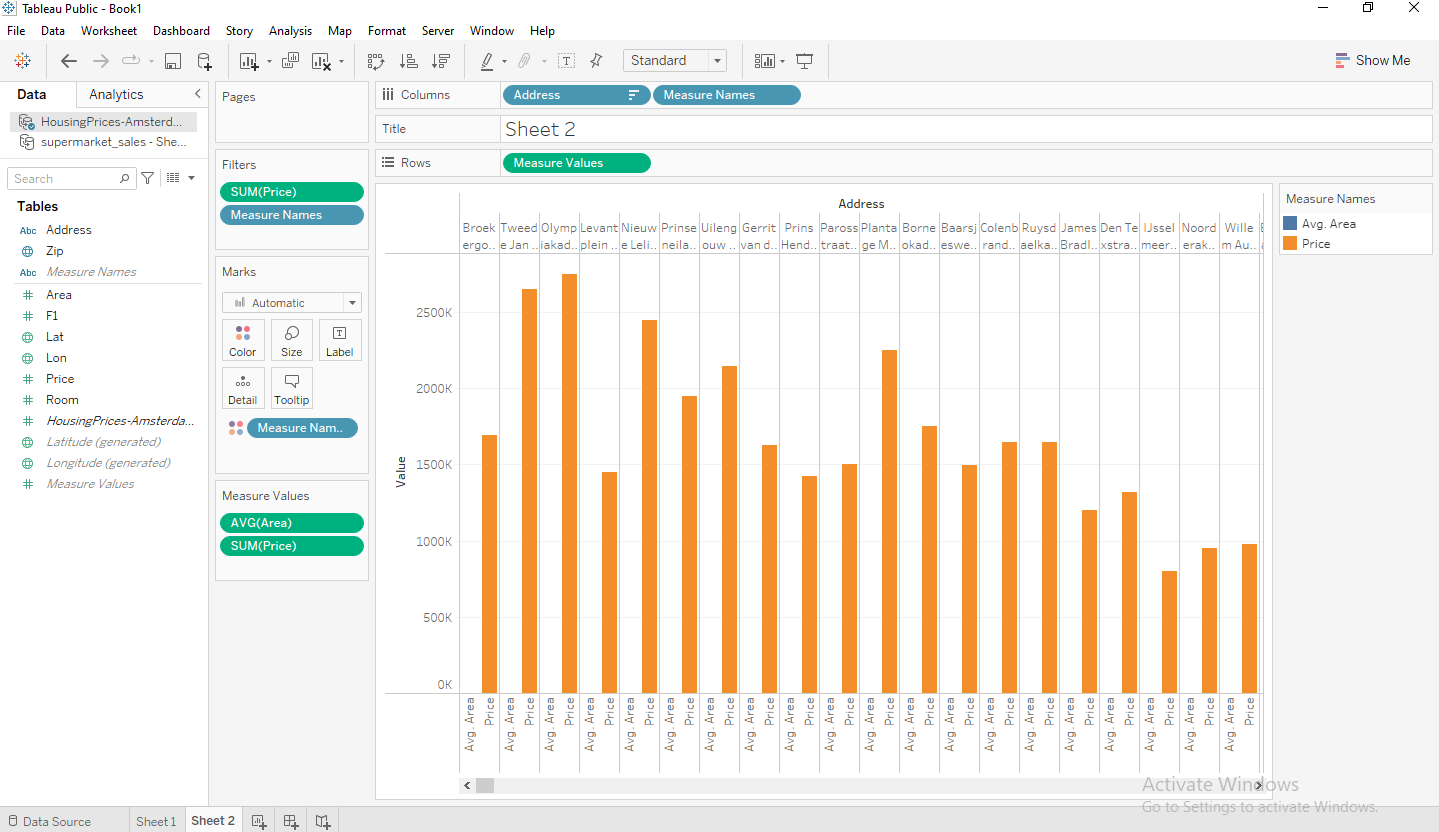


**EXECUTION:**

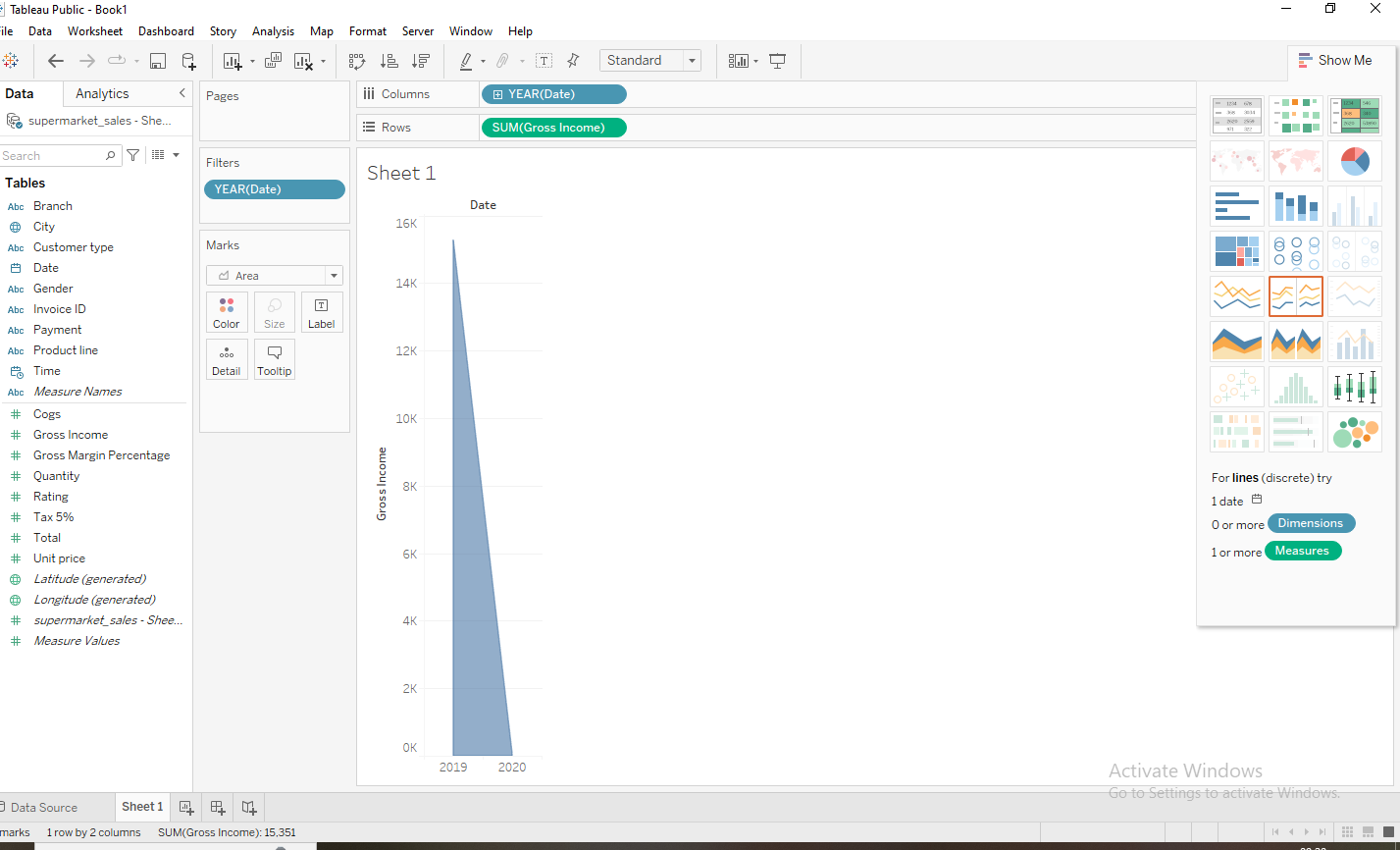
* Filter Dimensions are the filters applied on the dimension fields.



* Filter Measures are the filters applied on the measure fields.



* Filter Dates are the filters applied on the date fields.



