EXPERIMENT-1

AIM: Understanding Data, what is data, where to find data, data wrangling, data clean up Basics -formatting, outliers, duplicates, normalizing and standardizing data.

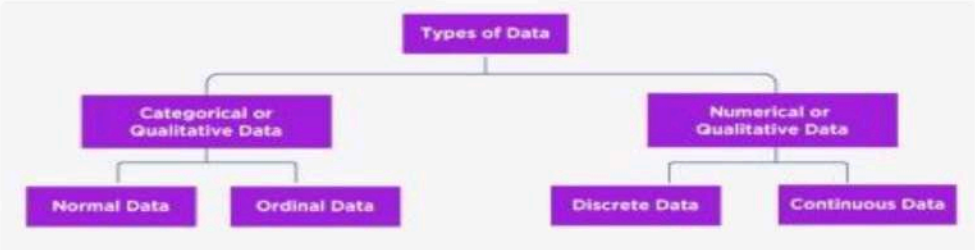
PROCEDURE:

Step 1: Study about data and its importance
Step 2: Understanding about data wrangling procedure
Step 3: Understanding the importance of normalizing and standardizing the data

SOURCE CODE:

Data is defined as "**facts and statistics** collected together for reference or analysis."
Data helps us **understand** the world around us by providing **context and insights that would otherwise be unavailable**

Data is also important because it allows us **to test hypotheses and make predictions**



Following are the ways to find data:

1. Using a Telephone
2. Government and Civic Open Data Worldwide
3. Organization and Non-Government Organization (NGO)
4. Education and University Data
5. Medical and Scientific Data
6. Crowd sourced Data and APIs

Data Wrangling:

Data wrangling is the process of transforming raw data into easily understandable formats and organizing sets into a single structure for further processing.

Examples of data wrangling include:

- Merging multiple data sources into a single dataset for analysis
- Identifying gaps in data (for example, empty cells in a spreadsheet) and either filling or deleting them

- Deleting data that's either unnecessary or irrelevant to the project you're working on
- Identifying extreme outliers in data and either explaining the discrepancies or removing them so that analysis can take place.

Code:

```
!pip install pandas

#Discovery

import pandas as pd

df=pd.read_csv('Customer.csv')

df
```

Output:

Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
0	CG-12520	Claire Gute	Consumer	67	United States	Henderson	Kentucky	42420	South
1	DV-13045	Darrin Van Huff	Corporate	31	United States	Los Angeles	California	90036	West

```
#Structuring

df.dtypes
```

Output:

```
Customer ID      object
Customer Name    object
Segment          object
Age               int64
```

```
df.shape
```

Output:

```
(793, 9)
```

```
df.head(10)
```

Output:

Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
0	CG-12520	Claire Gute	Consumer	67	United States	Henderson	Kentucky	42420	South
1	DV-13045	Darrin Van Huff	Corporate	31	United States	Los Angeles	California	90036	West
2	SO-20335	Sean O'Donnell	Consumer	65	United States	Fort Lauderdale	Florida	33311	South

```
df.tail()
```

Output:

Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
788	CJ-11875	Carl Jackson	Corporate	64	United States	Philadelphia	Pennsylvania	19140	East
789	RS-19870	Roy Skaria	Home Office	39	United States	Burlington	Iowa	52601	Central

df.info()

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 793 entries, 0 to 792
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Customer ID    793 non-null    object
 1   Customer Name  793 non-null    object
 2   Segment        793 non-null    object
```

df.describe()

Output:

	Age	Postal Code
count	793.000000	793.000000
mean	44.467844	55229.717528

df['City'].count()

Output:

```
793
```

df['City']=df['City'].str.lower()

df.head()

Output:

Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
0	CG-12520	Claire Gute	Consumer	67	United States	henderson	Kentucky	42420	South
1	DV-13045	Darrin Van Huff	Corporate	31	United States	los angeles	California	90036	West

df['State']=df['State'].str.replace('Iowa','Ohio')

df.tail()

Output:

Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
788	CJ-11875	Carl Jackson	Corporate	64	United States	philadelphia	Pennsylvania	19140	East

Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
789	RS-19870	Roy Skaria	Home Office	39	United States	burlington	Ohio	52601	Central

#Data Cleaning

df.isnull().sum().sum()

0

df.duplicated()

Output:

```
788    False
789    False
790    False
791    False
792    False
Length: 793, dtype: bool
```

df[df['Age'] < 40]

Output:

Customer ID	Customer Name	Segment	Age	Country	City	State	Postal Code	Region	
1	DV-13045	Darrin Van Huff	Corporate	31	United States	los angeles	California	90036	West
3	BH-11710	Brosina Hoffman	Consumer	20	United States	los angeles	California	90032	West

#atleast one cell is NAN

df.dropna(inplace=True)

df.shape

df.drop(['Age','Segment'],axis=1,inplace=True)

df.head(10)

#Adding

df['Amount']=df['Postal Code']/50

df.head()

Output:

Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Amount	
0	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420	South	848.40
1	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036	West	1800.72

EXPERIMENT- 2:

AIM: Develop the python script to parse the pdf files using pdfminer.

PROCEDURE:

Step1: Set up PDFMiner using
!pip install pdfminer
!pip install pdfminer.six.

Step2: Use extract text method found in pdfminer.highlevel to extract text from the PDF file

Step3: Tokenize the text file using NLTK.tokenize RegexpTokenizer

Step4: Perform operations such as getting frequency distributions of the words, getting words more than some length etc.

Step5: Use method such as collocations or collocation list to get most frequently sequence of words occurring in the text

```
from nltk.tokenize import RegexpTokenizer

from pdfminer.high_level import extract_text

from nltk.probability import FreqDist

# Extract the text from PDF file

text = extract_text('2010.00462.pdf')

# Create an instance of tokenizer using NLTK ResexpTokenizer

tokenizer = RegexpTokenizer('\w+')

# Tokenize the text read from PDF

tokens = tokenizer.tokenize(text)

# Find Frequency Distribution

freqdist = FreqDist(tokens)

# Find words whose length is greater than 5 and frequency greater than 20

long_frequent_words = [words for words in tokens if len(words) > 5 and freqdist[words] > 20]

long_frequent_words
```

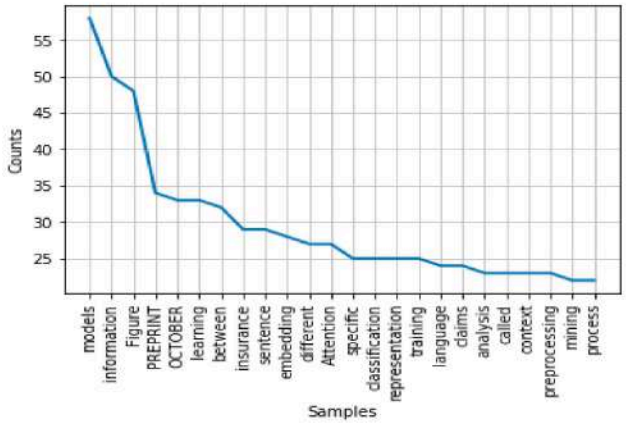
Output:

```
['PREPRINT',
 'different',
```

```
'insurance',
'language',
'insurance',
'information',
'analysis',
'mining',
'insurance',
```

FreqDist(long_frequent_words).plot()

```
In [36]: FreqDist(long_frequent_words).plot()
```




```
Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7fce54471450>
```

In case you want to find which all words occurred together more often, here is the command you will need to execute:

#Create NLTK.Text instance using tokens created using tokenizer

text = nltk.Text(tokens)

Execute collocation_list method on nltk.Text instance

text.collocation_list()

```
In [39]: text.collocation_list()
Out[39]: ['PREPRINT OCTOBER',
          'URL https',
          'machine learning',
          'Language Processing',
          'Natural Language',
          'text mining',
          'Multi Head',
          'neural networks',
          'Neural Networks',
          'original paper',
          'Question Answering',
          'Feed Forward',
          'Head Attention',
          'Data Analytics',
          'NLP models',
          'tabular data',
          'https www',
          'Stemming Algorithm',
          'fine tuning',
          'new opportunities']
```

EXPERIMENT-3

AIM: Develop the python Shell Script to do the basic data cleanup on child labor and child marriage data.xlsx

a) Check duplicates and missing data

b) Eliminate mismatches

c) Cleans line breaks, spaces, and special characters.

PROCEDURE:

Step1: Import required libraries like pandas ,numpy and xlrd

Step2: Read data using read.csv method in pandas

Step3: Find basic info about data

Step4: check duplicates and missing data.

Step5: Eliminate mismatches

Step6: Clean line breaks, spaces and special characters

Step7: Detect outliers

Step 8: Normalize Casing

Code:

```
import pandas as pd

cl= pd.read_csv('National Child Labour.csv')

cl

cl.shape

cm= pd.read_csv('Child Marriage.csv')

cm
```

Output:

	Country Name	Female Married by 15	Female Married by 18	Reference year	Data source	Male Married by 18	Male Reference year	Data source.1	
0	Afghanistan	4	28.0	2017.0	ALCS 2016-17	7	2015.0	DHS 2015	
1	Albania	1@	12.0	2018.0	DHS 2017-18	1	2018.0	DHS 2017-18	

cm.shape

Output:

```
(202, 8)
```

cl.isnull().sum()

Output:

```
eries Name      3
Series Code      5
Country Name     5
Country Code     5
```

cm.isnull().sum()

Output:

```
Country Name              0
Female Married by 15     74
Female Married by 18     74
```

#to remove missing values from data

cl=cl.dropna()

cl.duplicated().sum()

cm.duplicated().sum()

cm['Male Married by 18'].duplicated().sum()

cm.loc[cm.duplicated()]

cl = cl.drop_duplicates()

cm=cm.drop_duplicates()

cl.shape

cl=cl.rename(columns={'Country Name':'Country'})

cl.columns

Output:

```
Index(['Series Name', 'Series Code', 'Country', 'Country Code',
       '1990 [YR1990]', '2000 [YR2000]', '2013 [YR2013]', '2014 [YR2014]',
       '2015 [YR2015]', '2016 [YR2016]', '2017 [YR2017]', '2018 [YR2018]',
       '2019 [YR2019]', '2020 [YR2020]', '2021 [YR2021]', '2022 [YR2022]'],
      dtype='object')
```

cm.isnull().sum()

```
cm['Female Married by 18']=cm['Female Married by 18'].fillna(value=0)

cm.tail(30)
```

Output:

	Country Name	Female Married by 15	Female Married by 18	Reference year	Data source	Male Married by 18	Male Reference year	Data source.1	
172	Suriname	9	36.0	2018.0	MICS 2018	20	2018.0	MICS 2018	
173	Sweden	NaN	0.0	NaN	NaN	NaN	NaN	NaN	NaN

```
cm['Reference year']=cm['Reference year'].fillna(method='ffill')

cm.tail(30)

cm['Male Married by 18']=cm['Male Married by 18'].fillna(method='bfill')

cm.tail(30)

cm=cm.dropna()

cl[cl['Country Code']=='AND']

cl=cl.drop(cl[cl['Country Code']=='AND'].index)

new=pd.concat([cl,cm],axis=1)

cl.head()

new2=new.merge(cl,indicator=True,how='outer')

new2

#to remove line breaks columnswise

cl['Country'] = cl['Country'].str.replace('\n', ' ')

#to remove line breaks row-wise

cl.replace('\n','',regex=True)

#to display whitespaces

cm['Data source']=cm['Data source'].str.replace(' ','')

#to remove special characters from columns

cm['Female Married by 15'] = cm['Female Married by 15'].str.replace(r'\W', '', regex=True)
```

cm.head(20)

cm[cm['Female Married by 15'] == '1']

Output:

	Country Name	Female Married by 15	Female Married by 18	Reference year	Data source	Male Married by 18	Male Reference year	Data source.1	
1	Albania	1	12.0	2018.0	DHS2017-18	1	2018.0	DHS 2017-18	
60	Eswatini	1	5.0	2014.0	MICS2014	1	2014.0	MICS 2014	
99	Lesotho	1	16.0	2018.0	MICS2018	2	2018.0	MICS 2018	

XLRD PROGRAM:

```
!pip install xlrd

import xlrd

import re

book = xlrd.open_workbook("SOWC 2014 Stat Tables_Table 9.xls")

sheet = book.sheet_by_name("Table 9 ")

data = {}

for i in range(14, sheet.nrows):


# Start at 14th row, because that is where the countries begin

   row = sheet.row_values(i)

   country = row[1]

   data[country] = {

      'child_labor': {

         'total': [row[4], row[5]],

         'male': [row[6], row[7]],

         'female': [row[8], row[9]],
```

```
    },
    'child_marriage': {
      'married_by_15': [row[10], row[11]],
      'married_by_18': [row[12], row[13]],
    }
  }
if country == "Zimbabwe":
    break
import pprint
pprint.pprint(data)
```

OUTPUT:

```
{'Afghanistan': {'child_labor': {'female': [9.6, ''],
                                 'male': [11.0, ''],
                                 'total': [10.3, '']},
                 'child_marriage': {'married_by_15': [15.0, ''],
                                    'married_by_18': [40.4, '']}},
 'Albania': {'child_labor': {'female': [9.4, '  '],
                             'male': [14.4, '  '],
                             'total': [12.0, '  ']},
             'child_marriage': {'married_by_15': [0.2, ''],
                                'married_by_18': [9.6, '']}},
```

```
# Extract data from the sheet
data = []
for row_idx in range(14, sheet.nrows):  # Skip the header row
  row = sheet.row_values(row_idx)
  data.append(row)

# Check for duplicates
duplicates = []
seen = set()
for row in data:
  row_str = ','.join(map(str, row))
```

```
    if row_str in seen:
        duplicates.append(row)
    else:
        seen.add(row_str)
if duplicates:
    print("Duplicates found:")
    for duplicate_row in duplicates:
        print(duplicate_row)
else:
    print("No duplicates found.")
```

Output:

```
Duplicates found:
['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '', '']
['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '', '']
['', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '', '',
'', '', '', '', '', '',
```

```
# Check for missing data
missing_data_rows = []
for row_idx, row in enumerate(data):
    if any(cell == '' or cell is None for cell in row):
        missing_data_rows.append(row_idx)
if missing_data_rows:
    print("Missing data found in rows:")
    for row_idx in missing_data_rows:
        print(f"Row {row_idx + 2}")  # Adding 2 to account for header and 0-based indexing
else:
    print("No missing data found.")
```

Output:

```
Missing data found in rows:
Row 2
Row 3
Row 4
Row 5
Row 6
Row 7
Row 8
Row 9
Row 10
```

```
# Clean line breaks, spaces, and special characters

def clean_text(text):

    cleaned_text = re.sub(r'\s+', ' ', str(text))  # Replace multiple spaces and line breaks with a single space

    cleaned_text = re.sub(r'[^\w\s]', '', str(cleaned_text))  # Remove special characters

    return cleaned_text.strip()

for row_idx in range(len(data)):

    for col_idx in range(len(data[row_idx])):

        data[row_idx][col_idx] = clean_text(data[row_idx][col_idx])


# Since xlrd doesn't support writing changes to Excel files, you can only display cleaned data

print("Data cleanup complete. Cleaned data:")

for row in data:

    print(row)
```

Output:

```
Data cleanup complete. Cleaned data:
['', 'Afghanistan', 'Afghanistan', 'Afganistán', '103', '', '110', '', '96', '', '150', '', '404', '',
'374', '', '', '', '', '', '', '', '', '', '902', '', '744', '', '748', '', '741', '']
['', 'Albania', 'Albanie', 'Albania', '120', '', '144', '', '94', '', '02', '', '96', '', '986', '',
'', '', '', '', '', '', '364', '', '298', '', '751', '', '783', '', '714', '']
```

EXPERIMENT-4

AIM: **Draw the chart between perceived corruption scores compared to the child labour percentages using matplotlib.**

PROCEDURE:

Step1: Install the Matplotlib package
Step2: Read the required data using read.**csv method of pandas library**
Step3: Define the x-axis and corresponding y-axis values as lists.
Step4: Plot them on canvas using .plot() function.
Step5: Give a name to x-axis and y-axis using .xlabel() and .ylabel() functions.
Step6: Give a title to your plot using .title() function.
Step7: To view your plot, use .show() function.

```
pip install matplotlib

import matplotlib.pyplot as plt

import pandas as pd

cl=pd.read_csv('National Child Labour.csv')

cl
```

OUTPUT:

Out[14]:

	Series Name	Series Code	Country Name	Country Code	1990 [YR1990]	2000 [YR2000]	2013 [YR2013]	2014 [YR2014]	2015 [YR2015]
0	Children in employment, total (% of children a...	SL.TLF.0714.ZS	Afghanistan	AFG	..	..	..	..	..
1	Children in employment, total (% of children a...	SL.TLF.0714.ZS	Albania	ALB	..	36.58605	..	..	..
2	Children in employment, total (% of children a...	SL.TLF.0714.ZS	Algeria	DZA	..	..	7.50076369799783	..	..

```
cpi=pd.read_csv('history.csv')

cpi
```

OUTPUT:

	CPI 2016 Rank	Country	Country Code	Region	CPI 2018 Score	CPI 2019 Score	CPI 2020 Score	CPI 2021 Score	CPI 2022 Score	
0	1	New Zealand	NZL	Asia Pacific	90	88.0	91.0	91.0	90.0	

CPI 2016 Rank	Country	Country Code	Region	CPI 2018 Score	CPI 2019 Score	CPI 2020 Score	CPI 2021 Score	CPI 2022 Score	
1	1	Denmark	DNK	Europe and Central Asia	90	91.0	92.0	91.0	90.0
2	3	Finland	FIN	Europe and Central Asia	89	90.0	89.0	89.0	90.0
3	4	Sweden	SWE	Europe and Central Asia	88	89.0	87.0	89.0	88.0
4	5	Switzerland	CHE	Europe and Central Asia	86	86.0	86.0	85.0	86.0

```
new=cpi.merge(cl, how='inner')

new.head()
```

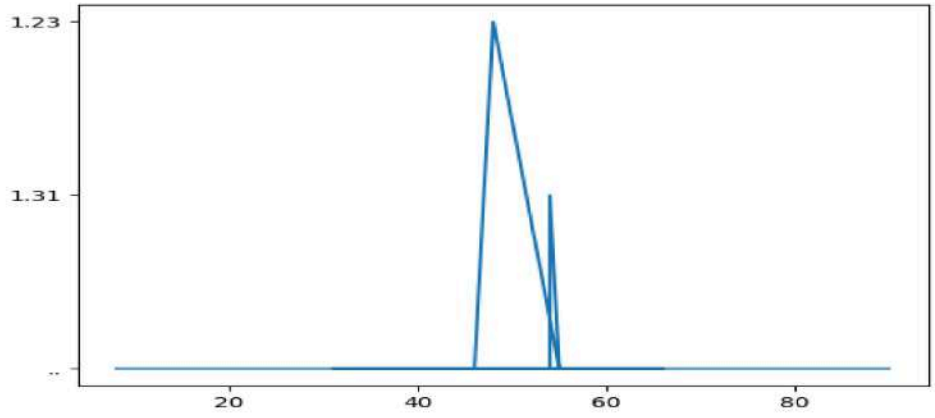
OUTPUT:

Out[17]:

	CPI 2016 Rank	Country	Country Code	Region	CPI 2018 Score	CPI 2019 Score	CPI 2020 Score	CPI 2021 Score	CPI 2022 Score	Series Name	...	2013 [YR2013]	2014 [YR2014]	2015 [YR2015]	2016 [YR2016]	2017 [YR2017]	20 [YR201
0	1	New Zealand	NZL	Asia Pacific	90	88.0	91.0	91.0	90.0	Children in employment, total (% of children a...	...	..	..	..	..	..	
1	1	Denmark	DNK	Europe and Central Asia	90	91.0	92.0	91.0	90.0	Children in employment, total (% of children a...	...	..	..	..	..	..	
2	3	Finland	FIN	Europe and Central Asia	89	90.0	89.0	89.0	90.0	Children in employment, total (% of children a...	...	..	..	..	..	..	

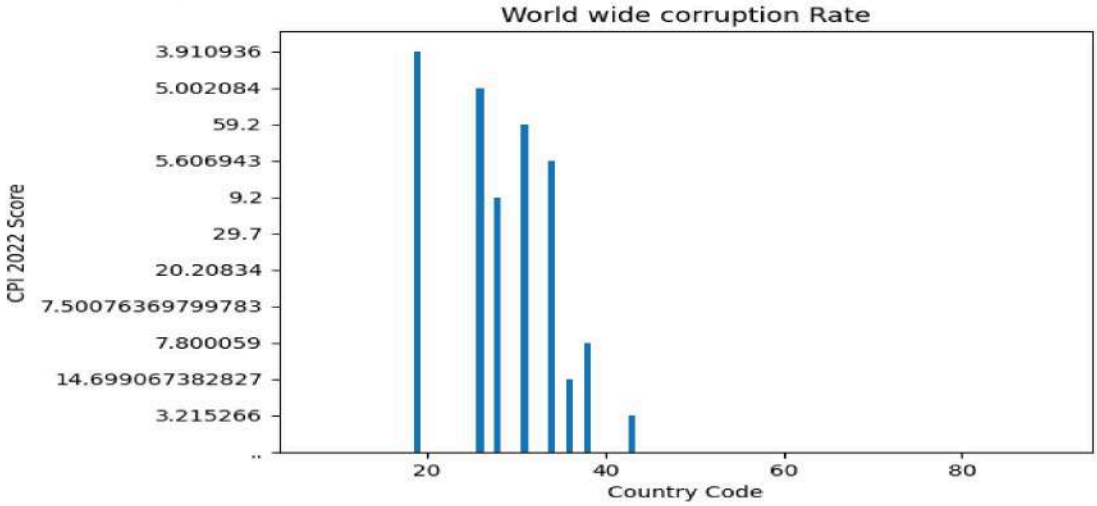
```
plt.plot(new['CPI 2022 Score'],new['2016 [YR2016]'])
```

