Experiment-1

Date: 31/08/24

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

```
import pandas as pd
import numpy as np
# generate data
data = {
    "id": [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 3],
    "data": ["2023-01-01", "2023-02-15", "2023-01-01", "2023-03-10",
             "2023-04-25", "2023-05-03", "2025-01-01", "2025-07-20",
             "2023-09-01", "15/08/2023", "2023-04-01"],
```

```
    "age" : [25, 30, 25, 22, 40, 33, 150, 29, 27, 35, 28],
    "income": [50000, 60000, 50000, 45000, 12000, 50000, 60000,
               58000, 72000, 50000, 40000],
    "score": [55, 90, 85, 75, 95, 86, 85, 88, 93, 85, 45]
}
#Convert to data frame
df = P.d.Data frame (data)
# Save to excel file
file.path = "Sample_ dataset.xlsx"
df.to excel (file_pat, index = False)
```

2)
```
#Load Sample data set
file_path = " Sample_dataset.XLSx"
dd = Pd.read_excel (file_path)
#display the original data set.
Print ("original data set:")
Print (df)
```

3)
```
# 1 formatting
# Convert data column to data time format
of ["data"] = Pd. to_date time (df ("data"], errors='coerce')
# handle non Convertible dates
# replace with not or a specific date
df ["data"]. fillna(Pd. TimeStamp ("2023-08-15"), inplace=True)
Print("formatted data Column")
Print (df("data"])
```

```
4) # 2. handLiners outliners
# detect outliners using z Score for age column
df["age_z Score"] = (df["age"]-df["age"]mean())/
                    df["age"]. Std()

outliners = df [np.abs (df ["age_z Score"]) >2)
Print(" In outliners. detected")
Print (outliners)

5) #remove outliners
df = df[np. abs (df ["age_z Score"]}}<=2}
# drop the particular helper column
df. drop (columns = ["age_z Score"], inplace = True)

6) # duplicate
duplicates = df [df. duplicated ()]
PrintC In duplicates detected")
Print (duplicates)

7) df ["norm_incom"]- (df ["income"] - df ["income"]. min()/
                       (df ["income"]. max ()-df ["income"]
                                              min ())

8) df ["Std_Score"] - df ["Score"] - df ["Score"].mean()
/df ["Score"]. Std ())
Print ("In Cleaned dataset:")
```

OUTPUT

```
print(df)
cleaned_file_path = "Cleaned_Sample_dataset.xlsx"
df.to_excel(cleaned_file_path, index=False)
```

OUTPUT:

	Id	data	age	Income	Score
0	1	2023-01-01	25	50000	85
1	2	2023-02-15	30	60000	90
2	3	2023-01-01	25	50000	85
3	4	2023-03-10	22	45000	75
4	5	2023-04-25	40	120000	95
5	6	2023-05-30	33	50000	80
6	7	2023-03-02	150	60000	85
7	8	2023-04-20	29	58000	88
8	9	2023-09-01	27	72000	91
9	10	15/08/2023	35	50000	85
10	3	2023-01-01	25	40000	45

Cleaned dataset

Id	data	age	Income	Score	name_income	Std_Score

Cleaned dataset

Id	data	age	Income	Score	name_income	Std_Score
1	2023-01-01	25	50000	85	0.1250	0.219270
2	2023-02-15	30	60000	98	0.2500	0.572932
3	2023-01-01	25	50000	85	0.1250	0.219270
4	2023-03-10	22	45000	75	0.0625	-0.498053
5	2023-04-25	40	120000	95	1.0000	0.926593
6	2023-05-30	33	50000	80	0.1250	-0.134391
7	—	—	—	—	—	—
8	2023-07-20	29	58000	88	0.2250	0.431467
9	2023-09-01	27	72000	91	0.4000	0.643664
10	2023-08-15	35	50000	85	0.1250	0.219270
3	2023-01-01	25	40000	45	0.0000	-2.610022

Lab Record

Experiment-2

Date: 10/08/24

AIM

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

PROCEDURE

Step-1: Set up PDFMiner using !pip install pdfminer.six.

Step-2: Use extract_text method found in pdfminer.high_level to extract text from the PDF file

Step-3: Tokenize the text file using NLTK.tokenize RegexpTokenizer

Step-4: Perform operations such as getting frequency distributions of the words, getting words more than some length etc.