**Sorting:**

Tableau Sort Data:

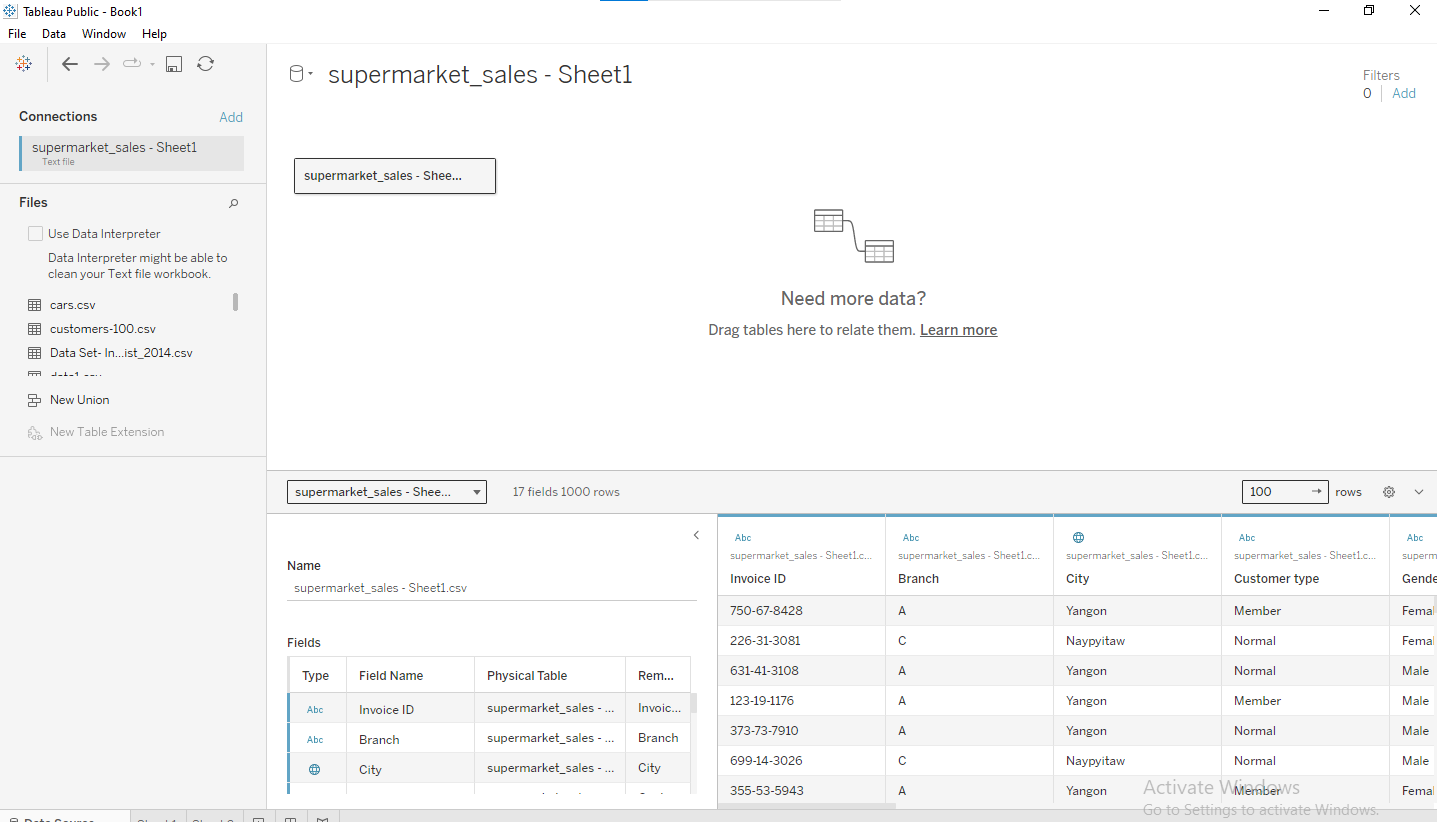
Data present in the worksheet can be sorted based on the requirement. It can sort the data

based on the data source such as ascending, descending order, or depend on any measured

value.

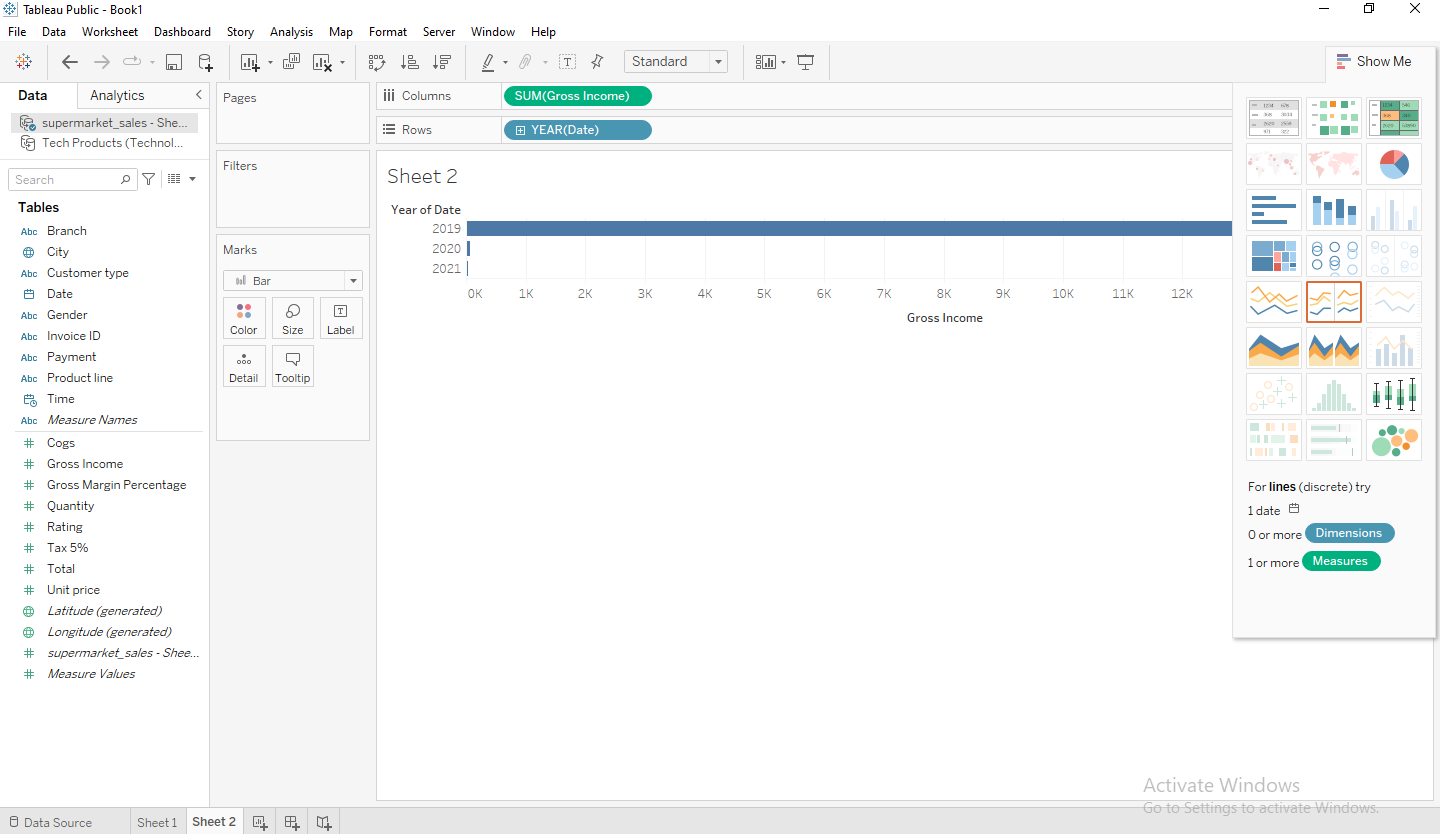
* Step1: Add the sample-superstore data source with Tableau and drag the Order table to the

pane shown in the below screenshot.



* Step2: Go to the worksheet and drag the date to the row shelf and the

Gross income to the column shelf.