```
[<matplotlib.lines.Line2D at 0x248c6352b90>]
```



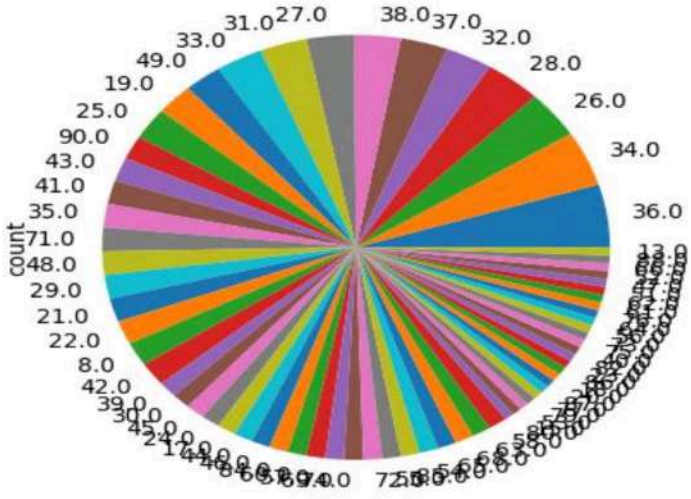
```
x=new['CPI 2022 Score']

y=new['2013 [YR2013]']

plt.title('World wide corruption Rate')

plt.xlabel('Child Labour')

plt.ylabel('CPI 2022 Score')

plt.bar(x,y)
```

```
<BarContainer object of 171 artists>
```



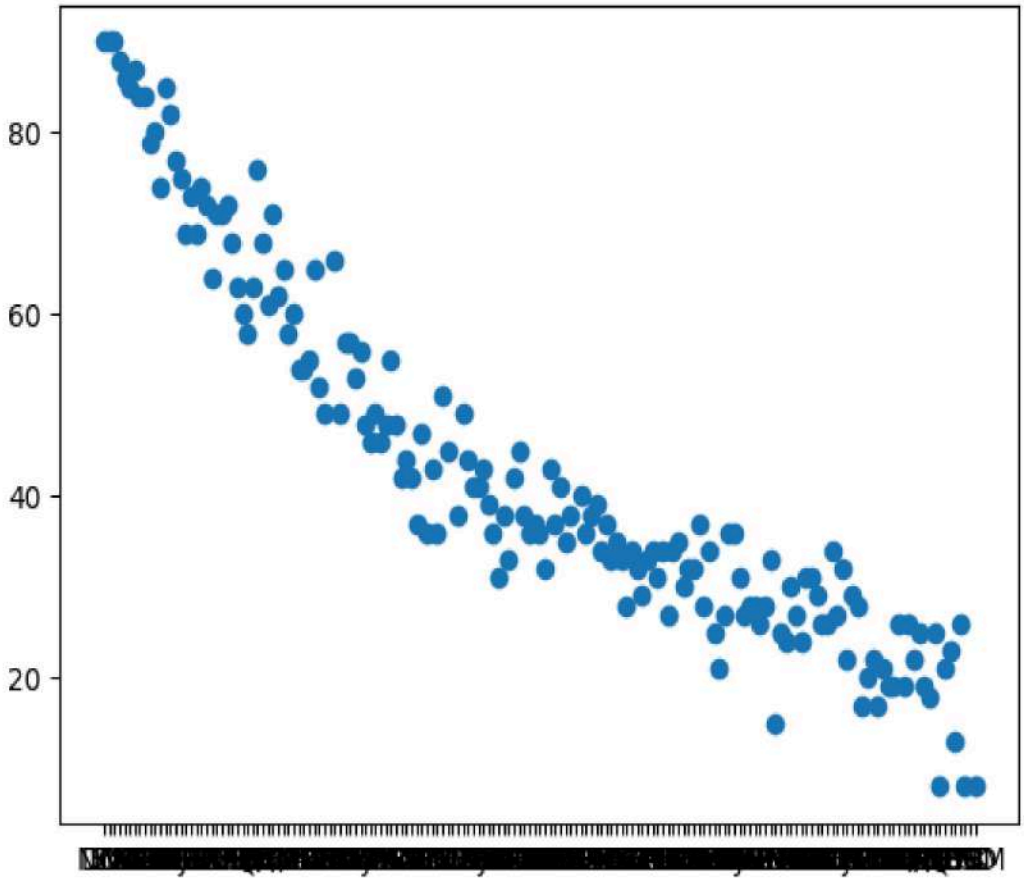

```
new['CPI 2022 Score'].value_counts().plot(kind='pie')
```

```
<Axes: ylabel='count'>
```



plt.scatter(x,y)

```
<matplotlib.collections.PathCollection at 0x248c5f60790>
```



EXPERIMENT-5

AIM: Write a python program to download & display content of robot.txt for en.wikipedia.org.

PROCEDURE:

Step1: Use Requests (HTTP for Humans) Library for Web Scraping
Step2*: Scrape the robots.txt file*
Step3: print the data

```
!pip install requests

import requests

response = requests.get("https://en.wikipedia.org/robots.txt")

test=response.text

print("robots.txt for https://en.wikipedia.org/")

print("==========================================")

print(test.encode("utf-8"))
```

OUTPUT:

```
robots.txt for https://en.wikipedia.org/
==========================================
b'\xef\xbb\xbf# robots.txt for http://www.wikipedia.org/ and friends\n#\n# Please note: There are a
lot of pages on this site, and there are\n# some misbehaved spiders out there that go _way_ too fast.
If you\'re\n# irresponsible, your access to the site may be blocked.\n#\n\n# Observed spamming large
amounts of https://en.wikipedia.org/?curid=NNNNNN\n# and ignoring
```

```
from urllib.request import urlopen

with urlopen("https://en.wikipedia.org/robots.txt") as stream:

   print("robots.txt for https://en.wikipedia.org/")

   print("==========================================")

   print(stream.read().decode("utf-8"))
```

OUTPUT:

```
robots.txt for https://en.wikipedia.org/
==========================================
# robots.txt for http://www.wikipedia.org/ and friends
#
# Please note: There are a lot of pages on this site, and there are
# some misbehaved spiders out there that go _way_ too fast. If you're
# irresponsible, your access to the site may be blocked.
```

```
#url of the robots.txt file for en.wikipedia.org

url="https://en.wikipedia.org/robots.txt"

try:

   #send a HTTP GET request to the URL

   response=requests.get(url)

   #check if the request is successful or not(status code 200)

   if response.status_code==200:

      with open("robots.txt","w",encoding="utf-8") as file:

         file.write(response.text)

      #Display the content of robots.txt file

      print("Content of robots.txt from en.wikipedia.org")

      print(response.text)

      print("robots.txt downloaded successfully and saved as 'robots.txt'")

   else:

      print(f"Failed to retrieve robots.txt.status code:{response.status_code}")

except requests.exceptions.RequestException as e:

         print(f"An error occurred:{e}")
```

OUTPUT:

```
Content of robots.txt from en.wikipedia.org
# robots.txt for http://www.wikipedia.org/ and friends
#
# Please note: There are a lot of pages on this site, and there are
# some misbehaved spiders out there that go _way_ too fast. If you're
# irresponsible, your access to the site may be blocked.
#

# Observed spamming large amounts of https://en.wikipedia.org/?curid=NNNNNN
# and ignoring 429 ratelimit responses, claims to respect robots:
# http://mj12bot.com/
User-agent: MJ12bot
Disallow: /