Step-5: Use method such as collocations or collocation_list to get most frequently sequence of words occurring in the text

SOURCE CODE

```
from nltk.tokenize import RegexpTokenizer
from pdfminer.high_level import extract_text
#Extract the text from pdf file
text = extract_text("2010.00462.pdf")
```

```
# create an instance of tokenizer using NLTK Reg exp Tokenizer
                                        (explokenizer)
tokenizer = RegexpTokenizer("\w+")
# Tokenize the text read from Pdf
tokens = tokenizer.tokenize(text)
# find frequency Distribution
fd = FreqDist(tokens)
# Find words whose lengths is greater than 5 and
frequency greater than 20.
long_frequent_words = [words for words in tokens
                if len(words)]>5 and fd[words]>20]
long_frequent_words
```

Output :-

```
['PREPRINT',
 'different',
 'insurance',
 'language',
 'information',
 'analysis',
 'mining',
 'different',
 'models',
 'Sentence_---]
len(long_frequent_word)
```

210
freq_dist(long_frequent_words).plot().

=> Pdf stands for Portable Document format
=> Pdf miner is a python library to extract textual data from PDF documents. It provides a way to parse, analyze & work with content of pdf files, making it useful for various applications, including data extracting, text analysis and Document processing.

-> PyPDF2: It is a python library used for merging splitting, extracting text, rotating pages and formatting PDF files. It does all the manipulation.

-> PDF parser: It is used to extract all types of data from pdf documents.

Need for passing is to avoid errors and get accurate data.

-> Pf Query: It is used to extract images and text data from PDF documents. It has a high performance in terms of text extraction.

How to install Pd fimer!

Command: pip install pdf miner-six.

17/8

```
import pandas as pd.
# load the two Excel files.
file_path 1 = 'XLS_Chid_labour_databas_July-2024.xlsx'.
file_path 2 = 'XLS_Chid_marriage_database_May.2024.xls'.
df 1 = Pd.read_excel(file_path 1, Sheet_name='chid labour')
df 2 = Pd.read_excel(file_path 2, Sheet_name='child marriage')
# merge the two datasets on a common column.
# Replace 'Common_Column' with actual column name present in both datasets.
merged_df = Pd.merge(df1, df2, how='outer', on='countries and areas')
#check for duplicates and remove them
merged_df.drop_duplicates(inplace=True)
#check for the missing data.
missing_data_Summary = merged_df.isnull().Sum()
#Eliminate mismatches (removing rows with missing values)
merged_df.dropna(inplace=True)
# clean line breaks, spaces, and special characters
# Assuming text columns need this cleaning.
def clean_text(text):
    if isinstance(text, str):
    # Remove line breaks and extra spaces.
    text = ' '.join(text.split())
    # Remove special character (keeping only alphanumeric and spaces)
    text = ''.join(e for e in text if e.isalnum() or e.isspace())
    return text.
```

```
# Applying the cleaning function to all columns
merged_df = merged_df.applymap(clean_text)
merged_df
# Save the cleaned data to a excel file
Cleaned_file_path = 'cleaned_combined_data.xlsx'
merged_df.to_excel(Cleaned_file_path, index=False)
# Print Summary of missing data before cleaning
Print("Summary of missing data before cleaning")
print(missing_data_Summary)
# print Sommary of missing data before cleaning
print("Sommary of missing data before cleaning")
print(missing_data_Somma ry)
# print Summary of the cleaned data)
Print("Sommary of cleaned data:")
print(merged_df.info())
```

Output :-

```
Summary of missing data before cleaning.
Countries and areas.                          1
Chid labour (%) + ln (2015-2023)              2
unnamed : 2-x                               188
Unnamed : 3                                   2
Unnamed : 4-x                               188
unnamed : 5                                   2
unnamed : 6                                 189
unnamed : 7                                 112
unnamed married by 15                        16
unnamed : 2-y                               173
```

Married by 18	16
unnamed : 4-y	171
Reference year	78
observation footnote	195
Data Source	78
married by 18.1	16
unnamed : 9	185
Reference year.1	120
observation footnote.1	208
Data Source.1	120

dtype: int64

Summary of cleaned data :

<class 'pandas.core.frame.DataFrame'>

Index : 4 entries, 18 to 135.

Data columns (total 20 columns).

#	Column	Non-Null Count	DType
0	Countries and areas	4 non-NULL	Object.
1	Child labour (%) t (2015 - 2023)*	4 non-NULL	float 64
2	unnamed : 2-x	4 non-NULL	Objet
3	unnamed : 3	4 non-NULL	float 64
4	unnamed : 4-x	4 non-NULL	Object
5	unnamed : 5	4 non-NULL	float 64
6	unnamed : 6	4 non-NULL	Object
7	unnamed : 7	4 non-NULL	Object.