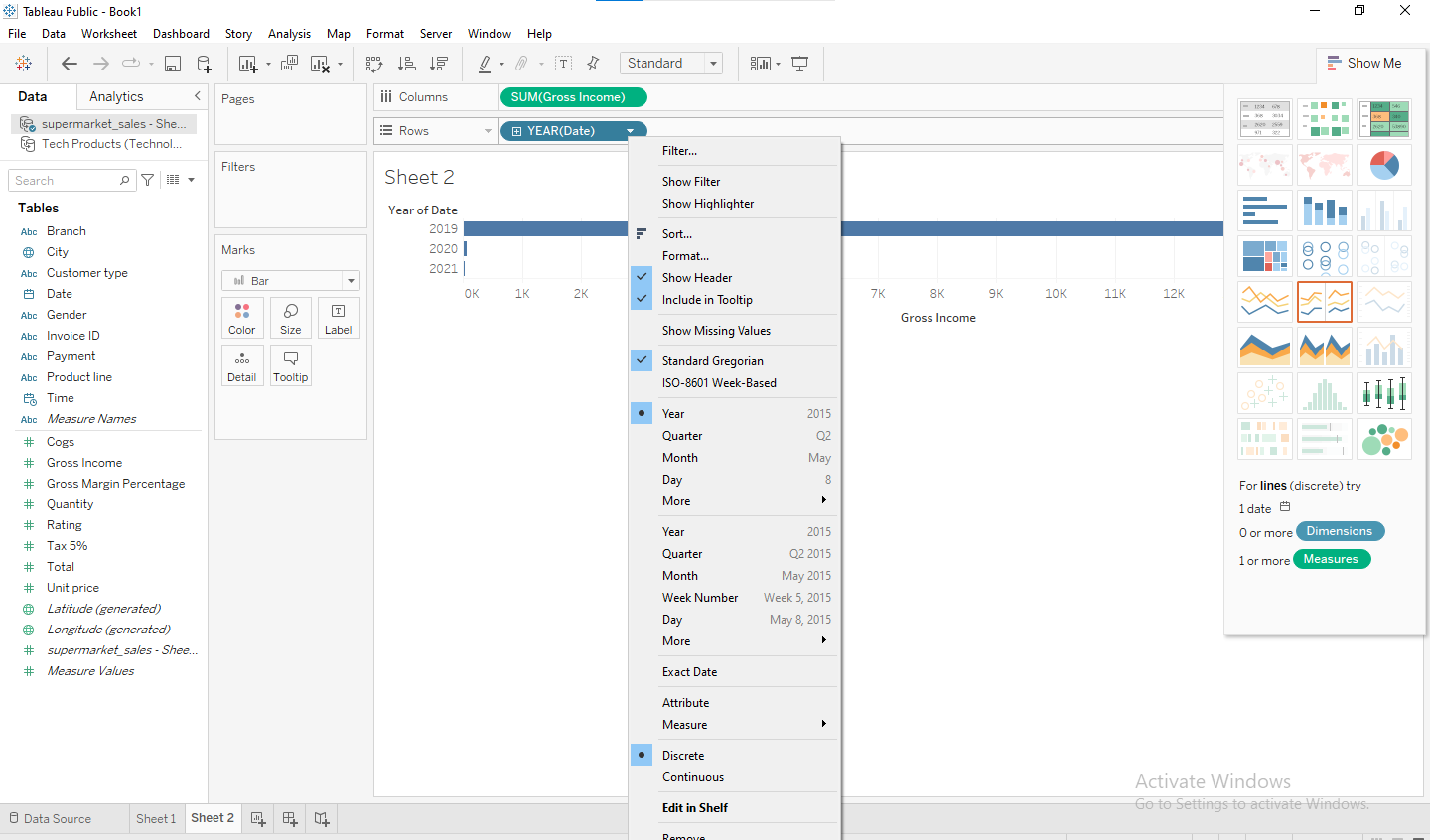


It creates a horizontal bar chart. Category field present in the visual order, and it is sorted

based on data source by default. We can change the order of sorting by following the below

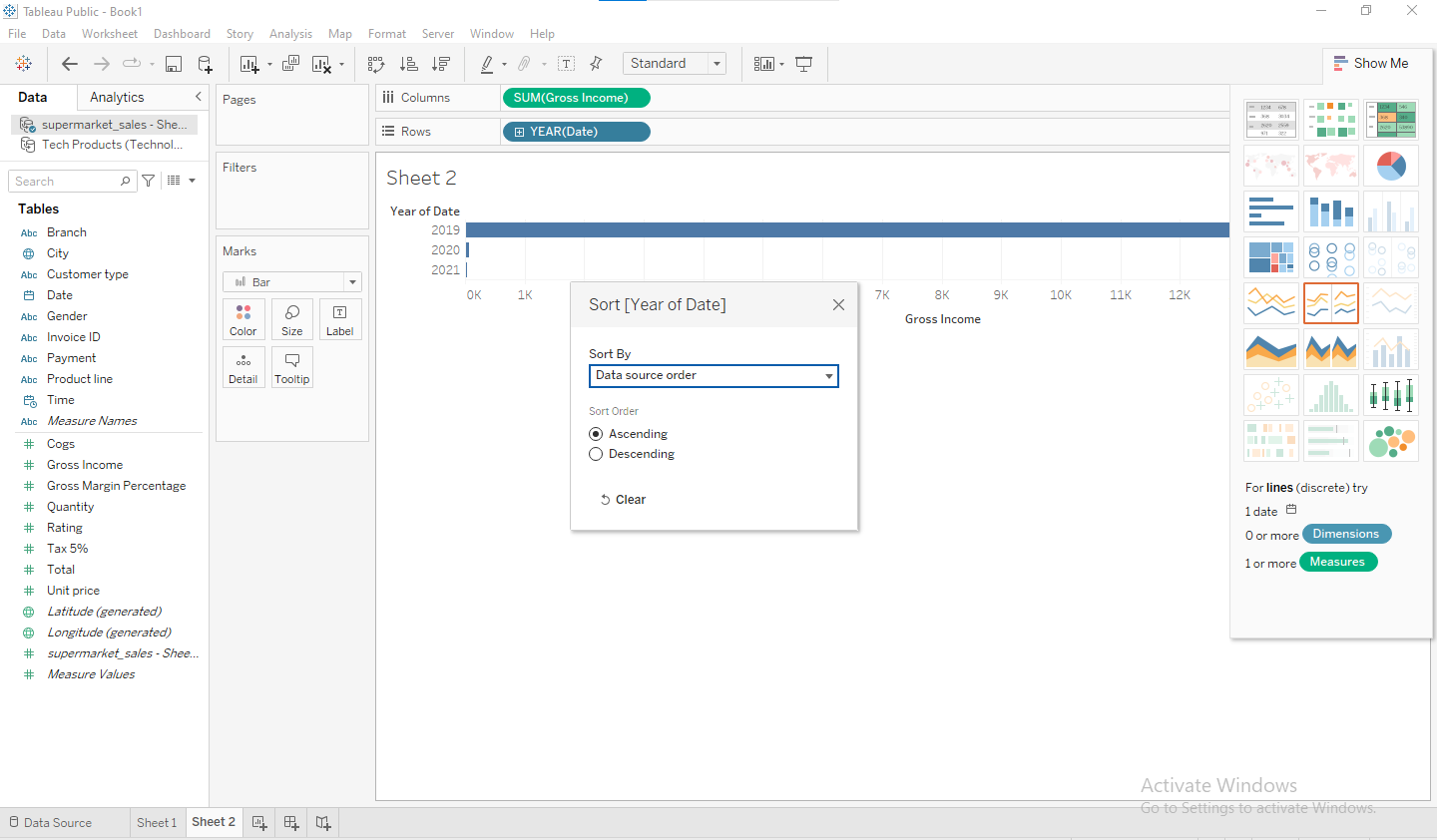
procedure.

* Step3: Right-click on the date and select Sort option.



After that, it opens the Sort window. All options present inside the sort window is shown

below as follows:



Sort Order:

* Ascending: It sorts the order of selected dimensions and measures in ascending order.
* Descending: It sorts the order of selected dimensions and measures in descending
* order.

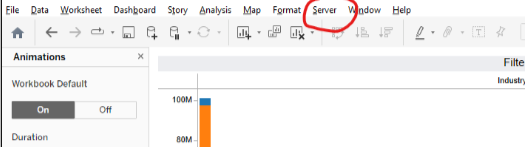
Sort By:

The field can be sorted in different types of methods that are explained below as follows.