# advertising-related bots:
User-agent: Mediapartners-Google*
```

EXPERIMENT-6

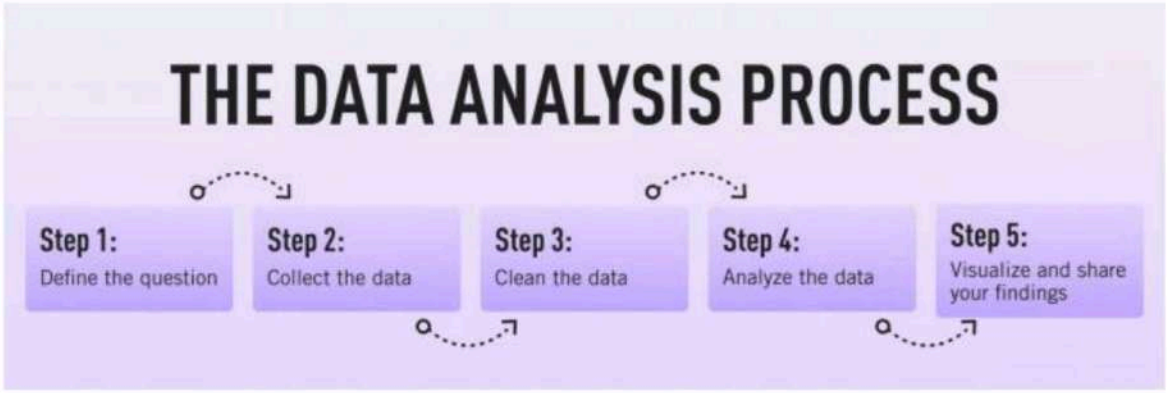
AIM: Foundations for building data visualizations, creating first visualization

PROCEDURE

Step1: Study about basics of data visualization
Step2: Study about prerequisites of data visualization
Step3: Study about different visualization charts for data visualization
Step4: read the required data into tableau using source connection in Tableau
Step5: Study different visualization charts available in Tableau on the data

SOURCE CODE:

- Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations. These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.
- The importance of data visualization is simple: it helps people see, interact with, and better understand data. Whether simple or complex, the right visualization can bring everyone on the same page, regardless of their level of expertise.
- In a nutshell, exploratory data visualization helps you figure out what's in your data, while explanatory visualization helps you to communicate what you've found. Exploration takes place while you're still analyzing the data, while explanation comes towards the end of the process when you're ready to share your findings.



Five data visualization categories

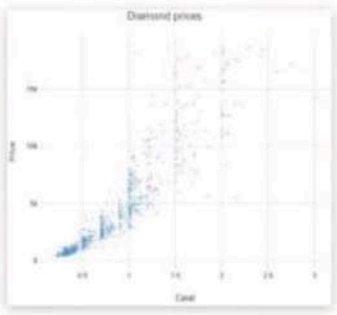
When considering the different types of data viz, it helps to be aware of the different categories that these visualizations may fall into:

- **Temporal data visualizations** are linear and one-dimensional. Examples include scatterplots, timelines, and line graphs.
- **Hierarchical visualizations** organize groups within larger groups, and are often used to display clusters of information. Examples include tree diagrams, ring charts, and sunburst diagrams.
- **Network visualizations** show the relationships and connections between multiple datasets. Examples include matrix charts, word clouds, and node-link diagrams.
- **Multidimensional or 3D visualizations** are used to depict two or more variables. Examples include pie charts, Venn diagrams, stacked bar graphs, and histograms.
- **Geospatial visualizations** convey various data points in relation to physical, real-world locations (for example, voting patterns across a certain country). Examples include heat maps, cartograms, and density maps.

Five common types of data visualization (and when to use them)

1. Scatterplots

Scatterplots (or scatter graphs) visualize the relationship between two variables. One variable is shown on the x-axis, and the other on the y-axis, with each data point depicted as a single "dot" or item on the graph. This creates a "scatter" effect, hence the name.



Scatterplots are best used for large datasets when there's no temporal element. For example, if you wanted to visualize the relationship between a person's height and weight, or between how many carats a diamond measures and its monetary value, you could easily visualize this using a scatterplot.

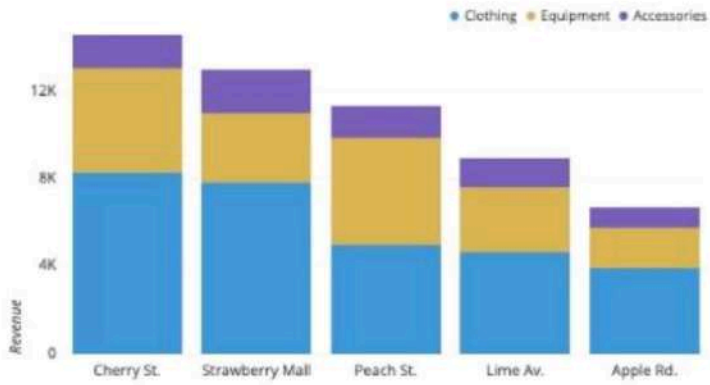
It's important to bear in mind that scatterplots simply describe the correlation between two variables; they don't infer any kind of cause-and-effect relationship.

2. Bar charts

Bar charts are used to plot categorical data against discrete values.

Categorical data refers to data that is not numeric, and it's often used to describe certain traits or characteristics. Some examples of categorical data include things like education level (e.g. high school, undergrad, or post-grad) and age group (e.g. under 30, under 40, under 50, or 50 and over).

Discrete values are those which can only take on certain values—there are no "half measures" or "gray areas." For example, the number of people attending an event would be a discrete variable, as would the number of sales made in a certain time period (think about it: you can't make "half a sale" or have "half an event attendee.")

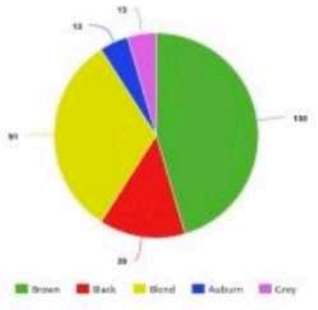


So, with a bar chart, you have your categorical data on the x-axis plotted against your discrete values on the y-axis.The height of the bars is directly proportional to the values they represent, making it easy to compare your data at a glance.

3. Pie charts

Just like bar charts, pie charts are used to visualize categorical data.

However, while bar charts represent multiple categories of data, pie charts are used to visualize just one single variable broken down into percentages or proportions. A pie chart is essentially a circle divided into different "slices," with each slice representing the percentage it contributes to the whole. Thus, the size of each pie slice is proportional to how much it contributes to the whole "pie."



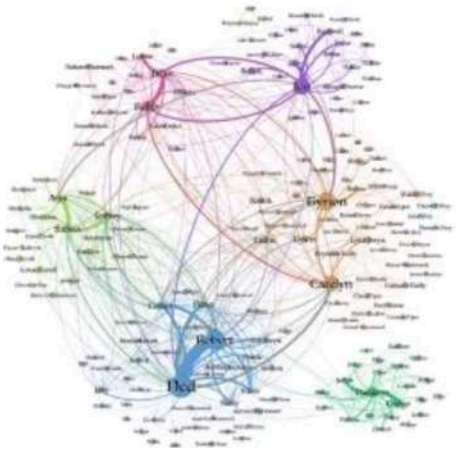
Imagine you have a class of thirty students and you want to divide them up based on what color t-shirt they're wearing on a given day.

The possible "slices" are red, green, blue, and yellow, with each color representing 40%, 30%, 25%, and 5% of the class total respectively. You could easily visualize this using a pie chart—and the yellow slice (5%) would be considerably thinner than the red slice (40%)! Pie charts are best suited for data that can be split into a maximum of five or six categories.

4. Network graphs

Not all data is simple enough to be summarized in a bar or pie chart. For those more complex datasets, there are a range of more elaborate data visualizations at your disposal—network graphs being one of them.

Network graphs show how different elements or entities within a network relate to one another, with each element represented by an individual node. These nodes are connected to other, related nodes via lines.



Network graphs are great for spotting and representing clusters within a large network of data.

Let's imagine you have a huge database filled with customers, and you want to segment them into meaningful clusters for marketing purposes. You could use a network graph to draw connections and parallels between all your customers or customer groups.

With any luck, certain clusters and patterns would emerge, giving you a logical means by which to group your audience.

5. Geographical maps

Geo maps are used to visualize the distribution of data in relation to a physical, geographical area.For example, you could use a color-coded map to see how natural oil reserves are distributed across the world, or to visualize how different states voted in a political election. Maps are an extremely versatile form of data visualization, and are an excellent way of communicating all kinds of location-related data.

Some other types of maps used in data visualization include dot distribution maps (think scatterplots combined with a map), and cartograms which distort the size of geographical areas to proportionally represent a given variable (population density, for example).

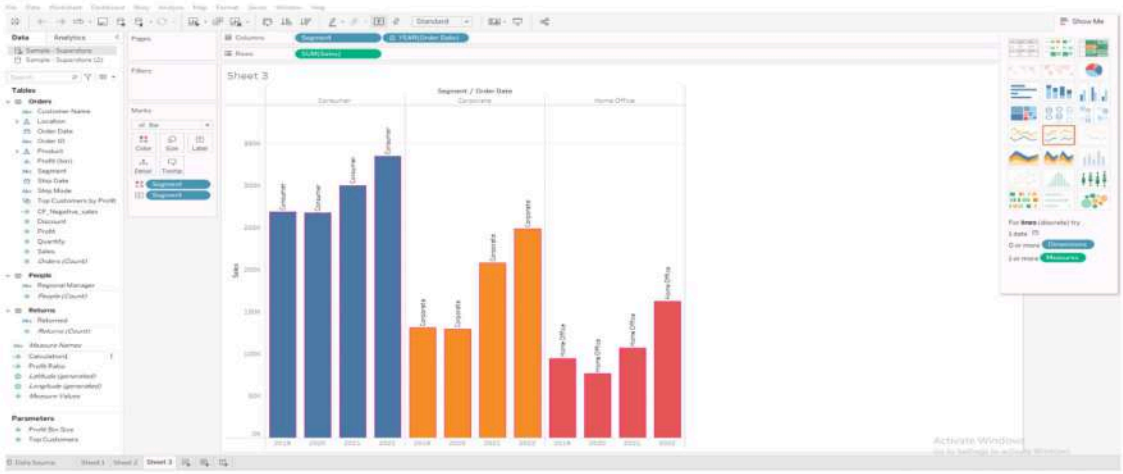
Creating first visualization in Tableau:

Step1 :Open Tableau. On the start page, under Connect load superstore dataset

Step2: After loading the data drag orders from superstore to canvas to explore data

Step3: Click the sheet tab to go to the new work sheet and begin your analysis

For sample drag required columns and rows into required fields default chart will create based on the data.



EXPERIMENT- 7