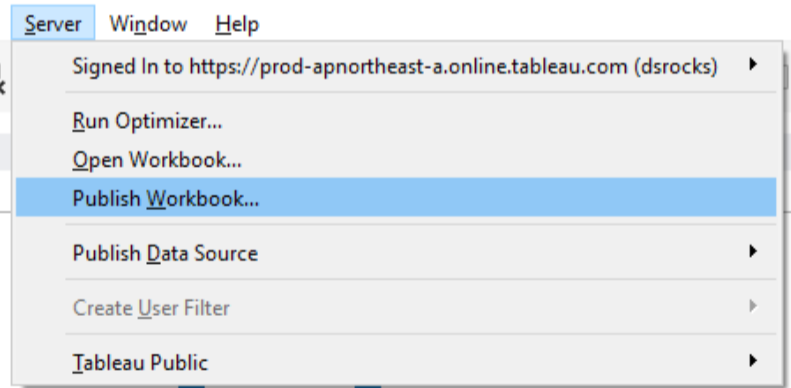
* Data source order: It sorts the field based on data source order.
* Alphabetic: It sorts the dimensions and measures in alphabetical order.
* Field: It sorts the field based on the other measure or dimension values.
* Manual: It can manually sort the data.

**Publish Data on Tableau cloud**

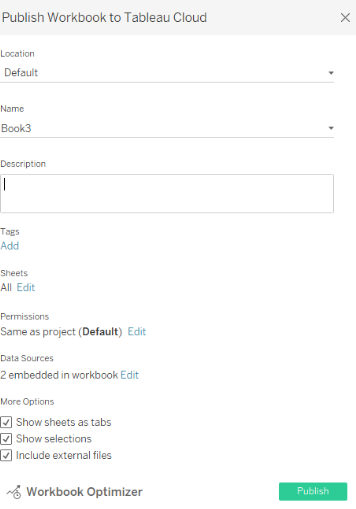
* Step 1: Create worksheets / dashboards in tableau.
* Step2: Click on “server” placed at top row selections

****

Step3: Click on publish workbook.

****

* Step4: Enter appropriate details in pop-up box.

****

Click on Publish button placed at bottom-right of pop-up box.

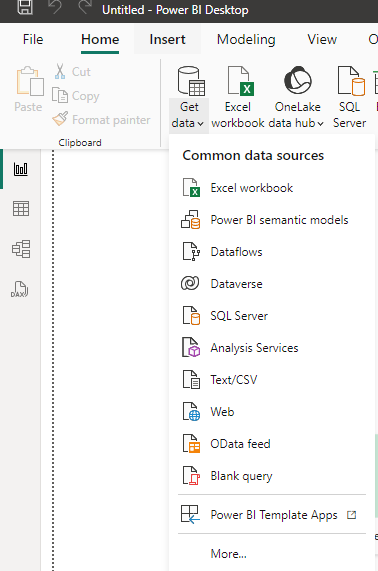
After clicking on “Publish” your workbook will be published on Tableau cloud.

### PRACTICAL NO-08

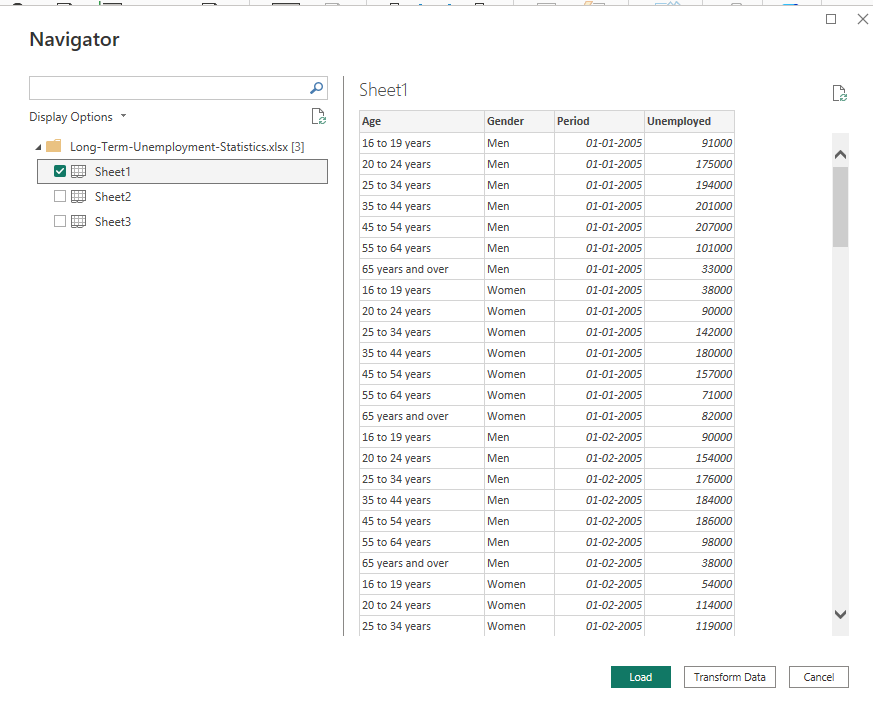
**AIM:** Perform data visualization using Power BI.

**STEPS:**

1. Click on the **"Home"** tab in the ribbon at the top.
2. Select **"Get Data"** from the options available,from the home tab to connect to your data source (e.g., Excel, SQL Server, CSV).

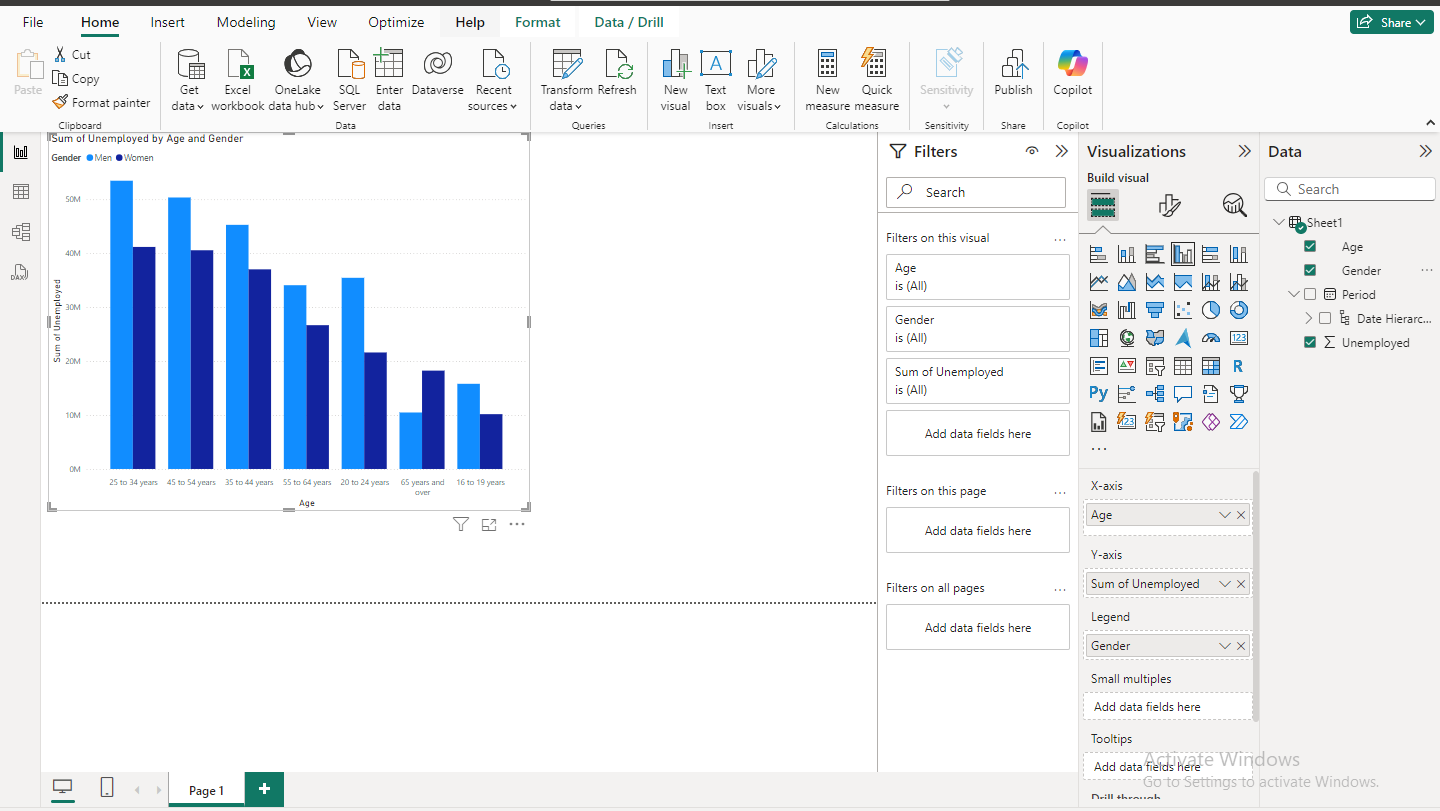


1. **Load the selected data source**



1. Create an visual

* In the **"Visual"** view, which is usually the default view when you open Power BI Desktop, you can start creating your report.
* Drag and drop fields from the **"Fields"** pane on the right to the **"Report Canvas"** in the center.
* Choose the type of visualization you want to create by selecting from the options in the **"Visualizations"** pane.

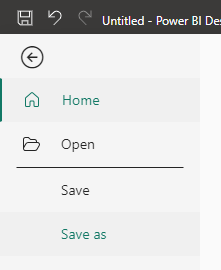


1. **Customize your visuals**

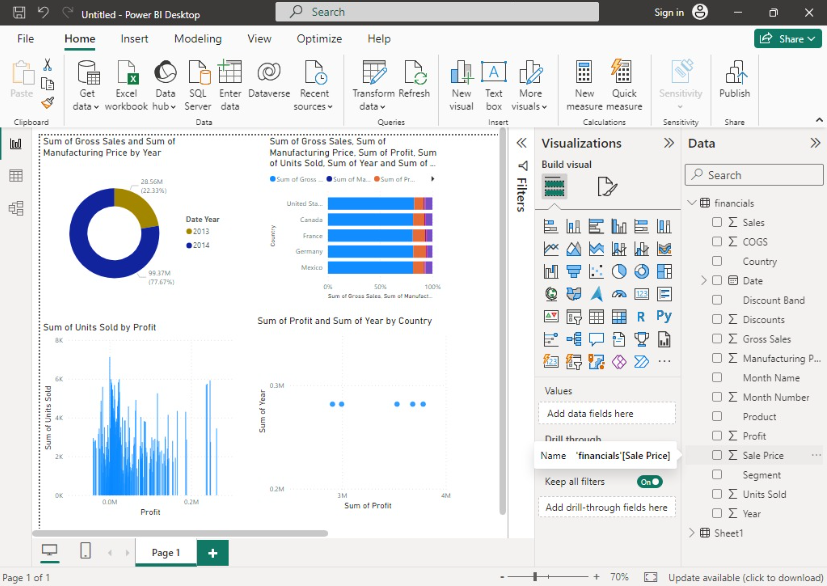
* Use the **"Format"** pane to adjust the appearance of your visualizations, such as changing colors, fonts, and labels.
* Add additional elements like text boxes, images, or shapes by using the options in the **"Insert"** tab.

1. **Save the created visual**

* Click **"File"** in the top left corner.
* Select **"Save"** to save your report



**OUTPUT:**

****

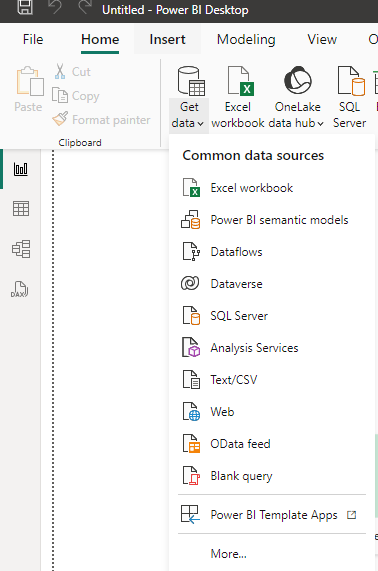
blob:https://web.whatsapp.com/be00b456-7771-42d8-947f-bb9083ca26f5blob:https://web.whatsapp.com/be00b456-7771-42d8-947f-bb9083ca26f5

### PRACTICAL NO-09

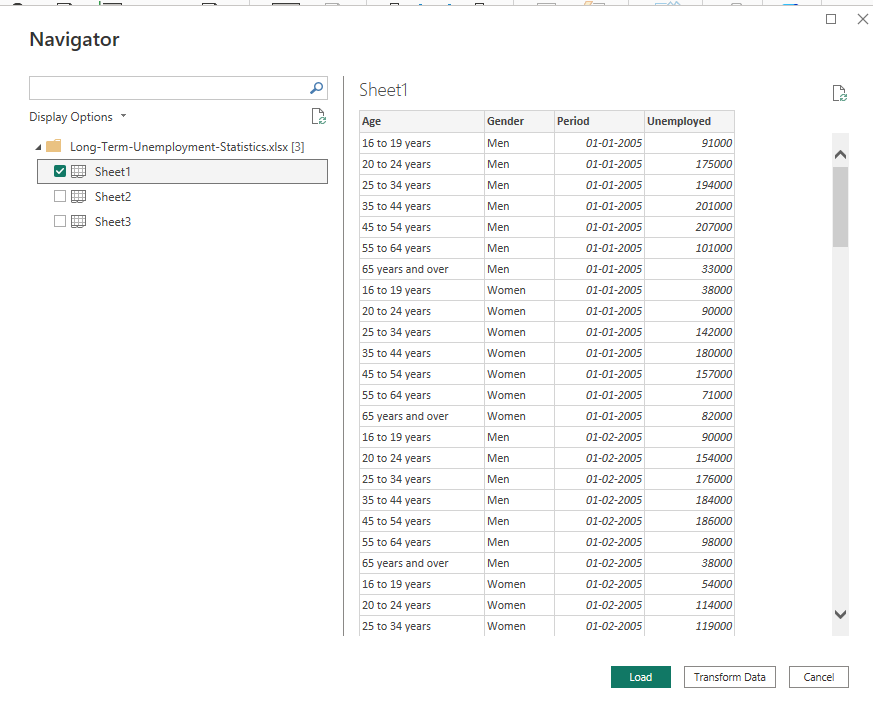
**AIM:** Create reports using Power BI.

**STEPS:**

1. Click on the **"Home"** tab in the ribbon at the top.
2. Select **"Get Data"** from the options available,from the home tab to connect to your data source (e.g., Excel, SQL Server, CSV).



1. **Load the selected data source**



1. Create an report

* In the **"Report"** view, which is usually the default view when you open Power BI Desktop, you can start creating your report.
* Drag and drop fields from the **"Fields"** pane on the right to the **"Report Canvas"** in the center.