AIM: Getting started with tableau software using data file formats, connecting data to tableau, creating basic charts (line, bar charts, tree maps) using the show me panel.

PROCEDURE:

Step1: Study about overview of Tableau

Step2: To open the application, click the Tableau icon on your desktop (or in your Start menu).

Step3: In the Connect panel at the left side of the Start page, click the Excel link under the "To a File" heading to the open file selection option.

Step4: Using the file selection box, select the Excel worksheet that you want to open, and then click the Open button to continue

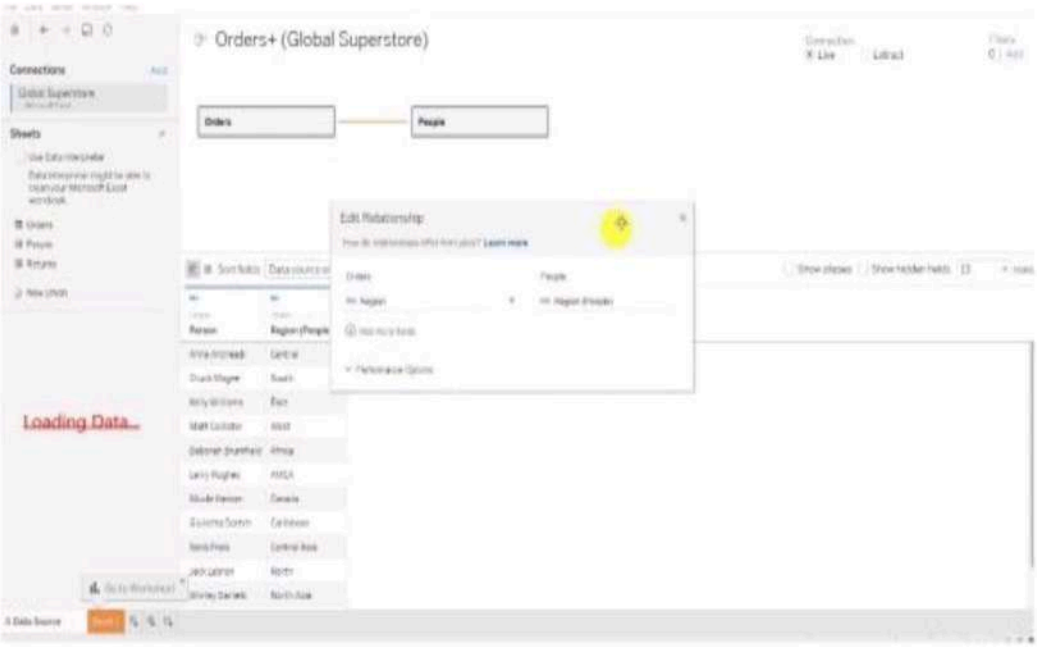
Step5: Drag the dimension and measure in row and column input field and it will automatically suggest a graph best fitted on data.

Step6: You can change the graph by clicking on the show me button and select whichever graph you want.

SOURCE CODE:

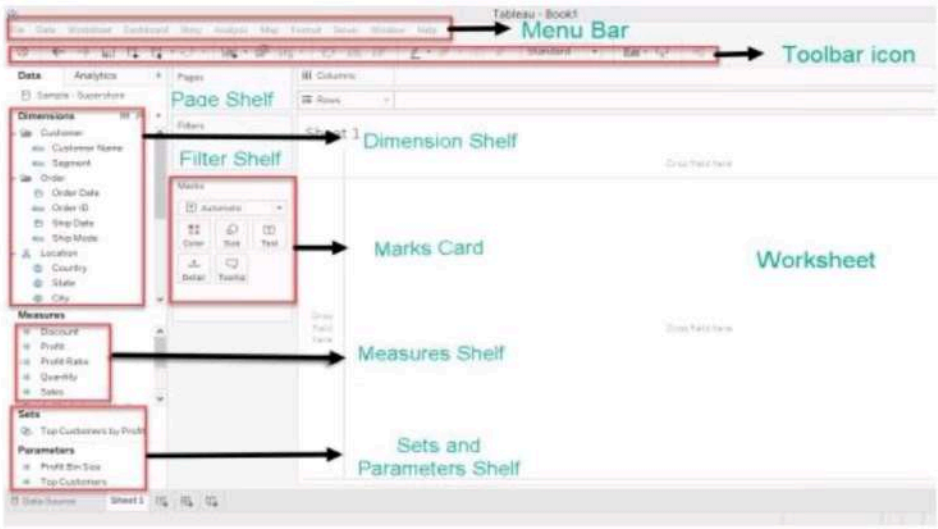
Connect Tableau to the data file:

1. To open the application, click the Tableau icon on your desktop (or in your Start menu).
2. In the Connect panel at the left side of the Start page, click the superstore from saved datasets
3. Select the Orders sheet from the navigation menu on the left and drag it onto
4. After loading we can perform data cleaning, data preprocessing, feature extraction to some extent.

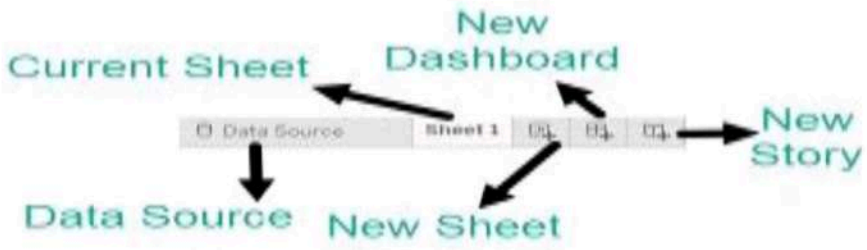


Understanding different Sections in Tableau:

Tableau work-page consist of different section. Let's understand them first before plotting our graphs



- **Menu Bar:** Here you'll find various commands such as **File**, **Data,** and **Format**.
- **Toolbar Icon:** The toolbar contains a number of buttons that enable you to perform various tasks with a click, such as Save, Undo, and New Worksheet.
- **Dimension Shelf:** This shelf contains all the categorical columns under it. example: categories, segments, gender, name, etc
- **Measure Shelf:** This shelf contains all numerical columns under it like profit, total sales, discount, etc
- **Page Shelf:** This shelf is used for joining pages and creates animations. we will come on it later
- **Filter Shelf:** You can choose which data to include and exclude using the Filters shelf, for example, you might want to analyze the profit for each customer segment, but only for certain shipping containers and delivery times. You may make a view like this by putting fields on the Filters tier.
- **Marks Card:** The visualization can be designed using the Marks card. The markings card can be used to change the data components of the visualization, such as color, size, shape, path, label, and tooltip.
- **Worksheet:** In the workbook, the worksheet is where the real visualization may be seen. The worksheet contains information about the visual's design and functionality.



- **Data Source:** Using Data Source we can add new data, modify, remove data.
- **Current Sheet:** The current sheets are those sheets which we have created and to those, we can give some names.
- **New Sheet:** If we want to create a new worksheet (blank canvas) we can do using this tab.
- **New Dashboard:** This button is used to create a dashboard canvas.
- **New Storyboard:** It is used to create a new story

Creating Visuals in Tableau

Tableau supports the following data types:

Boolean: True and false can be stored in this data type.

Date/Date time: This data type can help in leveraging Tableau's default date hierarchy behavior when applied to valid date or Date Time fields.

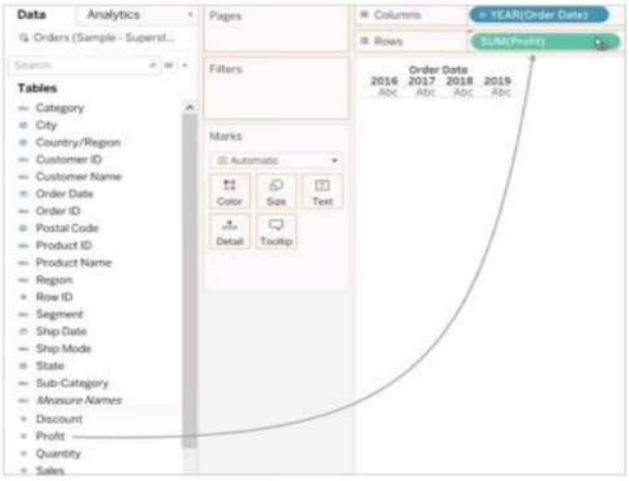
Number: These are values that are numeric. Values can be integers or floating-point numbers (numbers with decimals).

String: This is a sequence of characters encased in single or double quotation marks.

Geolocation: These are values that we need to plot maps.

Follow these steps to create a visual:

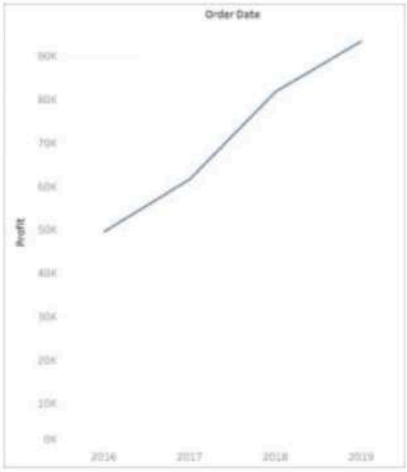
1. Drag the dimension and measure in row and column input field and it will automatically suggest a graph best fitted on data.
2. you can change the graph by clicking on the **show me** button and select whichever graph you want.
3. you can also remove the axis just by dragging and dropping them under the marks card (remove field).
4. **Show Me**: When you click this label, a palette appears, giving you rapid access to many options for showing the selected types of fields. The palette changes depending on the fields in the worksheet you've selected or are active.



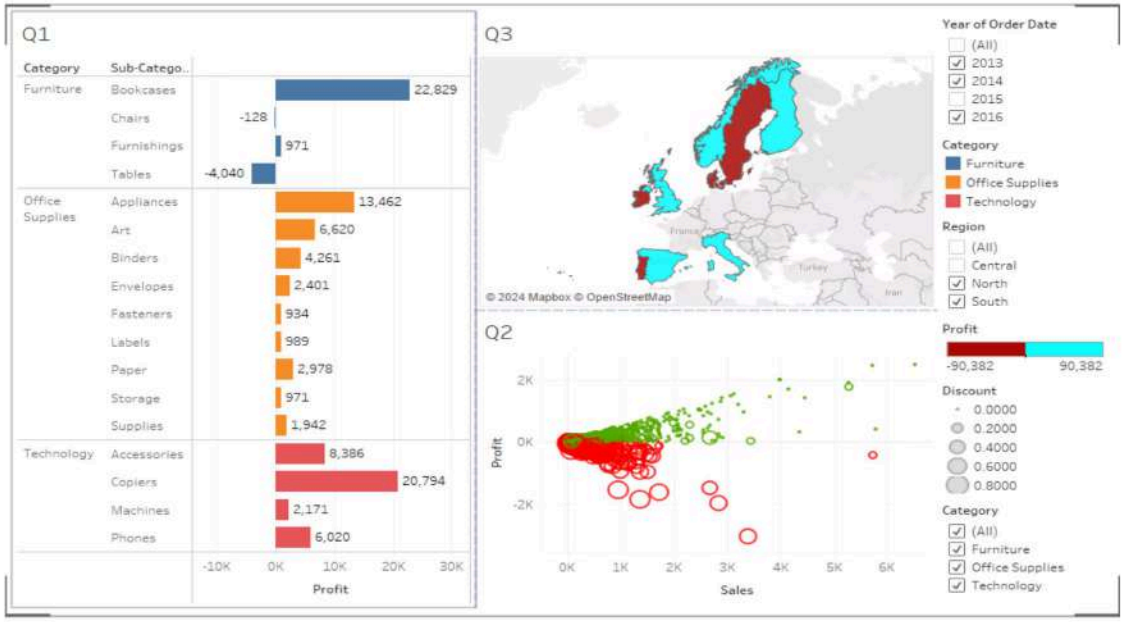
The Profit field is colored green on the Rows shelf, indicating that it's continuous. Also, the field name changes to SUM(Profit). That's because measures are automatically aggregated when you add them to the view, and the default aggregation for this measure is SUM

Tableau transforms the table into a line chart and creates a vertical axis (along the left side) for the measure.

A line chart is a great way to compare data over time and identify trends effectively.



This line chart shows profit over time. Each point along the line shows the sum of profit for the corresponding year



EXPERIMENT- 8

AIM: Tableau calculations, overview of SUM, AVG and aggregate features, creating custom Calculations and fields.

PROCEDURE:

Step1: create the calculated field

a. In a worksheet in Tableau, select Analysis > Create Calculated Field.
b. In the Calculation Editor that opens, give the calculated field a name

Step2: Enter a formula

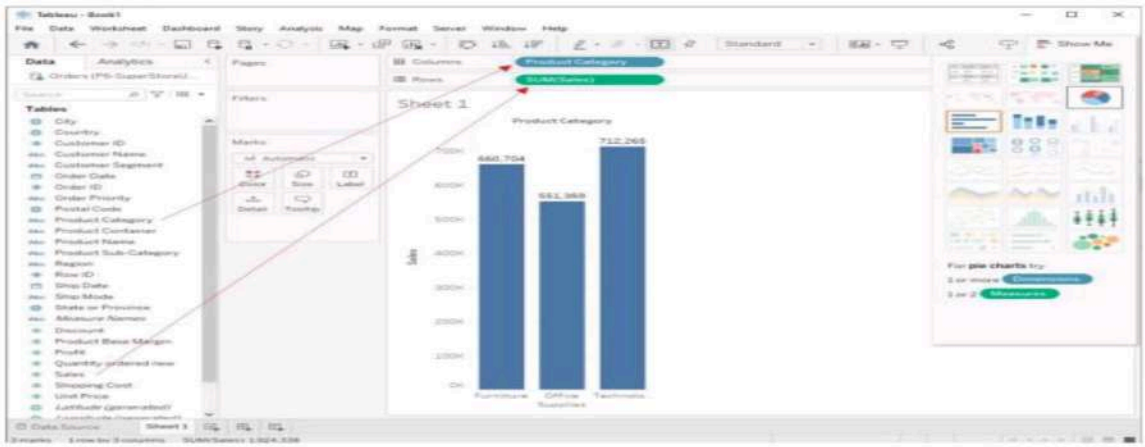
a. In the Calculation Editor, enter a formula.
   Example for how to enter formula:
   SUM([Profit])/SUM([Sales])
b. When finished, click OK.
   The new calculated field is added to the Data pane

Step3: study the overview of SUM, AVG and aggregate functions

SUM and AVG (Average) Functions

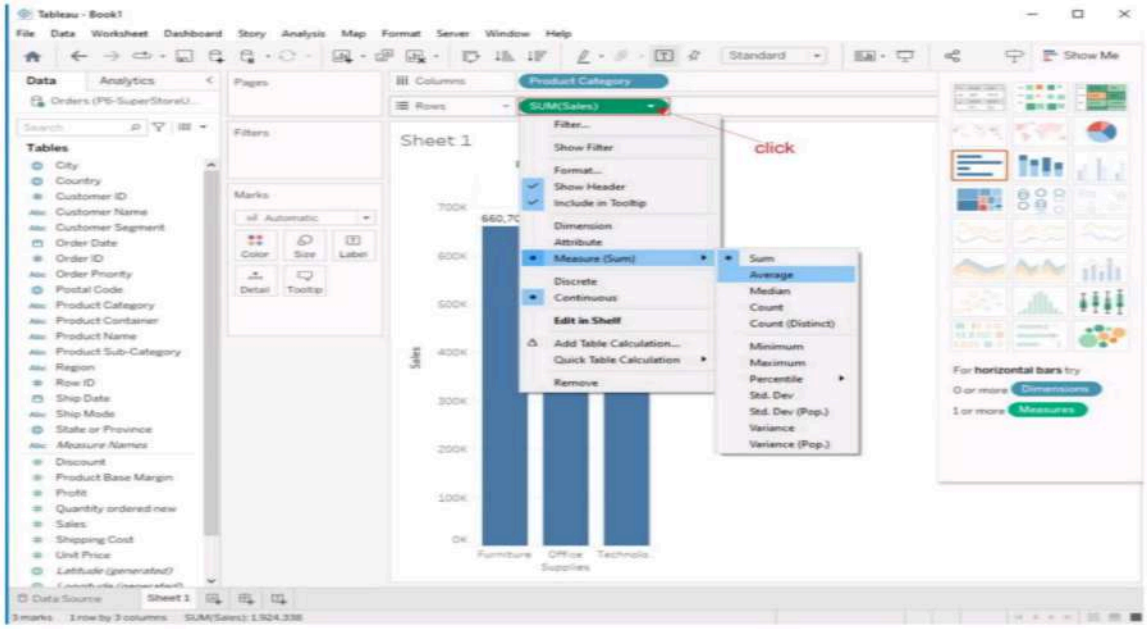
SUM Function

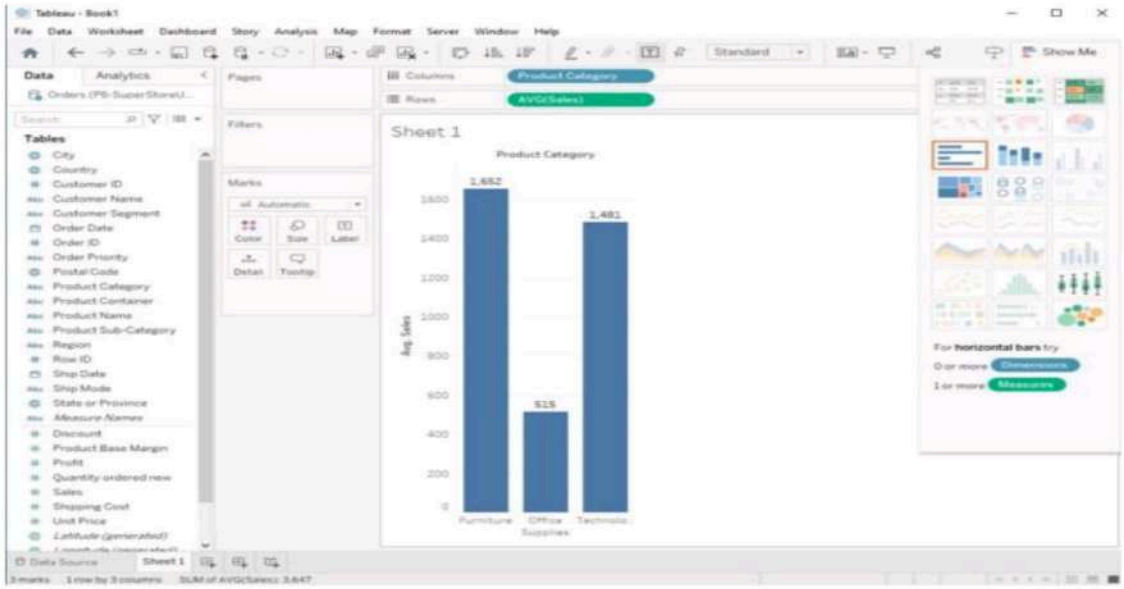
The SUM function in Tableau calculates the total sum of a numeric field. You can use it to find the sum of values in a column or as part of a more complex calculation. To use SUM, simply drag and drop a numeric field into the "SUM" shelf, or you can create a calculated field using the SUM function.



AVG (Average) Function

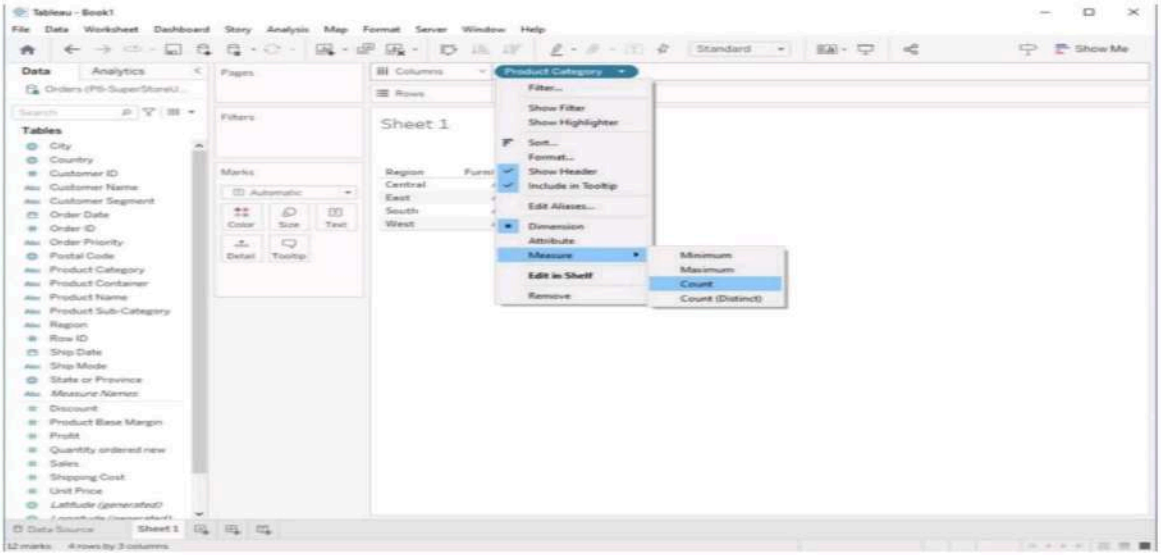
The AVG function calculates the average (mean) value of a numeric field. Like SUM, you can use it by dragging a numeric field into the "AVG" shelf or creating a calculated field with the AVG function.

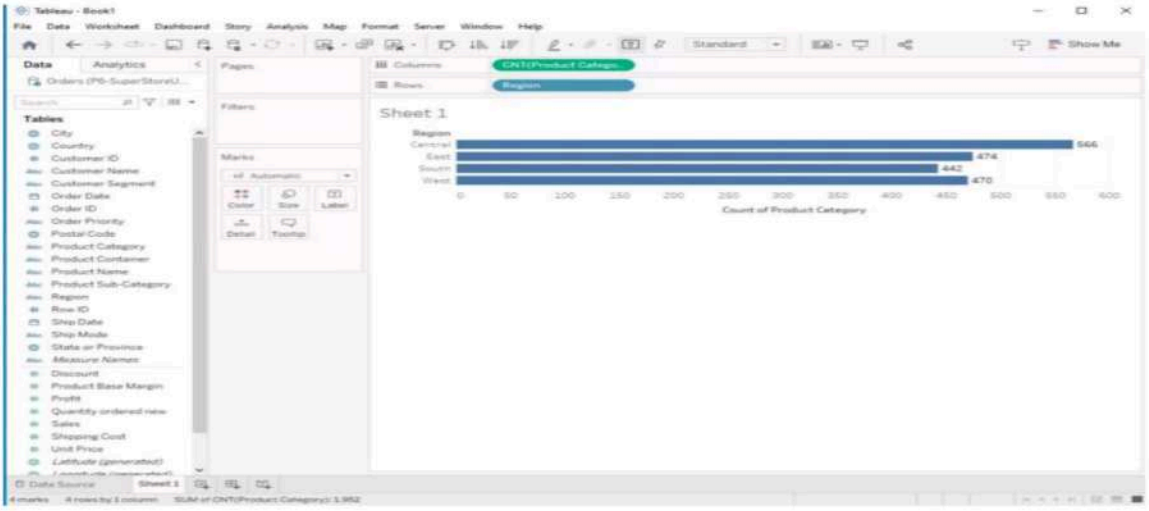




Aggregate Functions:

Tableau provides a range of aggregate functions that allow you to perform calculations on groups of data. Common aggregate functions include SUM, AVG, COUNT, MIN (minimum value), and MAX (maximum value). These functions are particularly useful when you want to analyze data at different levels of granularity (e.g., by category, region, or time period).



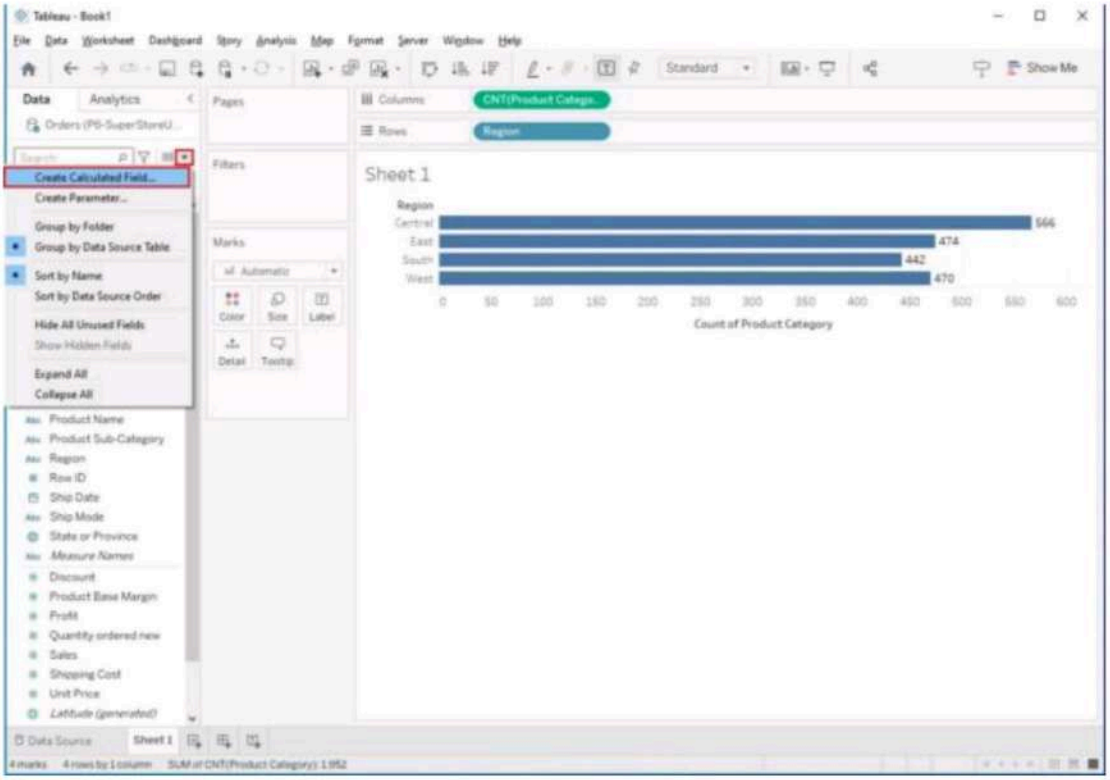


Creating Custom Calculations

Tableau allows you to create custom calculations using calculated fields. Here's how to create a custom calculation:

1. Create a New Calculated Field

In the Data Source Pane, right-click on your data source and select "Create Calculated Field".



Alternatively, you can create a calculated field by right-clicking on a shelf in your worksheet and choosing "Create Calculated Field".

2. Enter Your Calculation:

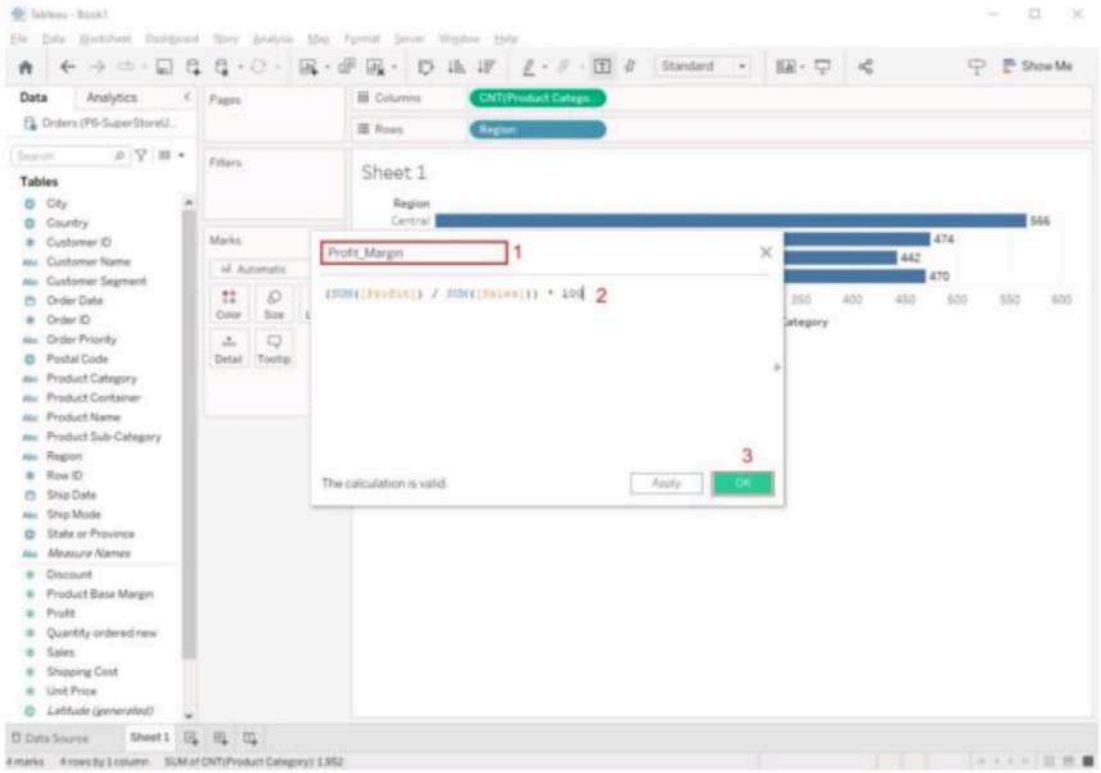