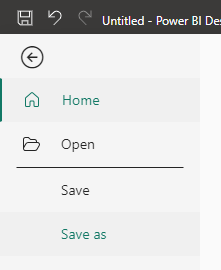
1. Choose the type of visualization you want to create by selecting from the options in the **"Visualizations"** pane
2. **Customize your report**

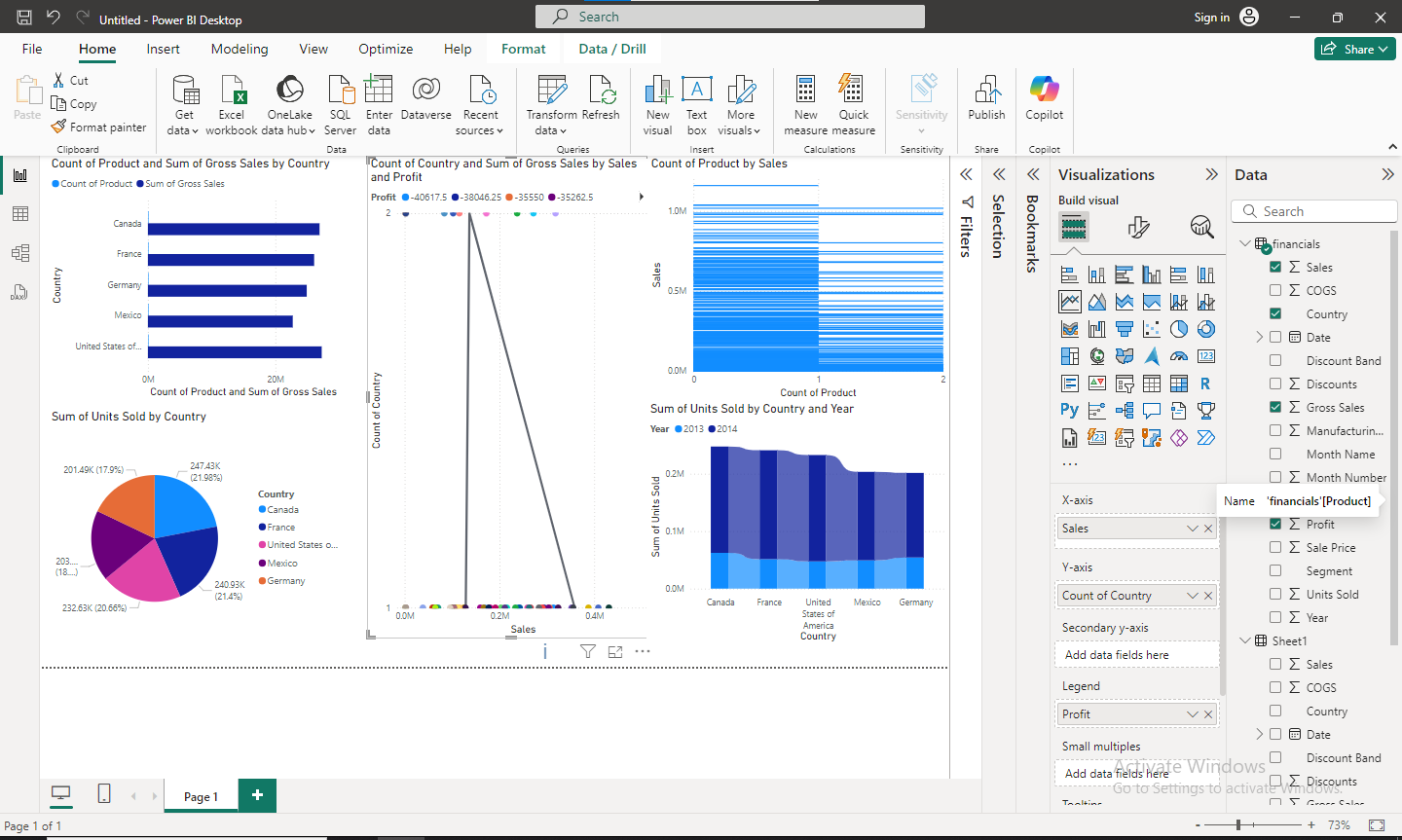
* Use the **"Format"** pane to adjust the appearance of your visualizations, such as changing colors, fonts, and labels.
* Add additional elements like text boxes, images, or shapes by using the options in the **"Insert"** tab.

1. **Save the created report**

* Click **"File"** in the top left corner.

Select **"Save"** to save your report

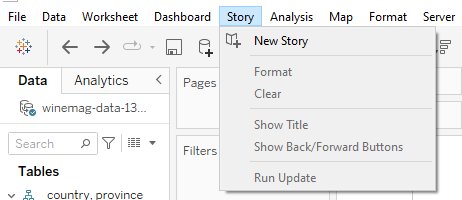
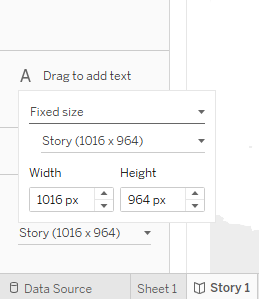
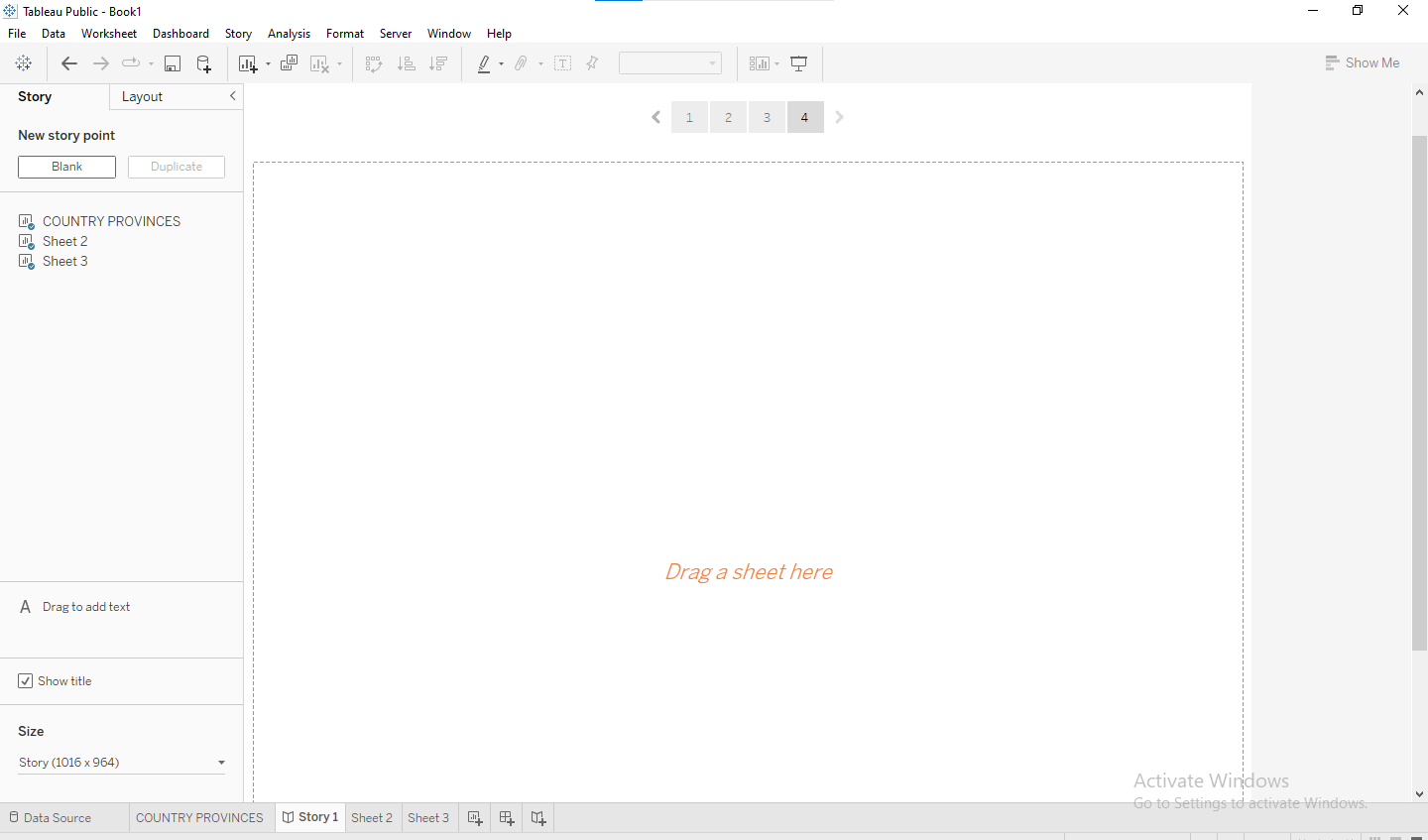
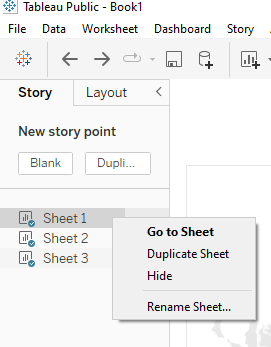
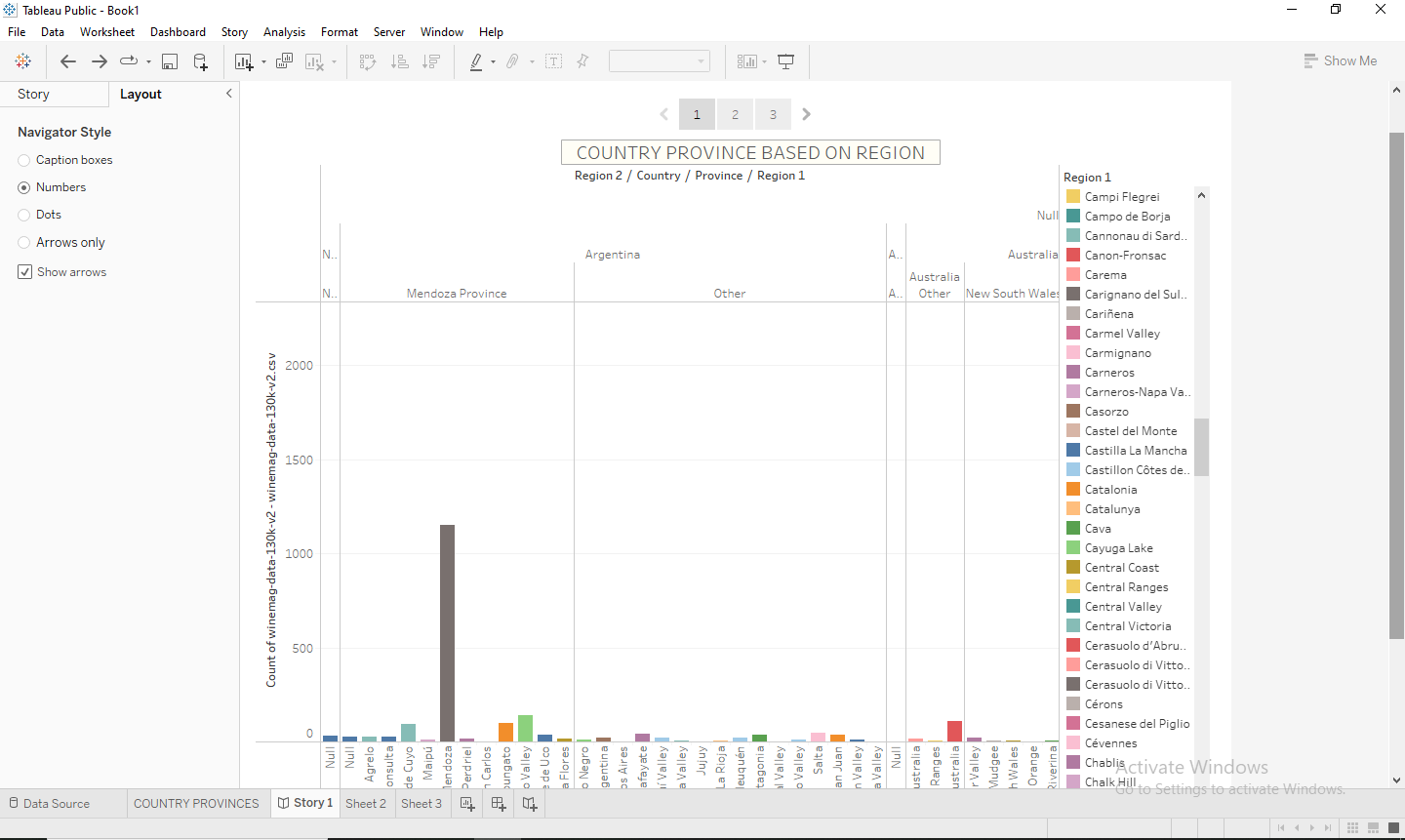
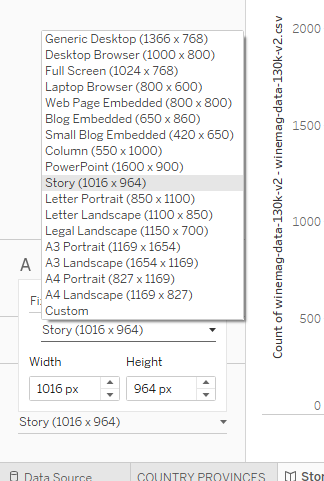


****

### PRACTICAL NO-10

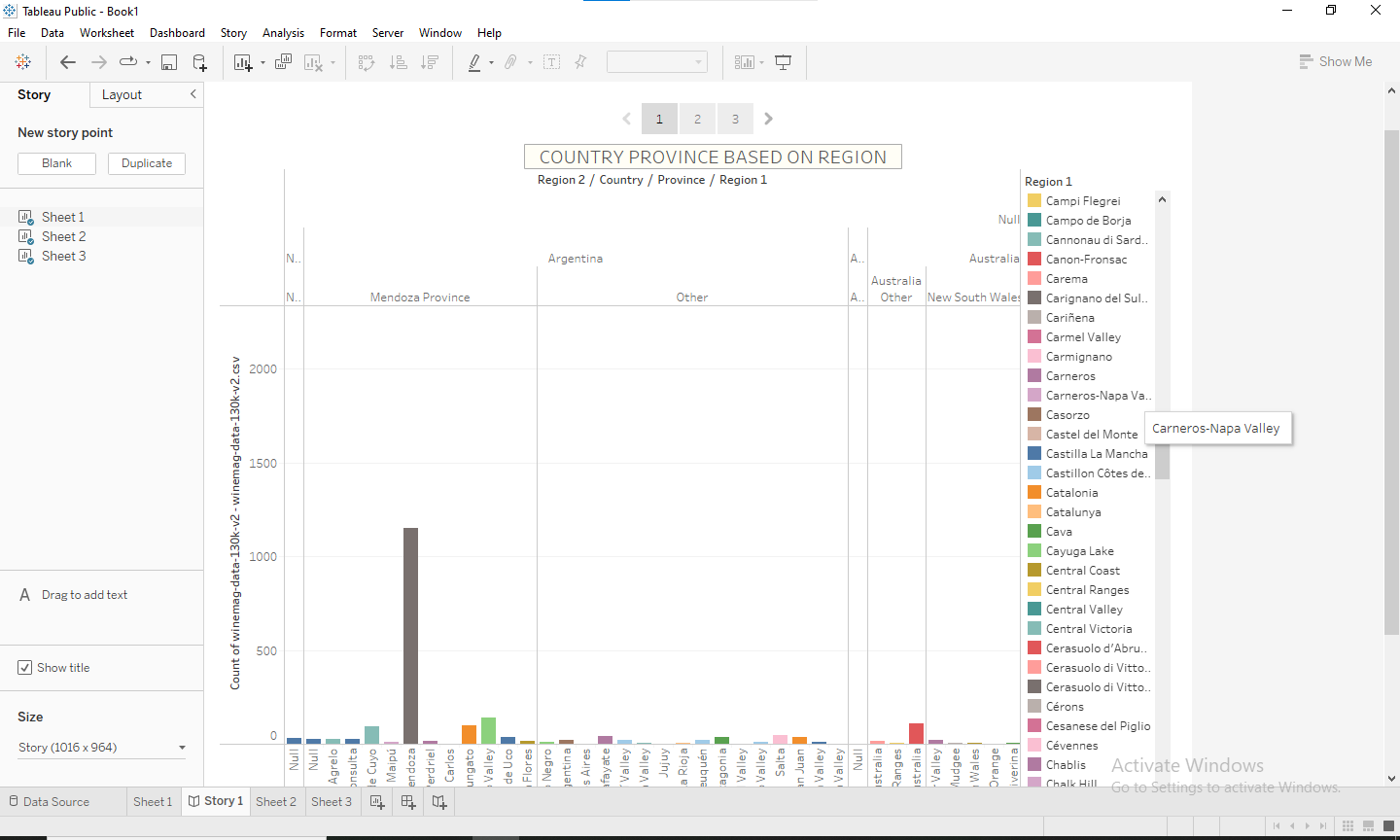
**AIM:** Create a data story in Tableau or power BI.