In the calculated field editor, you can use functions, operators, and field references to define your calculation.

For example, you can create a calculated field to calculate profit margin as (SUM([Profit]) / SUM([Sales])) * 100.

3. Name and Save the Calculated Field:

Give your calculated field a meaningful name.

Click the "OK" or "Apply" button to save the calculated field.



4. Use the Calculated Field in Your Worksheet:

You can now use the calculated field like any other field in your worksheet. Drag it to the Rows or Columns shelf, use it in filters, or create visualizations based on it.

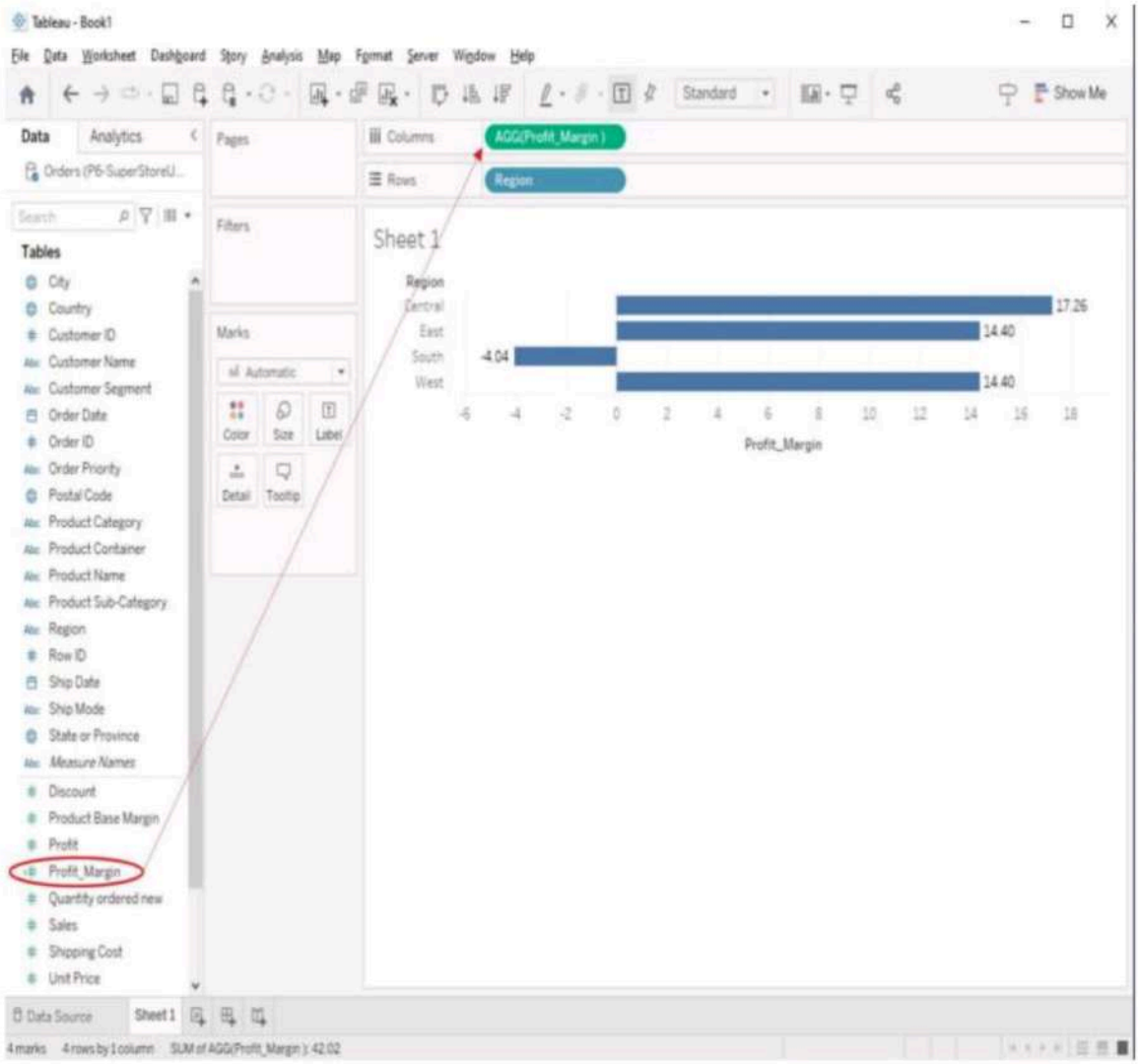


Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Show Me
Data
Analytics
Orders (P6-SuperStoreU...
Search
Tables
City
Country
Customer ID
Customer Name
Customer Segment
Order Date
Order ID
Order Priority
Postal Code
Product Category
Product Container
Product Name
Product Sub-Category
Region
Row ID
Ship Date
Ship Mode
State or Province
Measure Names
Discount
Product Base Margin
Profit
Profit_Margin
Quantity ordered new
Sales
Shipping Cost
Unit Price
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
Columns
AGG(Profit_Margin)
Rows
Region
Sheet 1
Region
Central
East
South
West
17.26
14.40
-4.04
14.40
-6
-4
-2
0
2
4
6
8
10
12
14
16
18
Profit_Margin
Data Source
Sheet 1
4 marks
4 rows by 1 column
SUM of AGG(Profit_Margin): 42.02

EXPERIMENT-9

AIM: Applying new data calculations to visualizations, formatting visualizations, formatting tools and menus, formatting specific parts of the view.

PROCEDURE

I. Applying new data calculations to visualizations

Step1: Build the view

a. From Dimensions, drag Required field to the Columns shelf.

b. From Dimensions, drag Required field to the Rows shelf.

c. On the Rows shelf, click the plus icon (+) on the Category field to drill-down to Sub category.

Step2: Add the calculated field to the view

From Measures, drag Required Field to Color on the Marks card.

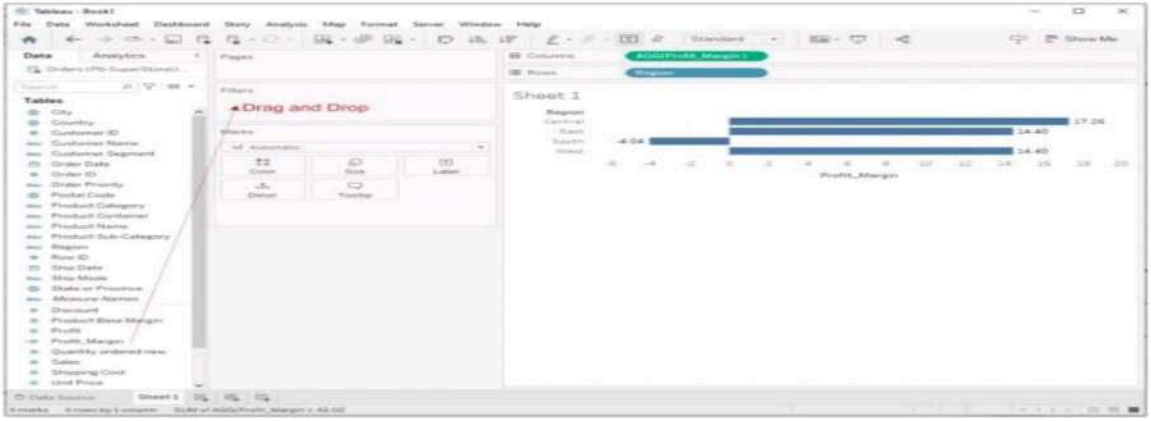
On the Rows shelf, right-click select required field and

Select Measure (Sum) > Average.

Applying New Data Calculations to Visualizations

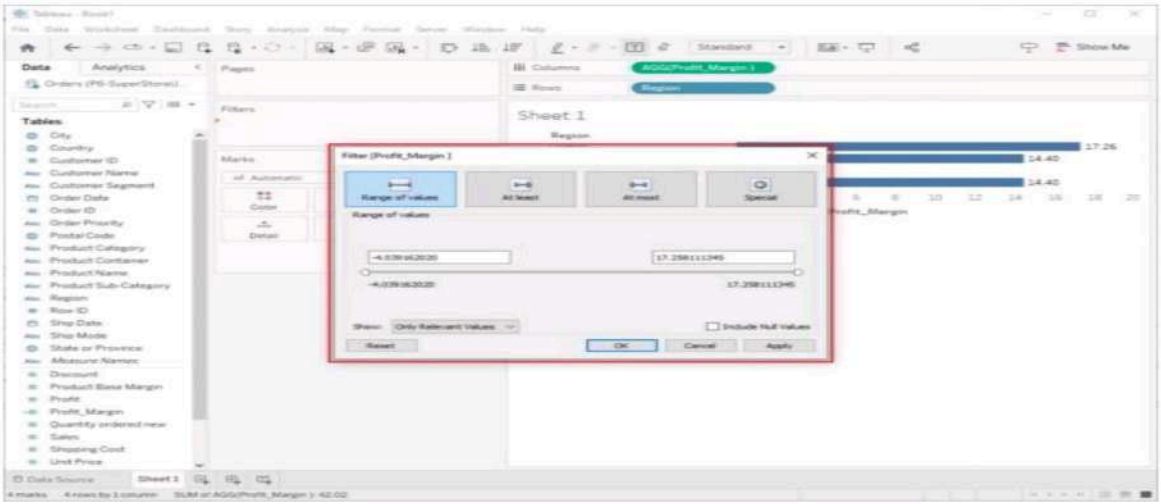
1. Drag and Drop Calculated Fields:

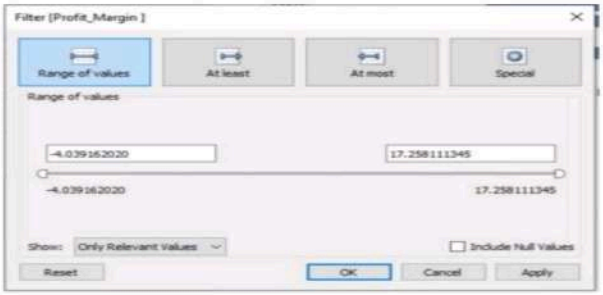
To apply your newly created calculated fields to a visualization, simply drag and drop them onto the appropriate shelves in your worksheet. For example, you can drag a calculated field to the Rows or Columns shelf, use it in filters, or place it on the Marks card to control the appearance of marks.

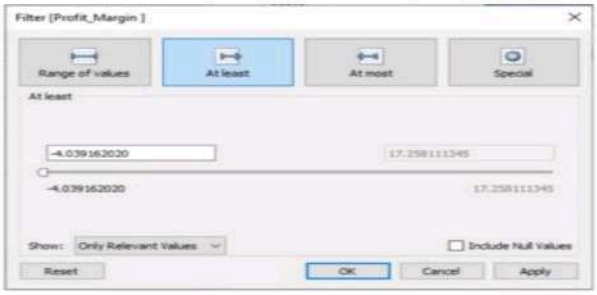


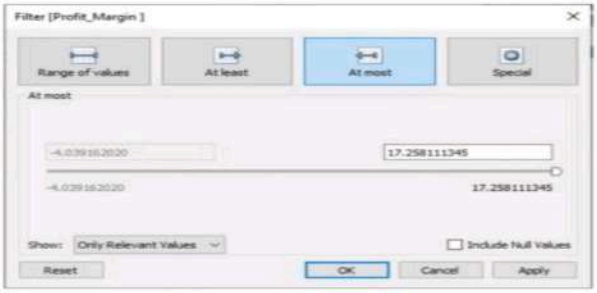
2. Filter with Calculated Fields:

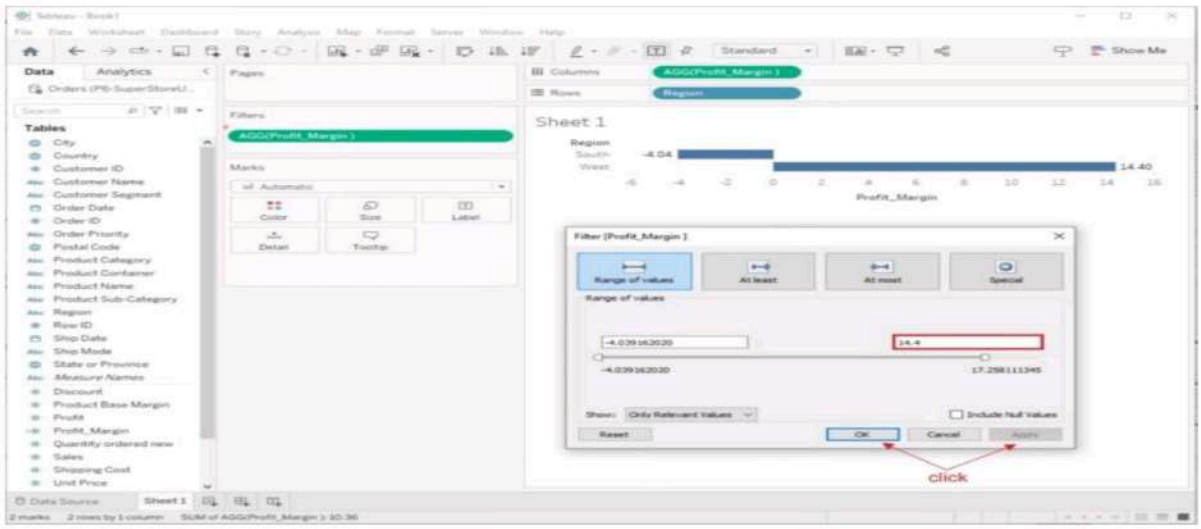
Create filters using calculated fields to control which data points are displayed in your visualization. You can use calculated fields to filter by specific criteria, such as a calculated date range or a custom ranking.









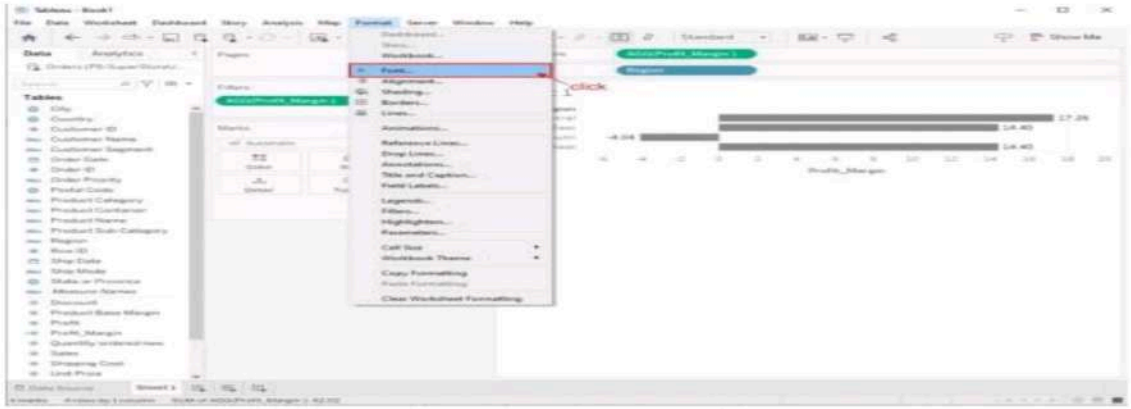


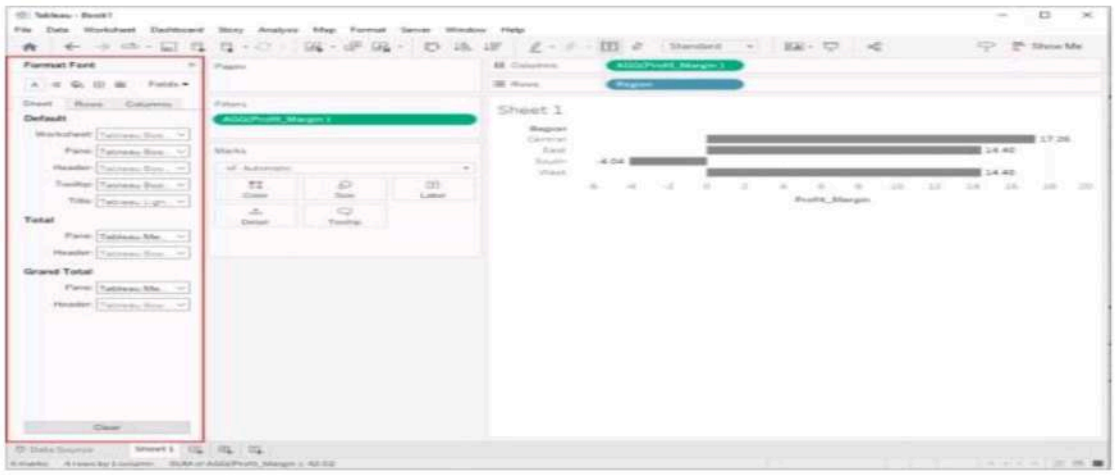
Formatting Visualizations

Tableau provides a wide range of formatting options to make your visualizations more appealing and informative:

1. Format Pane:

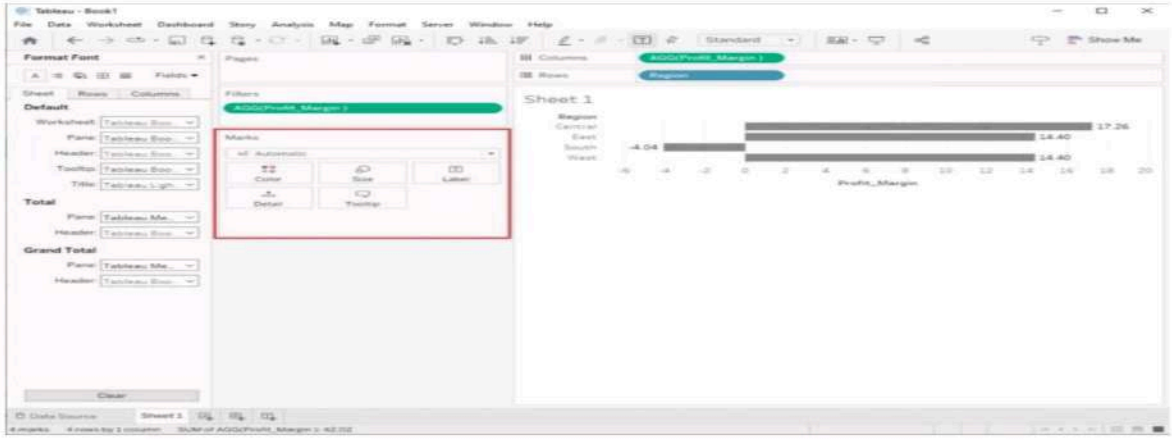
On the left side of the Tableau interface, you'll find the Format pane. It allows you to format various aspects of your visualization, such as fonts, colors, lines, shading, and borders. Simply select the element you want to format and use the options in the Format pane to make changes.





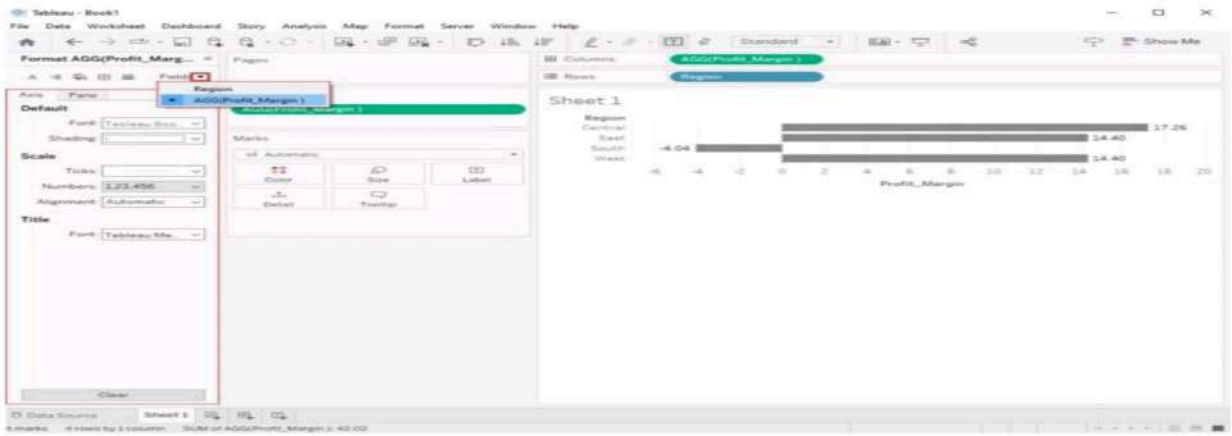
2. Marks Card:

The Marks card, located above your visualization, offers formatting options specific to the type of marks you're using (e.g., color, size, label). Click on the Marks card to access these options and modify how your data is represented.



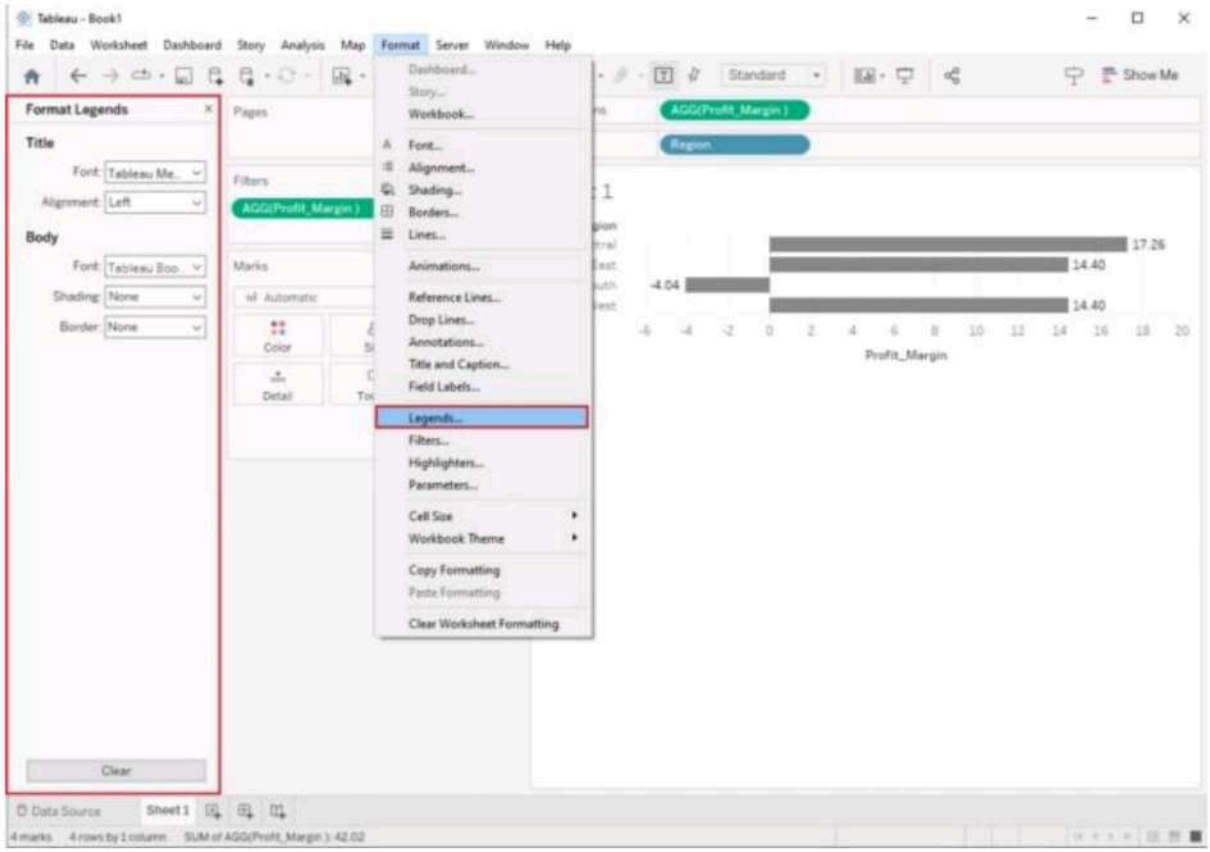
3. Axis and Gridlines:

You can format axis labels, titles, and gridlines to improve the readability of your visualization. Right-click on an axis or gridline to access formatting options.



4. Legends and Color Scales:

Customize legends and color scales to provide context for your visualizations. You can change colors, labels, and the position of legends to match your data.

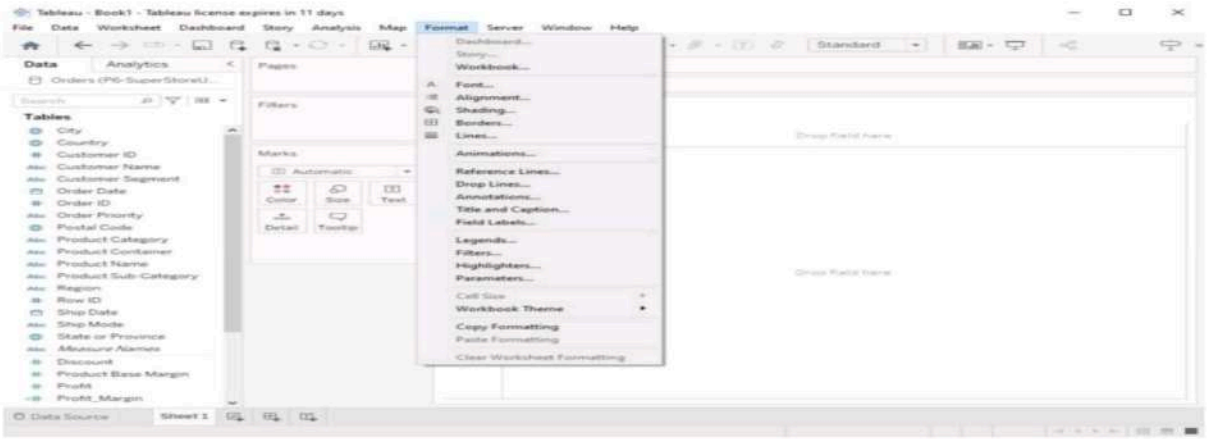


Formatting Tools and Menus

Tableau provides several formatting tools and menus to help you refine the appearance of your visualizations:

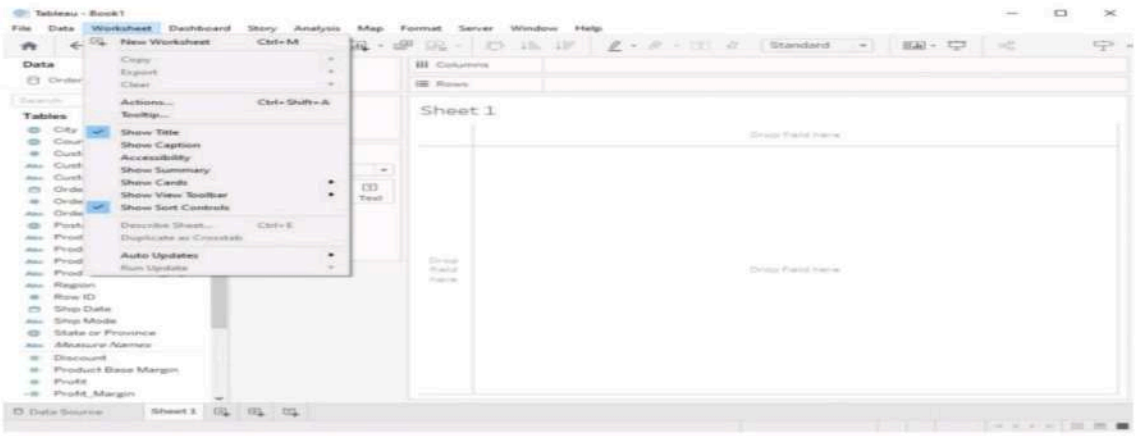
1. Format Menu:

The Format menu at the top of the Tableau interface provides access to various formatting options, including font styles, shading, borders, alignment, and more. You can use this menu to format text, labels, and other elements.



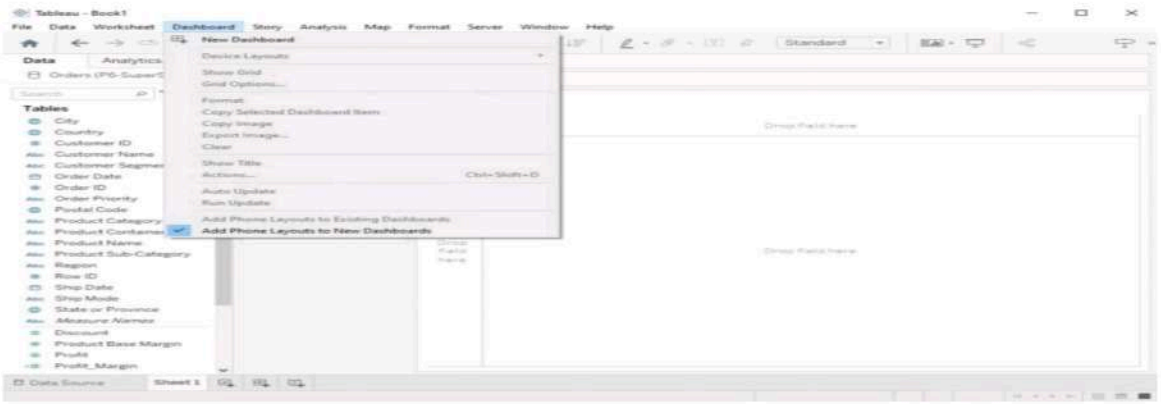
2. Worksheet Menu:

In the Worksheet menu, you'll find options to format the entire worksheet, including background color, borders, and worksheet title. You can also adjust the worksheet size.



3. Dashboard Menu:

If you're working with dashboards, the Dashboard menu allows you to format the entire dashboard layout, including background, size, and title.

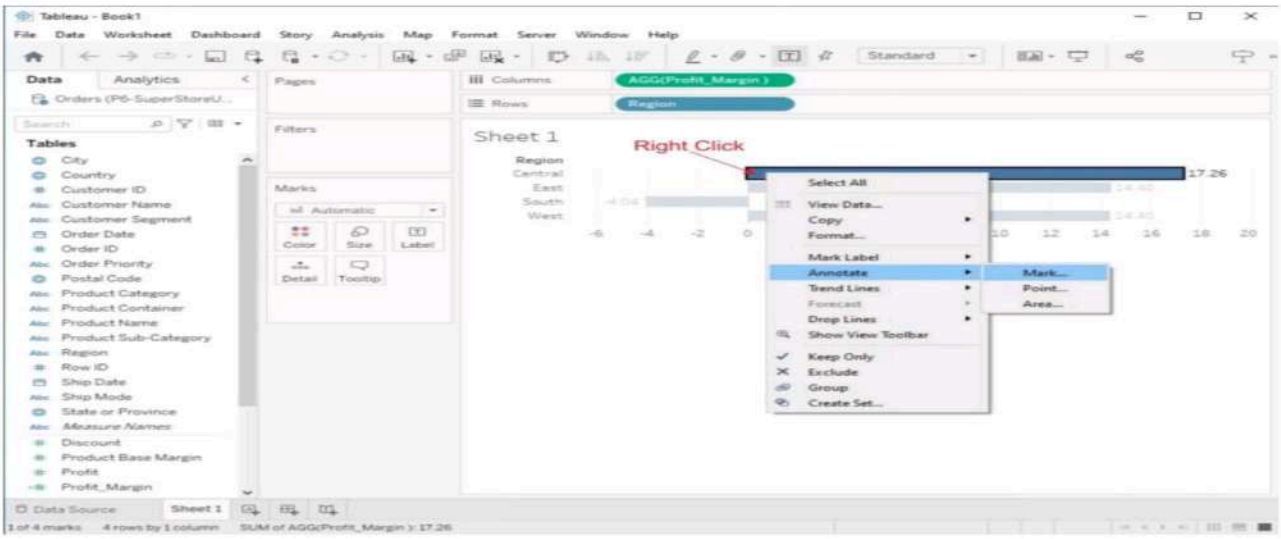


Formatting Specific Parts of the View

Tableau lets you format specific elements of your visualization:

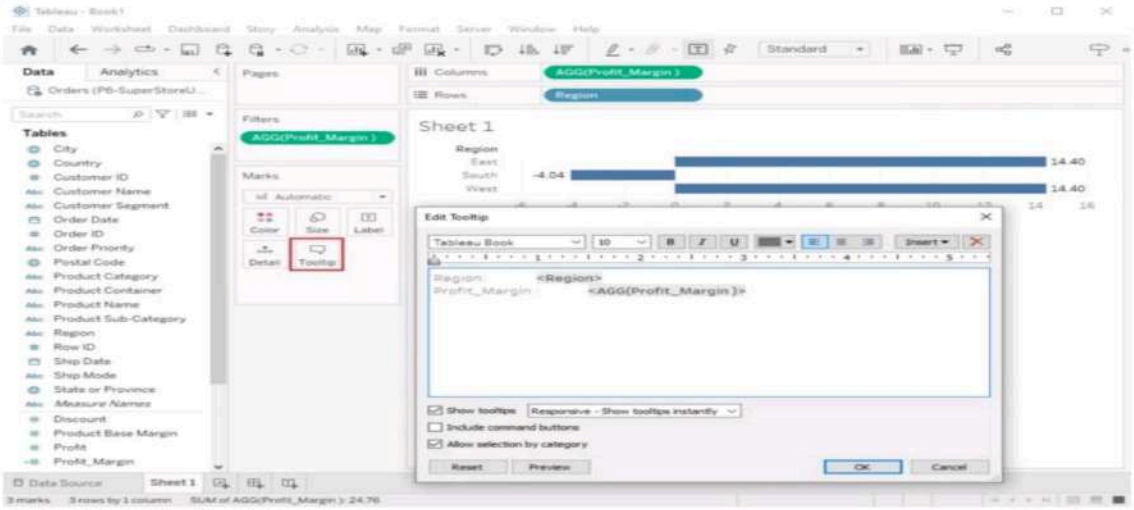
1. Annotations:

You can add annotations to your visualizations to highlight important points or provide additional context. Format these annotations using the options available when you right-click on an annotation.



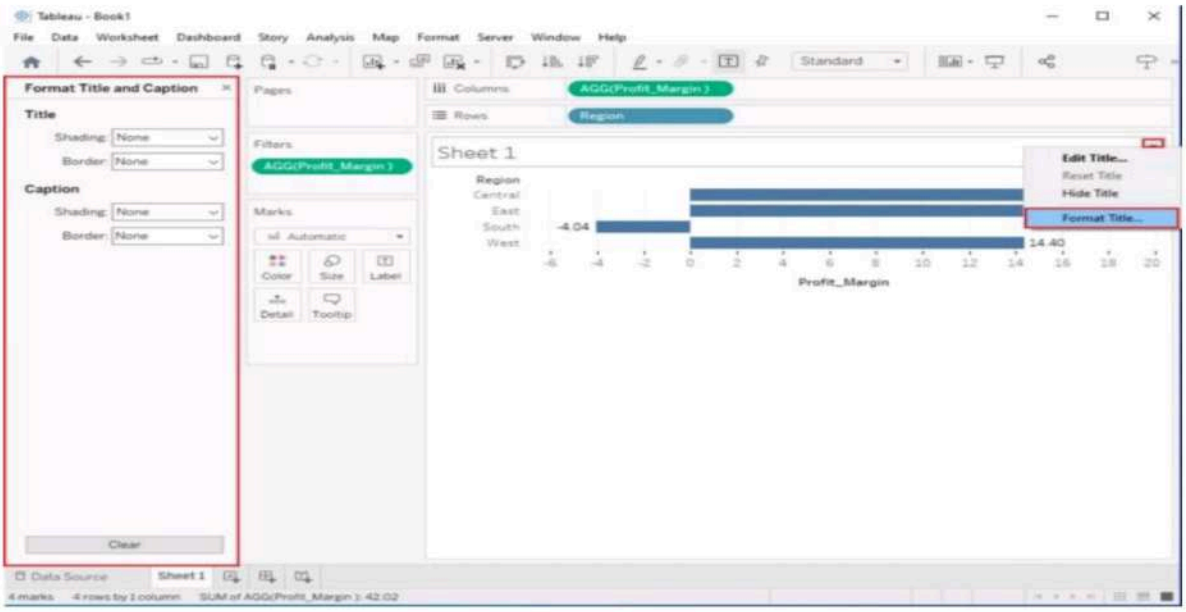
2. Tooltips:

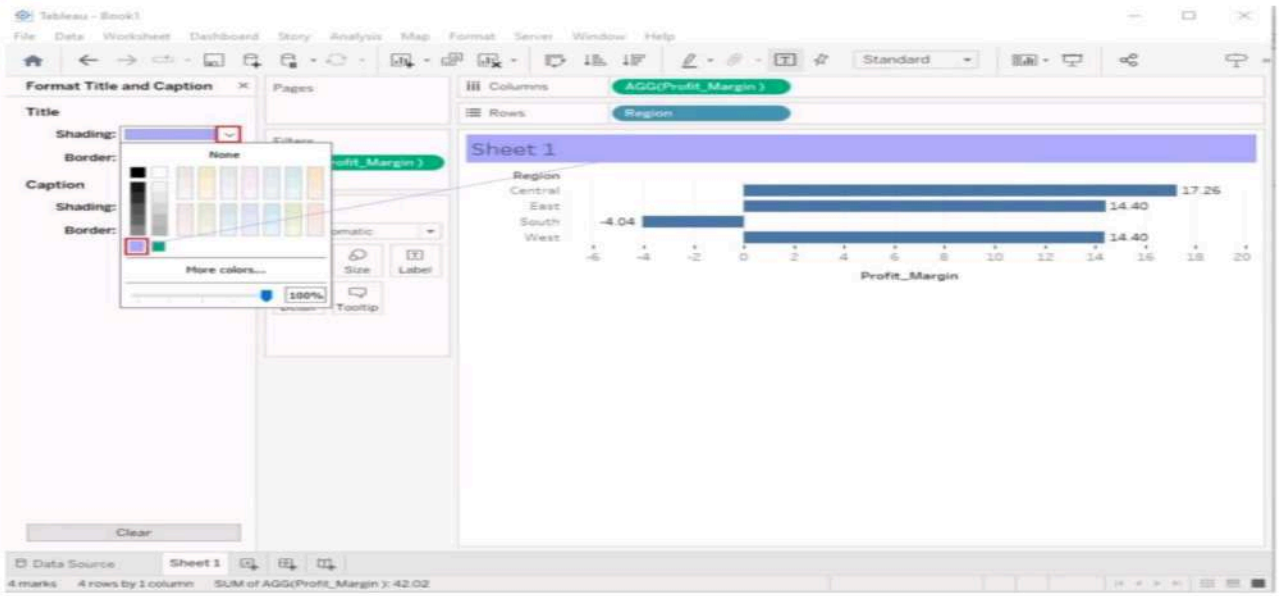
Customize tooltips to display relevant information when users hover over data points. You can format tooltips to show or hide specific fields and control their appearance.

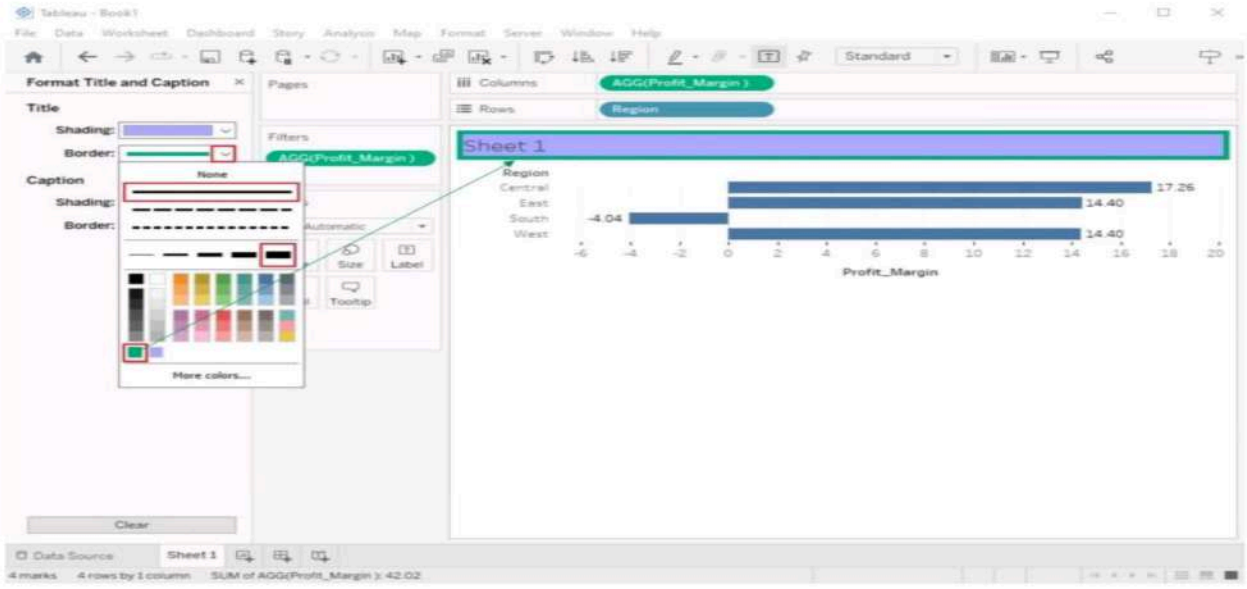


3. Headers and Titles:

Format headers, titles, and subtitles for clarity and consistency. Use the Format pane or the Format menu to adjust text formatting, alignment, and shading.




Format Title and Caption
Title
Shading:
Border:
Caption
Shading:
Border:
None
More colors...
100%
Pages
Columns
AGG(Profit_Margin)
Rows
Region
Sheet 1
Region
Central
East
South
West
17.26
14.40
-4.04
14.40
Profit_Margin
Size
Label
Tooltip
Clear
Data Source
Sheet 1
4 marks
4 rows by 1 column
SUM of AGG(Profit_Margin): 42.02


Format Title and Caption
Title
Shading:
Border:
Caption
Shading:
Border:
None
More colors...
Pages
Columns
AGG(Profit_Margin)
Rows
Region
Sheet 1
Region
Central
East
South
West
17.26
14.40
-4.04
14.40
Profit_Margin
Size
Label
Tooltip
Clear
Data Source
Sheet 1
4 marks
4 rows by 1 column
SUM of AGG(Profit_Margin): 42.02

EXPERIMENT-10

AIM: Editing and formatting axes, manipulating data in tableau data, pivoting tableau data.

PROCEDURE:

I. Editing and Formatting Axes:

Step1: Double-click an axis to open the Edit Axis dialog box and change the axis configuration and formatting

Step2: To select the marks associated with the axis, right-click the axis and select Marks.

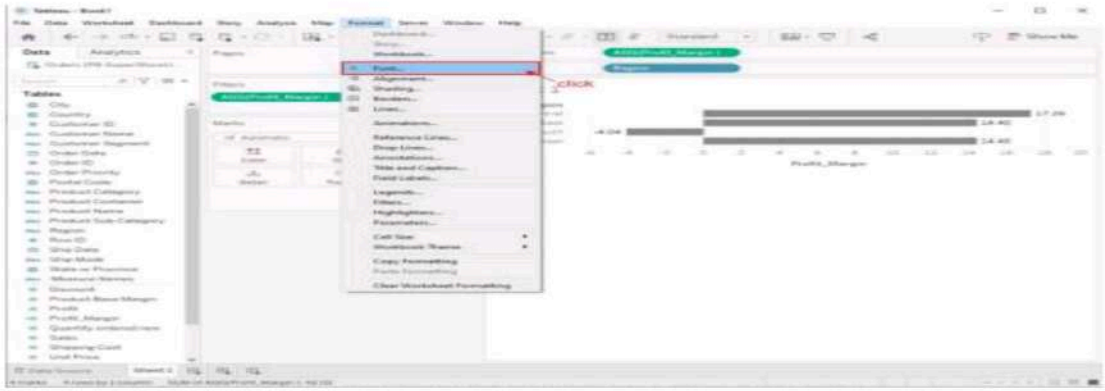
Step3: To hide an axis Right-click (control-click on windows) the axis in the view, and then clear the check mark next to the Show Header option.

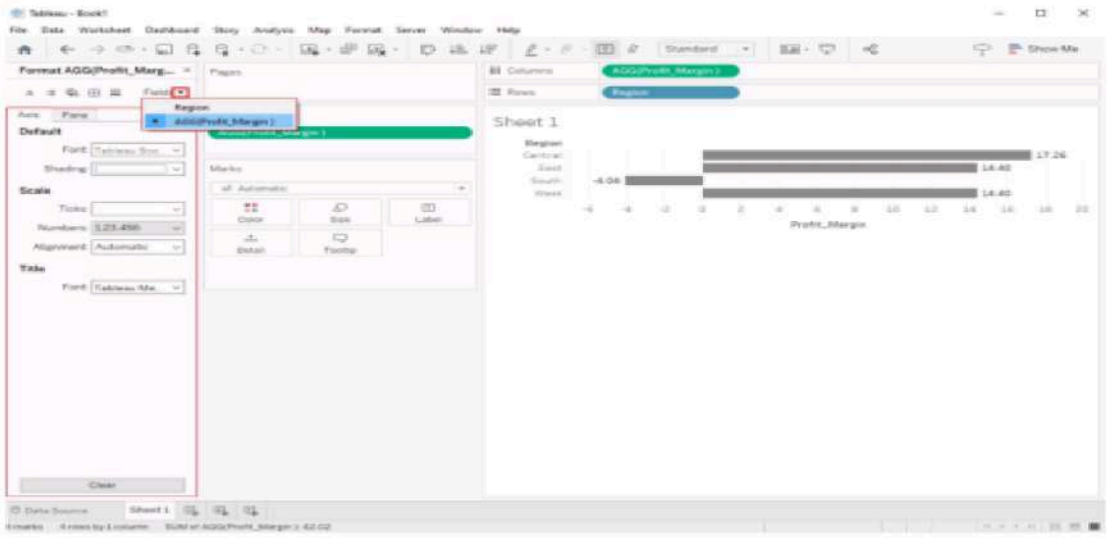
Step4: Try remaining options of axis

Editing and Formatting Axes, Manipulating Data in Tableau data, Pivoting Tableau data.