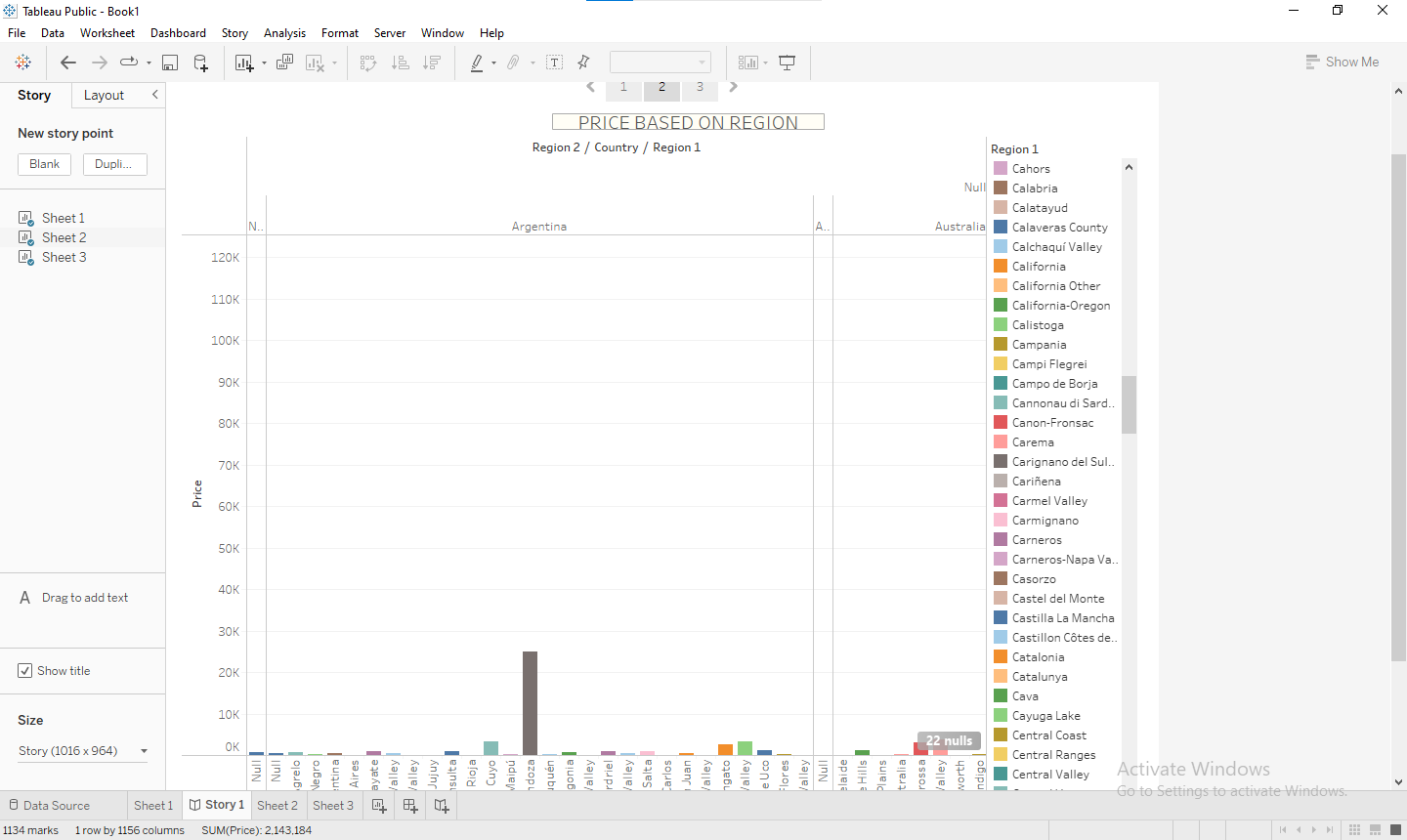
**STEPS:**

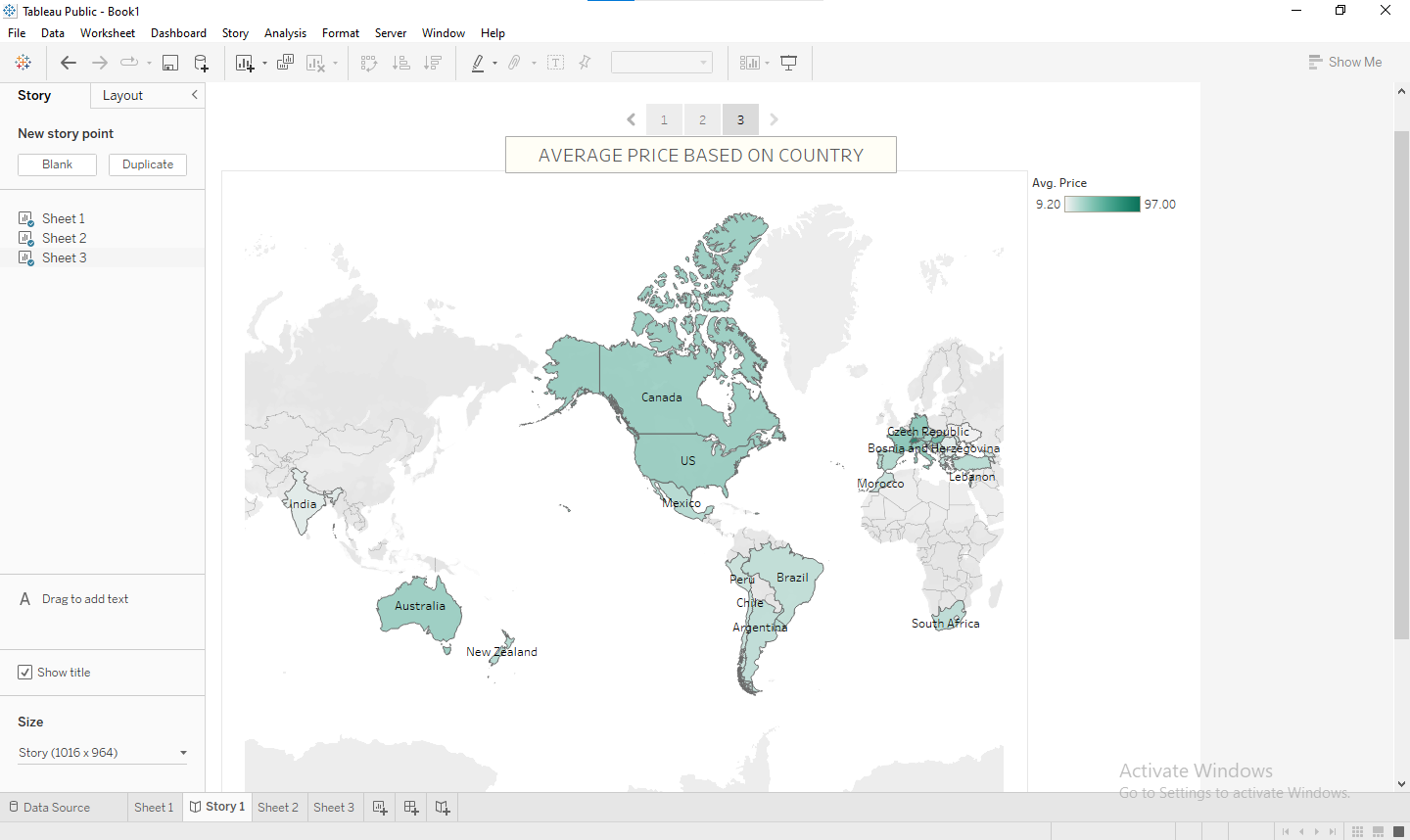
1. Click the New Story tab
2. In the lower-left corner of the screen, choose a size for your story. Choose from one of the predefined sizes, or set a custom size, in pixels: 
3. To start building your story, double-click a sheet on the left to add it to a story point.In Tableau Desktop, you can also drag sheets into your story point. 
4. By default, your story gets its title from the sheet name. To edit it, right-click the sheet tab,and choose Rename Sheet. If you're using Tableau Desktop, you can also rename a story by double-clicking the title. 
5. Click the Layout tab. Choose a navigator style that best suits your story, and show or hide the next and previous arrows. 
6. Click the Size drop-down menu and select the story you want the dashboard to fit inside. 
7. Present your story

**OUTPUT:**

Creating story in tableau

****

****

****