Solution :

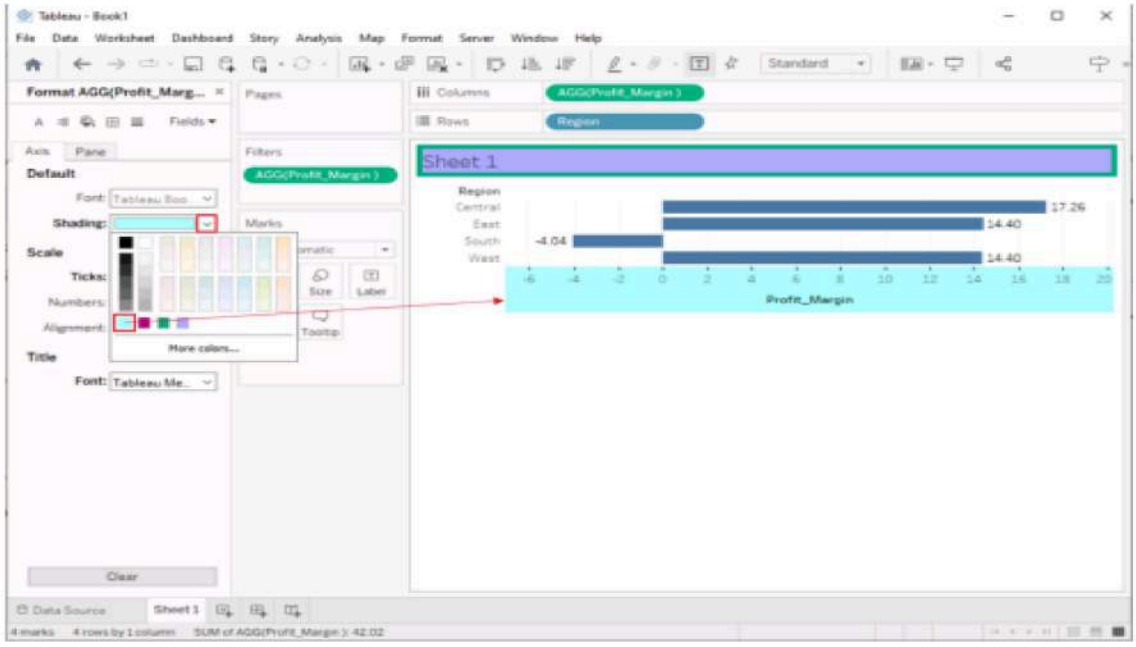
Editing and Formatting Axes:

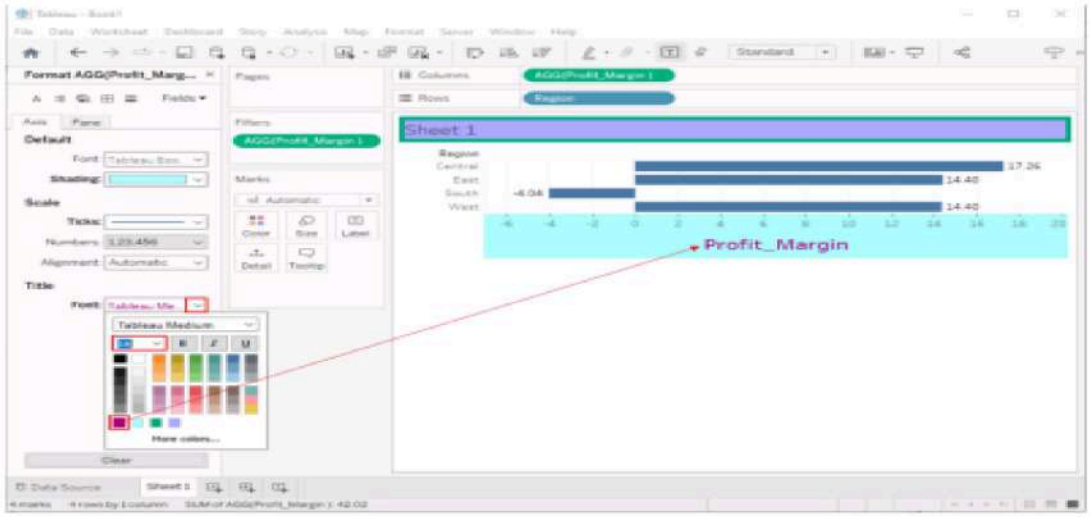




1. Edit Axis Title:

- Click on the axis title you want to edit.
- You can now modify the title text, font, size, color, and alignment using the Format pane or the toolbar at the top.



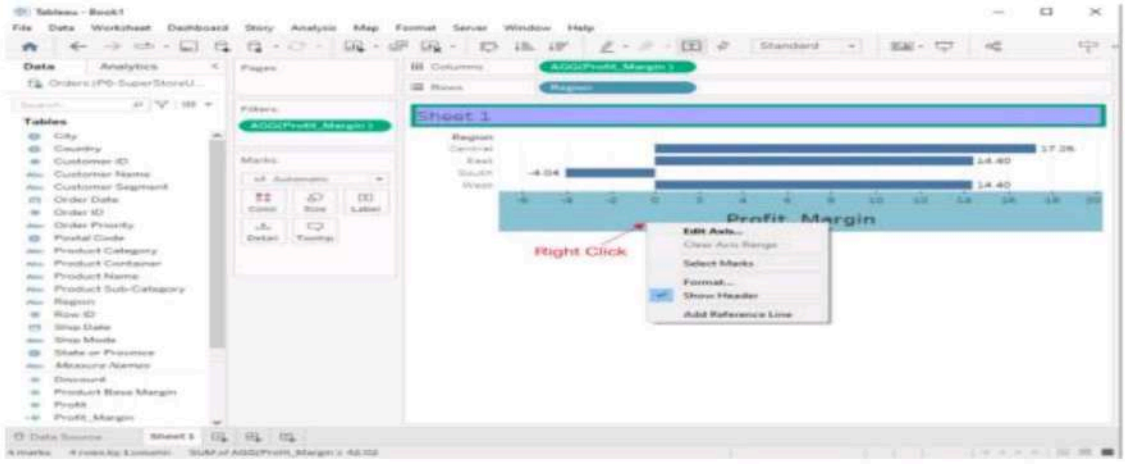


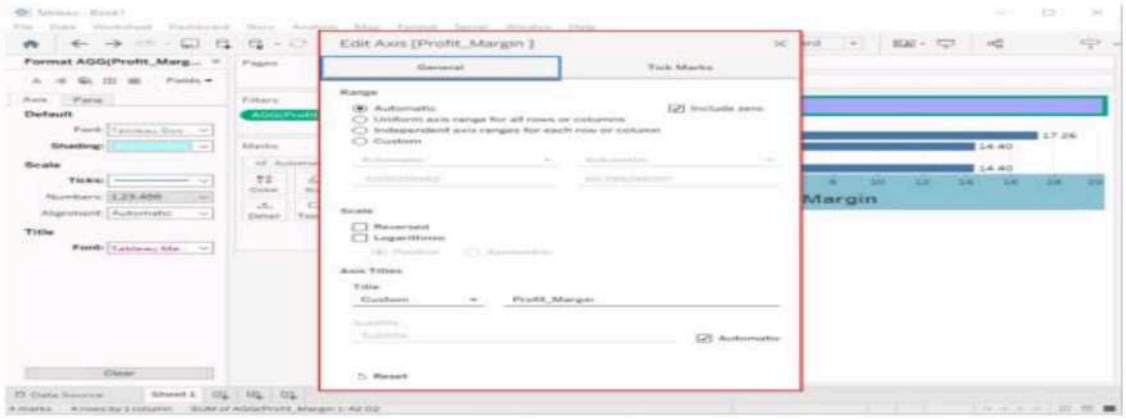
2. Edit Axis Labels:

- Right-click on an axis and select "Edit Axis."
- In the Edit Axis dialog box, you can change the formatting of labels, tick marks, and other axis-related properties.

3. Scale and Range:

- To change the scale or range of an axis, right-click on it and select "Edit Axis."
- In the dialog box, adjust the Minimum and Maximum values, scale, or range according to your needs.




Edit Axis [Profit_Margin]
General
Tick Marks
Range
Automatic
Include zero
Scale
Axis Titles
Profit_Margin
Reset
17.26
14.40
14.40
Margin


Edit Axis [Profit_Margin]
General
Tick Marks
Major Tick Marks
Automatic
Fixed
None
Tick origin
0
Tick interval
10
Minor Tick Marks
Automatic
Fixed
None
Reset

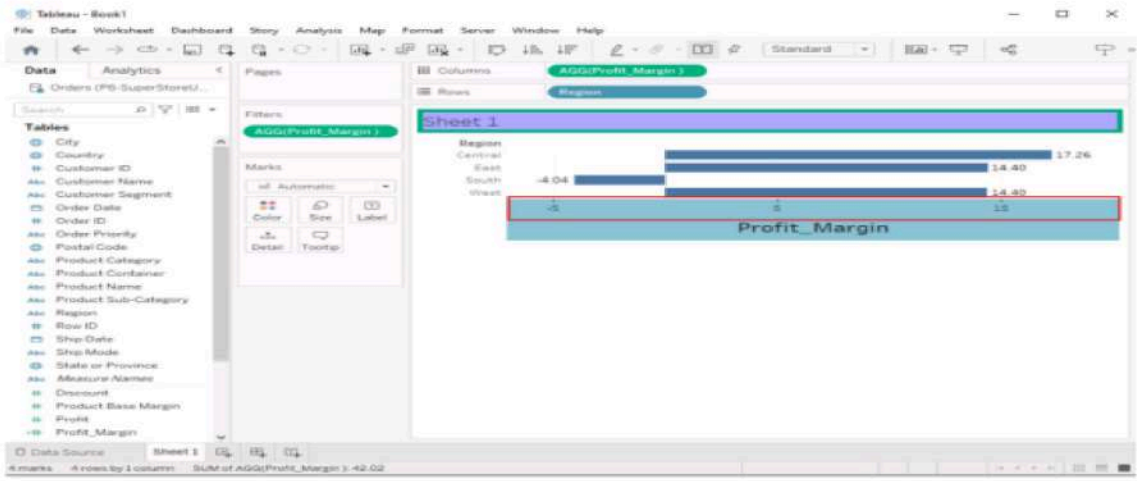
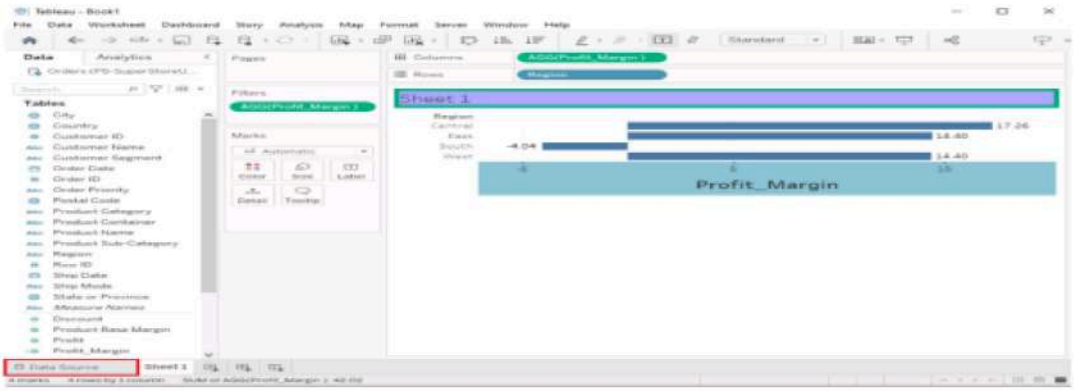


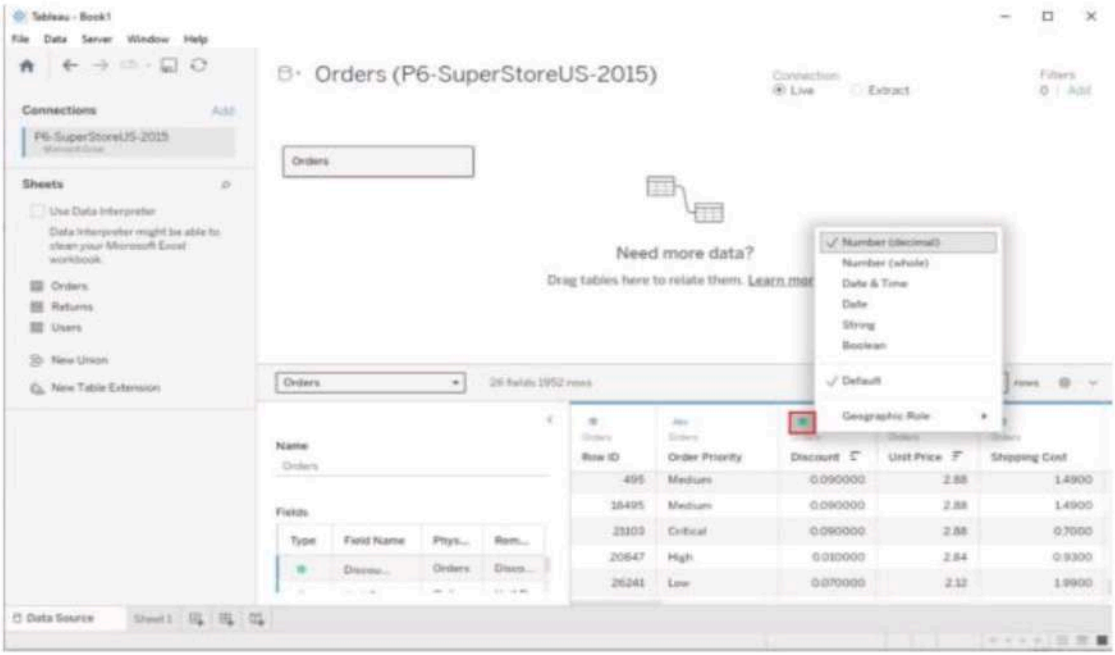
Tableau - Book1
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Data
Analytics
Tables
City
Country
Customer ID
Customer Name
Customer Segment
Order Date
Order ID
Order Priority
Postal Code
Product Category
Product Container
Product Name
Product Sub-Category
Region
Row ID
Ship Date
Ship Mode
State or Province
Measure Names
Discount
Product Base Margin
Profit
Profit_Margin
Pages
Filters
Marks
Color
Size
Label
Detail
Tooltip
Columns
AGG(Profit_Margin)
Rows
Region
Sheet 1
Region
Central
East
South
West
17.26
14.40
-4.04
14.40
Profit_Margin

Manipulating Data in Tableau data



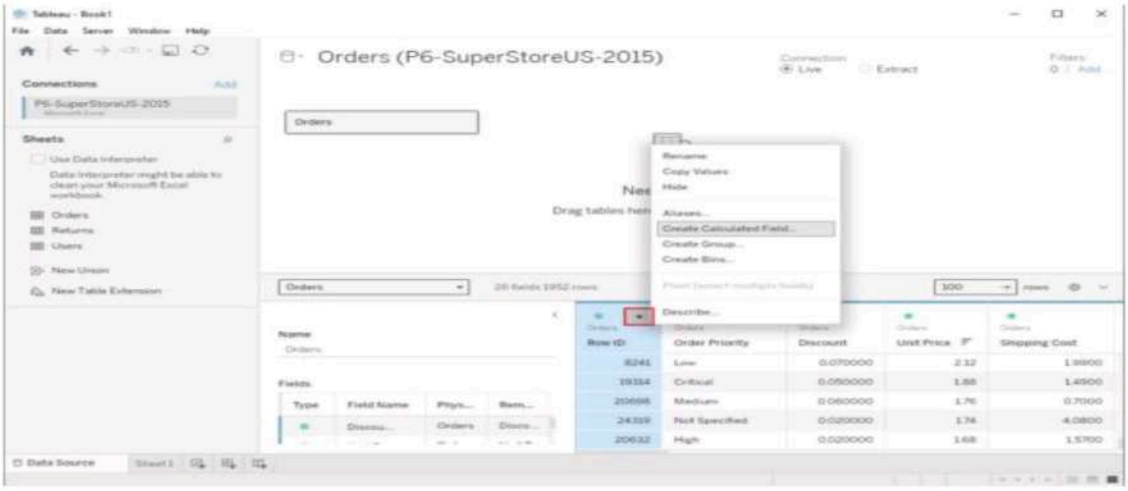
Change Data Type

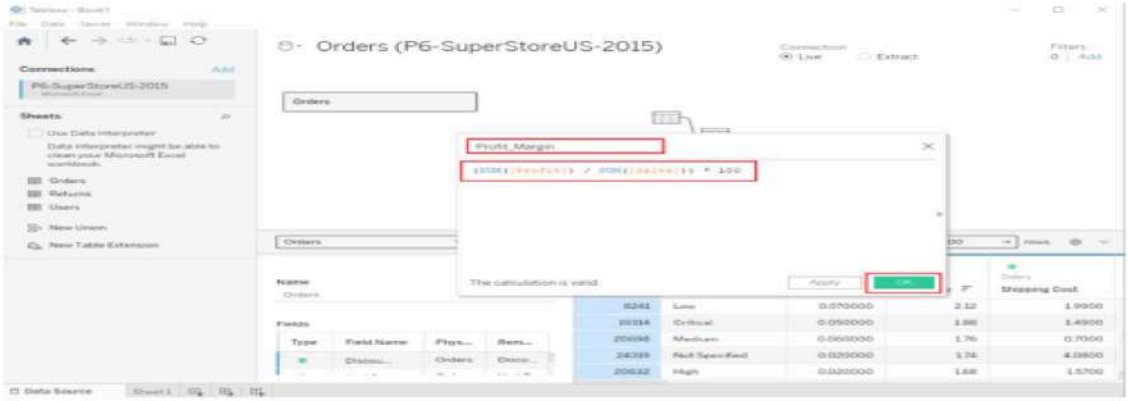
If Tableau has inferred a wrong data type for a column, the data type can be changed by clicking on the data type symbol in the column header

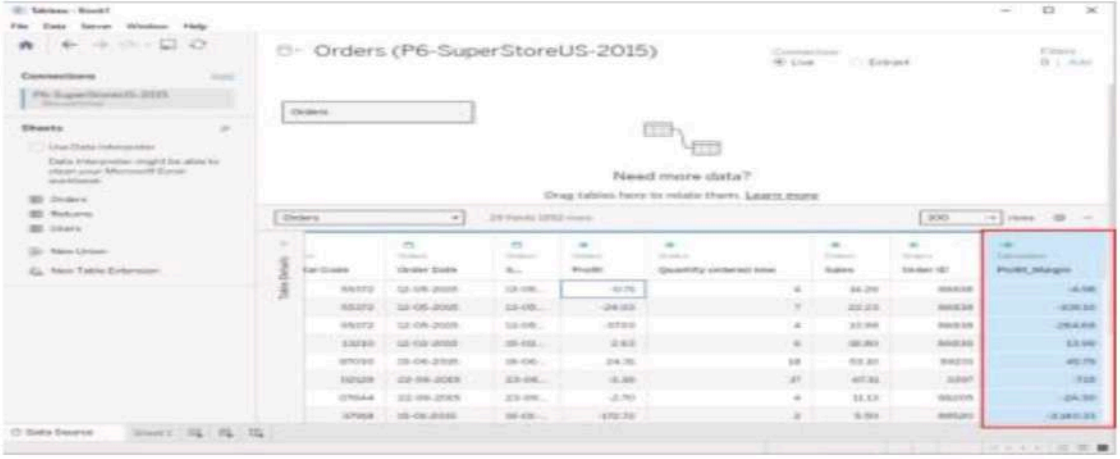


New Column(Calculated Fields)

Calculated fields can be used if you need to create customized logic for manipulating certain data types or data values. There are a large-range of functions available in Tableau that can used individually or collectively for data manipulation

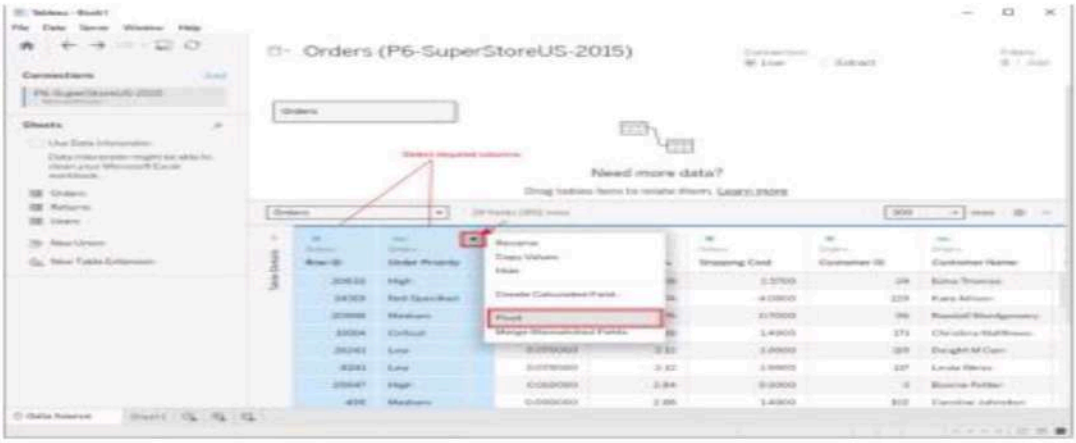

Orders (P6-SuperStoreUS-2015)
Create Calculated Field...
Row ID
Order Priority
Discount
Unit Price
Shipping Cost


Orders (P6-SuperStoreUS-2015)
Profit_Margin
The calculation is valid.
Apply
OK


Orders (P6-SuperStoreUS-2015)
Need more data?

Pivoting Tableau data

Data pivoting enables you to rearrange the columns and rows in a report so you can view data from different perspectives



EXPERIMENT-11

AIM: Structuring the data, sorting and filtering tableau data, pivoting tableau data.

PROCEDURE:

Step1: To create a filter, drag a field directly from the Data pane to the Filters shelf.

Step2: To filter entire rows or columns of data from your view, select the header in the view on the tooltip that appears, select to Exclude or Keep Only the selected data.

Step3: Table calculation filter to apply to the totals, you can select Apply to totals in the

drop-down menu for that filter (on the Filters shelf). This option lets you decide

when a table calculation filter should be applied to totals.

Step4: To select a filter card mode, in the view, click the drop-down menu on the filter card and then select a mode from the list.

Step5: To sort specific field in viz Right-click (Windows) or control-click (Mac) the field you want to sort, and select Sort. Select a Sort By option, and configure its behavior

Step6: To remove the sort on a specific field, right click to open the menu and select Clear Sort.

Structuring Data

Your organization is having a challenge of dealing with large volume of data. There is no proper flow to understand the data in such volume. we have the the challenge of structuring the data to make it in a proper order for the presentation.

The following are different concepts which can help use to understand the data in such a large volume

a)sorting, b)sets, c)groups, d)bins, e)hierarchies

Create Hierarchy

Hierarchies can be building in Tableau to visualize the data in granular level. Tableau hierarchies can be created by following the given steps.

Step 1) Go to a worksheet.

Select a dimension to create a hierarchy. Right-click on the dimension.

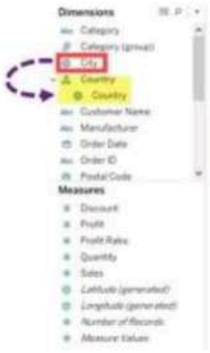
Select 'Hierarchy' option.

Click on 'Create hierarchy' option

Step 2) It opens the 'Create Hierarchy' Window.

1. Enter a name for hierarchy.
2. Click on OK

It creates a Hierarchy as shown in the image.

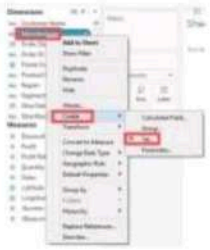


Build Sets

Sets create a set of members out of the field present in a data set. It acts as a separated field or dimension. The procedure to build sets is given as follows.

Step 1) Go to a Worksheet.

1. Right-click on a dimension.
2. Select 'Create' option.
3. Click on 'Set' option.



Step 2) It opens 'Create Set' Window.

1. Name the set to be created.
2. Select the members needs to be added in the set.
3. Click on OK.

Filter Data from Data Sources

Data source filters can be useful for restricting the data users can see when you publish a workbook or data source. When you publish a data source to Tableau Server, the data source and any associated files or extracts are transported in entirety to the Server.

Create a data source filter

The primary way to create a data source filter is from the data source page

To create a data source filter

1. On the data source page, click Add in the Filters section in the upper-right corner of the page

   To create a data source filter on a worksheet, right-click (control-click on a Mac) the data source and choose Edit Data Source Filters.

   Whether you start from the Data Source page or from a worksheet, you see an Edit Data Source Filter dialog box, listing any existing data source filters.

Click Add to open an Add Filter dialog box listing all fields in the data source.

Click to select a field to filter; then specify how the field should be filtered, just as you would for a field on the Filters shelf.

To add an additional data source filter, repeat this procedure

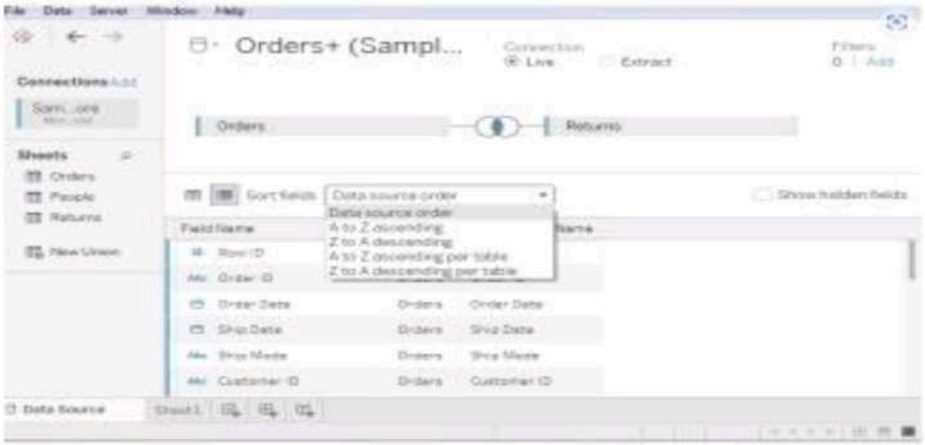
Tableau Data Sorting

In the data source, data can be stored based on the user requirement. It can be sorted using data source order such as `A to Z ascending`, `Z to A descending`, `A to Z ascending per table` and `Z to A descending per table`.

Once the data is connected with Tableau, data sorting is done using the `Sort Fields` option. The `Sort Fields` option is present in the `Data Source` tab.

There are two ways to sort the data in Tableau:

1. Manual sorting: Manual sorting is a sort that rearranges the order of dimension fields by dragging them next to each other in ad hoc fashion.
2. Computed sorting: The computed sorting is a sort which is directly applied on the axis using the sort dialog button.



EXPERIMENT-12

AIM: **Advanced visualization tools: using filters, using the detail panel, using the size panels, customizing filters, using and customizing tooltips, formatting data with colors**.

PROCEDURE:

Step1: To assign a color to marks in the view, do one of the following:

a. On the Marks card, click Color, and then select a color from the menu.
b. From the Data pane, drag a field to Color on the Marks card.

Step2: To change the color palette or customize how color is applied to your marks:

a. On the Marks card, click **Color > Edit Colors.**

Step3: To change the size of marks in the view, do one of the following:

a. On the Marks card, click Size, and then move the slider to the left or right.
b. From the **Data** pane, drag a field to Size on the Marks card.

Step4: To edit the size of marks, or change how size is being applied to marks in the view:

a. On the Size legend card (which appears when you add a field to Size on the

Marks card), click the drop-down arrow in the right-hand corner and select Edit Sizes.

b. In the Edit Sizes dialog box that appears, make your changes and then click OK.

Step5: To add a tooltip

a. Drag a field to Tooltip on the Marks card.
b. Click Tooltip on the Marks card to open the Edit Tooltip dialog box, where you can add text, rearrange the tooltip contents, and insert more fields.

Assign colors to marks

To assign a color to marks in the view, do one of the following:

- On the Marks card, click **Color**, and then select a color from the menu.

  This updates all marks in the view to the color you choose. All marks have a default color, even when there are no fields on **Color** on the **Marks** card. For most marks, blue is the default color; for text, black is the default color.

- From the **Data** pane, drag a field to **Color** on the Marks card.

  Tableau applies different colors to marks based on the field's values and members. For example, if you drop a discrete field (a blue field), such as Category, on Color, the marks in the view are broken out by category, and each category is assigned a color.

Add tooltips to marks

Tooltips are details that appear when you hover over one or more marks in the view. Tooltips are also convenient for quickly filtering or removing a selection, or viewing underlying data. You can edit a tooltip to include both static and dynamic text. You can also modify which fields are included in a tooltip and whether you want to be able to use those fields to select marks in the view..

Add a tooltip

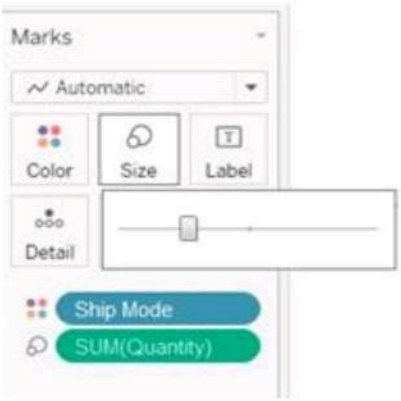
Drag a field to Tooltip on the Marks card.

Click Tooltip on the Marks card to open the Edit Tooltip dialog box, where you can add text, rearrange the tooltip contents, and insert more fields.

Change the size of marks

To change the size of marks in the view, do one of the following:

On the Marks card, click Size, and then move the slider to the left or right.



Separate marks in the view by dimension members

To separate marks in the view (or add more granularity):

- From the **Data** pane, drag a dimension to Detail on the Marks card.

When you drop a dimension on Detail on the Marks card, the marks in a data view are separated according to the members of that dimension. Unlike dropping a dimension on the Rows or Columns shelf, dropping it on Detail on the Marks card is a way to show more data without changing the table structure.

Filter Data from Your Views

Tableau performs actions on your view in a very specific order; this is called the Order of Operations. Filters are executed in the following order:

Extract filters

Data source filters

Context filters

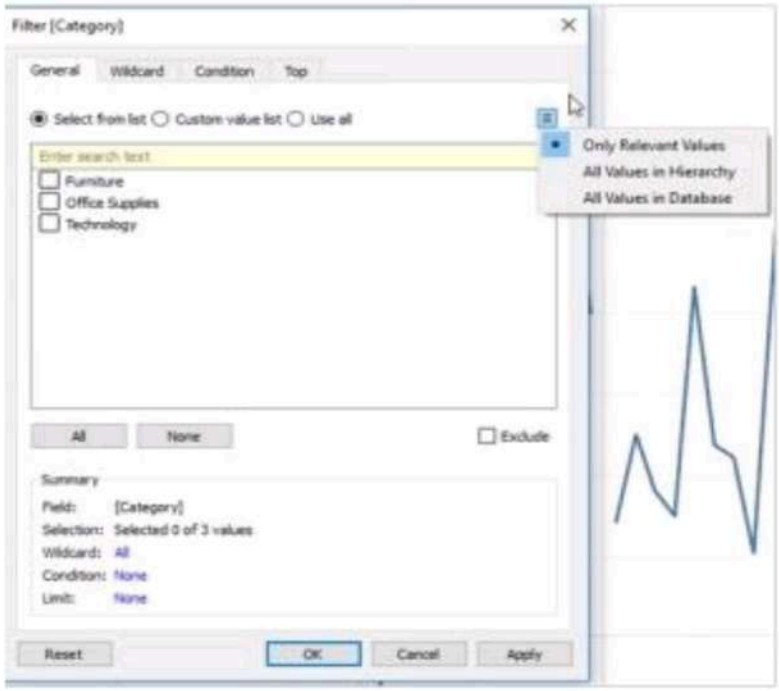
Filters on dimensions (whether on the Filters shelf or in filter cards in the view)

Filters on measures (whether on the Filters shelf or in filter cards in the view)

Filter categorical data (dimensions)

Dimensions contain discrete categorical data, so filtering this type of field generally involves selecting the values to include or exclude.

When you drag a dimension from the Data pane to the Filters shelf in Tableau Desktop, the following Filter dialog box appears:



- **General**: Use the General tab to select the values you want to include or exclude.
- **Wildcard** (Tableau Desktop only): Use the Wildcard tab to define a pattern to filter on. For example, when filtering on email addresses you might want to only include emails from a specific domain. You can define a wildcard filter that ends with "@gmail.com" to only include Google email addresses.
- **Condition**: Use the Condition tab in the Filter dialog box to define rules to filter by. For example, in a view showing the average Unit Price for a collection of products, you may want to only show the Products that have an average unit price that is greater than or equal to $25. You can use the built-in controls to write a condition or you can write a custom formula.
- **Top**: Use the Top tab in the Filter dialog box to define a formula that computes the data that will be included in the view. For example, in a view that shows the average Time to Ship for a collection of products, you can decide to only show the top 15 (or bottom) products by Sales. Rather than having to define a specific range for Sales (e.g., greater than $100,000), you can define a limit (top 15) that is relative to the other members in the field (products).

EXPERIMENT-13

AIM: Creating dashboards and storytelling, design for different displays, adding interactivity in the dashboard, distributing, publishing data visualization.

PROCEDURE:

Step1: You create a dashboard in much the same way you create a new worksheet.

a. At the bottom of the workbook, click the New Dashboard icon:

b. From the Sheets list at left, drag views to your dashboard at right
c. To replace a sheet, select it in the dashboard at right. In the Sheets list at left, hover over the replacement sheet, and click the Swap Sheets button

Step2: You can add interactivity to dashboards to enhance users' data insights. Try these techniques:

a. In the upper corner of sheet, enable the Use as Filter option to use selected marks in
the sheet as filters for other sheets in the dashboard.
b. When authoring in Tableau Desktop, add actions to use multiple sheets as filters,
navigate from one sheet to another, display web pages

Step3: To publish to Tableau Server or Tableau Cloud, your server or site administrator must grant you the following capabilities:

a. A site role of Creator (formerly Publisher) on the site you're publishing to.
b. View and Save capabilities set to Allowed on the project into which you publish.

Step4: In Tableau Desktop, open the workbook you want to publish.

a. Select **Server > Publish Workbook**.
b. If the Publish Workbook option does not appear on the Server menu, make sure a worksheet or dashboard tab is active (not the Data Source tab).
c. If necessary, **sign in** to a server. For Tableau Cloud, enter https://online.tableau.com.
d. In the Publish Workbook dialog box, select the project, enter a name for the workbook, and add search tags.

Using the Sample-superstore, plan to create a dashboard showing the sales and profits for different segments and Sub-Category of products across all the states.

To achieve this objective, following are the steps.

Step 1 − Create a blank worksheet by using the add worksheet icon located at the bottom of the workbook.

Drag the dimension **Segment** to the columns shelf and the dimension **Sub-Category** to the Rows Shelf.

Drag and drop the measure **Sales** to the **Color shelf** and the measure **Profit to the Size shelf**. This worksheet is referred as the Master worksheet. Right-click and rename this worksheet as **Sales_Profits**.

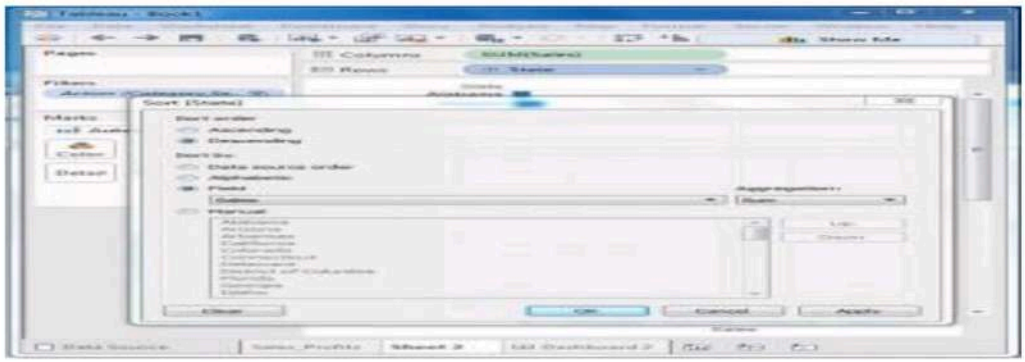
The following chart appears:

Step 2 − Create another sheet to hold the details of the Sales across the States. For this, drag the dimension

State to the Rows shelf and the measure Sales to the Columns shelf as shown in the following screenshot.

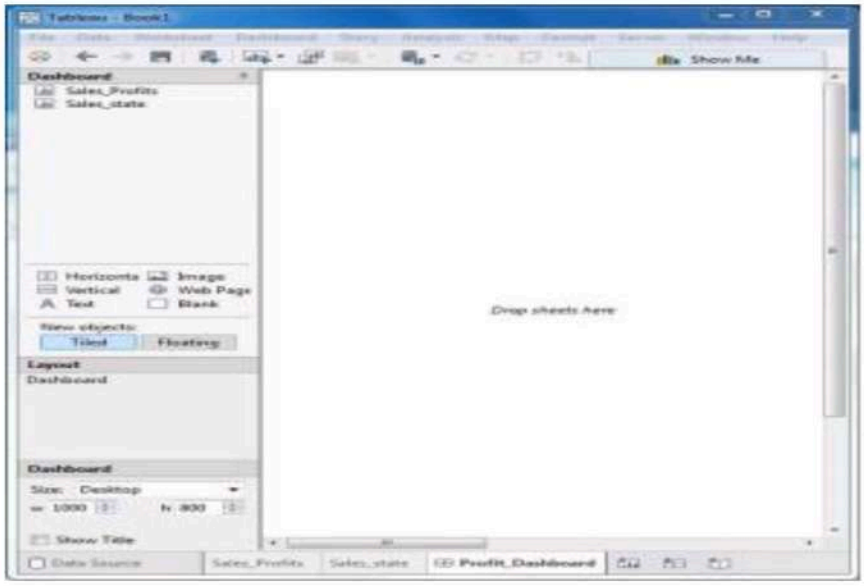
Next, apply a filter to the State field to arrange the Sales in a descending order.

Right-click and rename this worksheet as **Sales_state**.

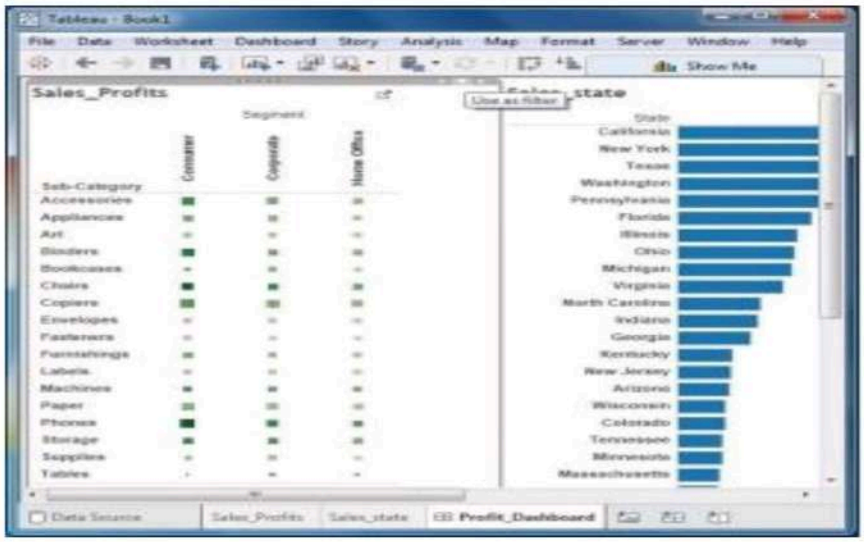


Step 3 − Next, create a blank dashboard by clicking the Create New Dashboard link at the bottom of the workbook.

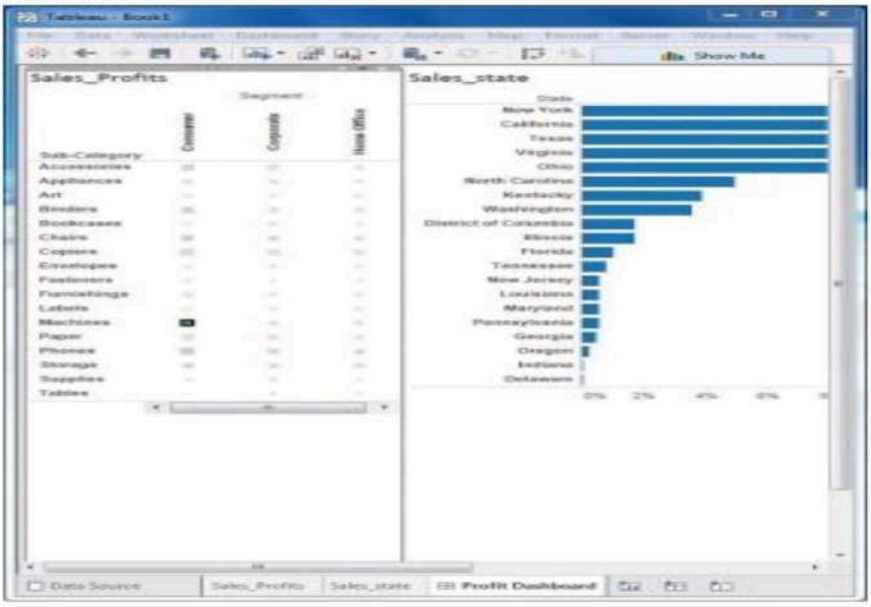
Right-click and rename the dashboard as Profit_Dashboard.



Step 4 − Drag the two worksheets to the dashboard. Near the top border line of Sales Profit worksheet, you can see three small icons. Click the middle one, which shows the prompt Use as Filter on hovering the mouse over it.



Step 5 − Now in the dashboard, click the box representing Sub-Category named Machines and segment named Consumer.
You can notice that only the states where the sales happened for this amount of profit are filtered out in the right pane named **Sales_state**. This illustrates how the sheets are linked in a dashboard.



Share your findings

Before you continue, **select an option below:**

- If you or your company does not use Tableau Server, or if you want to learn about a free, alternative sharing option, jump to **Use Tableau Public**.
- If you or your company uses Tableau Server, and you are familiar with what permissions are assigned to you, jump to **Use Tableau Server**.

When you publish to Tableau Public, as the name suggests, these views are publicly accessible.

This means that you share your views as well as your underlying data with anyone with access to

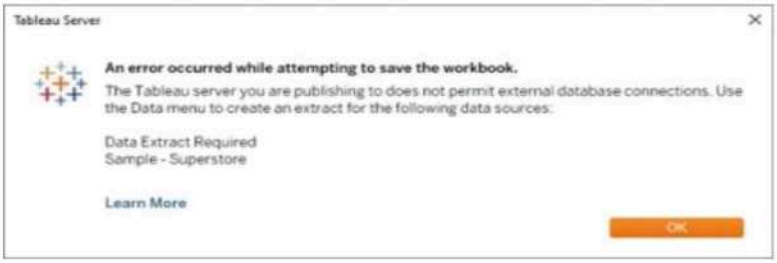
the internet.

Select **Server** > **Tableau Public** > **Save to Tableau Public**.

Enter your Tableau Public credentials in the dialog box.

If you don't have a Tableau Public profile, click **Create one now for free** and follow the prompts.

If you see this dialog box, open the Data Source page. Then in the top-right corner, change the **Connection** type from **Live** to **Extract**.



For the second (and last) time, select **Server** > **Tableau Public** > **Save to Tableau Public**.

When your browser opens, review your embedded story. It will look like this:

Click **Edit Details** to update the title of your viz, add a description, and more.

Click **Save**.

Your story is now live on the web.

To share with colleagues, click **Share** at the bottom of your viz.

EXPERIMENT-14

AIM: Creating custom charts, cyclical data and circular area charts, dual axis charts.

PROCEDURE:

Step1: To create dual axis charts use the following steps:

a. Make a graph for one of the measures
b. Drag the second measure onto the opposite axis
c. Create a dual-axis combination chart by changing one of the mark types

Step2: To create the Circular area chart:

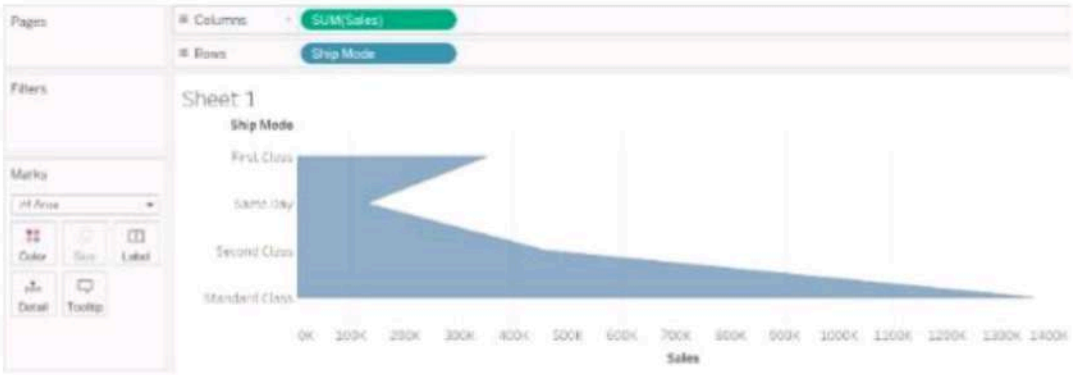
a. First drag and drop required fields from data pane to both rows and columns
b. On the Marks card, click the Mark Type drop-down and select circular area chart.

Step3: To create custom funnel chart

a. **Add Measures to the Rows Section**
b. **Select Dimensions in the Marks Section**
c. **Convert Standard View to Entire View**
d. **Add Labels in the Marks Section**

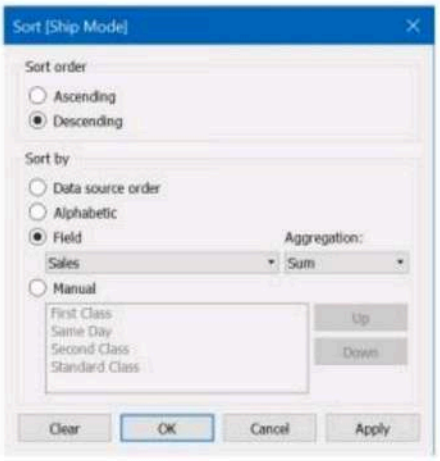
Funnel chart:

Step 1:
Connect the Sample superstore data's Order tab to your tableau.
Step 2:
Drag the "Ship Mode" dimension to the row shelf and "Sales" measure to the column shelf.
Step 3:
Select the chart type as area. This you can select from the Marks as shown below.



Step 4:

Now we need to sort this graph. For this right-click on the ship mode dimension available in the row shelf and select "sort". And select descending order by the sales field as shown below:



Step 5:

Create a calculated field for the negative sales and name it anything you like (CF_Negative_Sales, in my case). This will be just –[sales].

Step 6:
Drag this negative sale calculated field in the column shelf before the sum(sales). This will look like below. Make some adjustments to height, width, and color as needed.

Step 7:

Now show/hide/change axis and header as required. If you want to show the sales value also, you can enable it from the label. You should also hide the gridlines wherever not required by formatting the worksheet. Once done, the final funnel chart in the tableau will look like below.

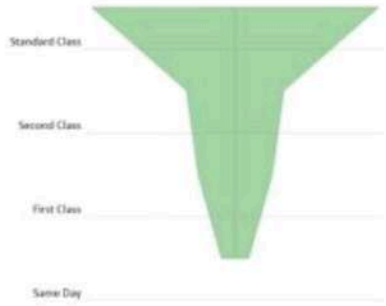


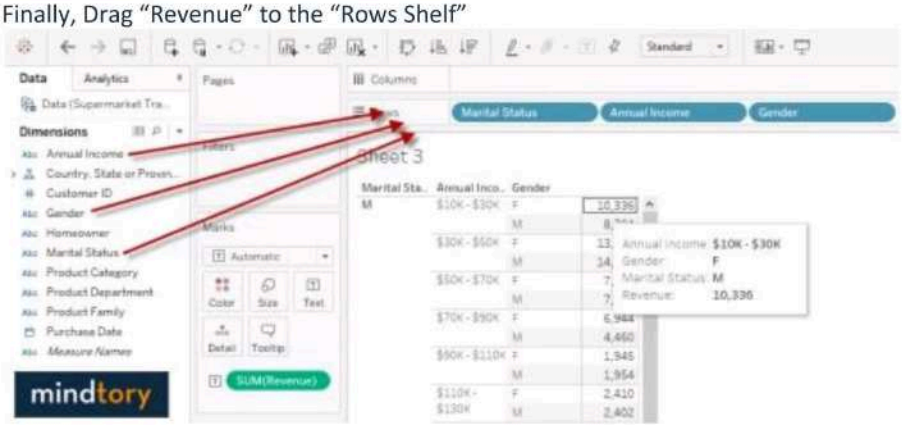
Tableau **Cycle Fields** can be used to cycle the current included dimensions in the current Tableau sheet. This is very useful when you want to see the effect of different dimension variations.

Tableau Cycle Fields will cycle / rotate dimensions on the current worksheet

1. Open Tableau
2. Then, Select "Excel" from the left "Connect" panel. (Connect –> To File –> Excel)
3. Finally, Navigate to "Supermarket" transaction Excel file and click on "Open"

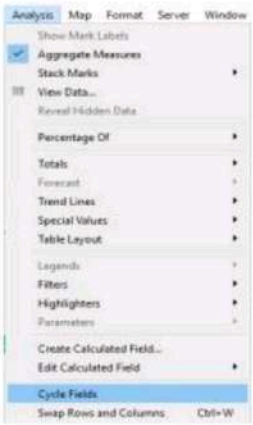
After loading dataset ,create a sample visualization. To do that

1. Click on "Sheet"
2. Then, Drag "Marital Status" to the "Rows Shelf"
3. Then, Drag "Gender" to the "Rows Shelf"
4. After that, Drag "Annual Income" to the "Rows Shelf"
5. Finally, Drag "Revenue" to the "Rows Shelf"



Note: Cycle Fields will be disabled till you have more than 1 dimension in your visualization. You need at least 2 dimensions inorder to use Cycle fields in Tableau.

The dimensions order is: Marital Status – Gender – Annual Income. Now, try to select "Cycle Fields" from "Analysis" menu. Note that the order changed to: Annual Income – Gender – Marital Status. Click Again on "Cycle Fields" and the order will change to: Annual Income – Marital status – Gender.



We will implement on stacked Bars

In the following example we can see revenue by Marital status, Gender and Annual Income. The gender used to color stacked bars in this case. Let's assume that we are not sure which dimension is the best one to be used as a color for stack bars. Here cycle Fields comes as a handy feature that allow cycling through available dimensions. In our case you may need to use cycle fields 3 times to see all combinations of the selected dimensions.

What is a Dual Axis Chart?

Dual-axis charts are a combination of two charts. It is used to visualize two or more different measures in two different chart types i.e. it depicts the relationship between two variables with different amplitude and scale.

To create a dual-axis chart we must have one date column and two measures.

Let's understand how to create a Dual Axis Chart in Tableau with an example

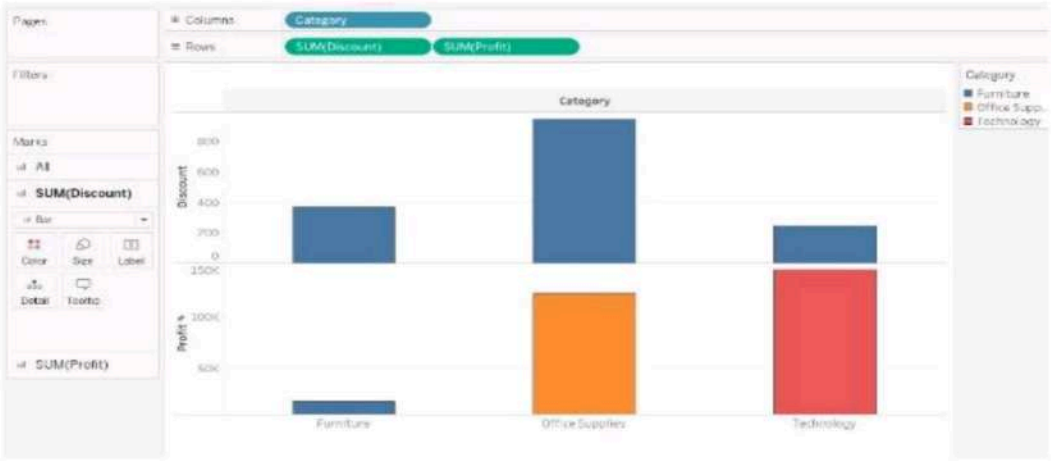
Steps to create a Dual Axis Chart

1.Connect a file (here, we will use Sample SuperStore data)

2.Drag and Drop **Category** into the column shelf

3.Drag and Drop **Profit and Discount** into the row shelf

4.Here, we have two charts present on multiple axes



- Right-click on Sum (Profit)
  - Select Dual-axis from the drop-down
  - We get the desired dual-axis chart