OUTPUT

8	Married by 15	4 non-NULL	float64
9	Unnamed : 2-g	4 non-NULL	object
10	Married by 18	4 non-NULL	~~o~~ bfloat64
11	Unnamed : 4-g	4 non-NULL	object
12	Reference year	4 non-NULL	object
13	Observation footnote	4 non-NULL	object
14	Data Source	4 non-NULL	object
15	Married by 18.1 -	4 non-NULL	float64
16	Unnamed 19	4 non-NULL	object
17	Reference year 2	4 non-NULL	object

19	Data Source 1.	4 non-NULL	Object.

Countries and child labour (%) Unnamed Unnamed Unnamed Unnamed Unnamed 7

Countries and areas	child labour (%) + in (2015-2023)	unnamed 2-x	unnamed 3	unnamed 4-x	unnamed 5	unnamed 6	unnamed 7
18. Belize	3.3	x	8.9	x	2.6	x	CSA 2018 unicef-d
19. Benin	19.9	y	20.4	y	19.4	y	ILO calculation
130 Nigeria	31.5	y	33.0	y	30.0	y	MICS 202122
135 Pakistan	11.4	y	12.5	y	10.1	y	MICS 2021 IFS20 AIS unicef-d ILO calculation

unnamed 2-y	married by 18	unnamed 4-y	Reference year	observation footnote	Date Source	Married by 18.1	unmarried 9
y	33.5	y	2015-16	Includes references to visiting unions	MICS 20516	22.2	y
y	27.5	y	2021-22	Two clusters could not be visited due to ince...	MICS 202122	4.6	y
y	30.3	y	2021	Due to prolonged insecured concins source part	MICS 2021	1.6	y
y	18.3	y	2017-18	Azad Jamut Kashmir AJK and Gilgit Baltistan	DHS 201718	4.7	y

Experiment-4

Date: 24/08/24

AIM

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

PROCEDURE

Step-1: Install the Matplotlib package

Step-2: Read the required data using read.csv method of pandas library

Step-3: Define the x-axis and corresponding y-axis values as lists.

Step-4: Plot them on canvas using .plot() function.

Step-5: Give a name to x-axis and y-axis using .xlabel() and .ylabel() functions.

Step-6: Give a title to your plot using .title() function.

Step-7: To view your plot, use .show() function.

SOURCE CODE

```
import pandas as pd
# attempt to load the datasets
corruption_df = pd.read_to_excel('corruption.Perception
                                  index.xlxs)
unicef_df = pd.read_excel('unicef_oct_2014_xlsx')
```

```
#display the first few rows of each dataset to
understand their structure.
corruption_df_head = corruption_df.head()
unicef_df_head = unicef_df.head()
corruption_df_head
unicef_df_head
unicef_df.head
#extract and clean the relevant columns from the
corruption data.
corruption.cleaned = corruption.df.iloc[2:,[1,7]]
Corruption.Cleaned=Colums=['country','corruption Score
corruption.cleaned = Corruption_cleaned.dropna()
                    reset_index(drop=True)
unicef.cleaned = unicef_cleaned.dropna().
                    df.iloc[4:,[0,1]]
unicef.cleaned.columns=['country','child labour percentage
unicef_cleaned= unicef_cleaned.dropna().
                    reset_index(drop=True)
#merge the datasets on the country columns
merge_df = pd.merge(corruption_cleaned,unicef.
                    cleaned,on='country')
```

```
# Convert the columns to appropriate types
merged_df['corruption score'] = pd.to_numeric(merged_df['corruption score'], errors='coerce')
merge_df['child Labour Percentage'] = pd.to_numeric(merged_df['child labour percentage'], errors='coerce')

# drop rows with any Non values.
merged_df = merged_df.dropna().reset_index(drop=True)

merged_df
```

o/p:-

	country	corruption score	Child labour Percentage	corruption score
0	Uruguay	6	7.9	6
1	Chile	9	6.6	9
2	Saint lucia	3	3.9	3
3	Botswana	7	9.0	7
4	Bhutan	4	2.9	4

```
import matplotlib.pyplot as plt
# create the scatte .plot
plt.figure(figsize=(10,6))
plt.scatter(merged.df['Corruption Score'],
merged.df['chid Labour Percentage'],color
# add labels and title:
plt.xlabel('Corroption perception Index(CPIoSc
plt.ylabel('chid labour Percentag')
plt.title('Comparision of Corruption Percep
Index & chid Labour percentage')
plt.grid(True)
# Show the graph
plt.Show()
```

o/p :-

```
:
:
89   Haiti'      5   24.4   5
90   Yemen       6   22.7   6
91   Iraq        4   4.7    4
92   Afganistan  3   10.3   3
93   Somalia     4   49.0   4
94rows x 4 columns
```

Experiment-5

Date: 10/08/24

AIM

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

PROCEDURE

Step-1: Use Requests (HTTP for Humans) Library for Web Scraping

Step-2: Scrape the robots.txt file

Step-3: print the data

SOURCE CODE

Web Crawler:- A web crawler or web spider, is a computer program that is used to search and automatically index website content and other information over the internet.

Crawling:- It is the discovery process in which search engine sendout a team of robots (spiders) to find new and updated content.

Indexing :- Website Ideation is the process by which a search engine add web content to its index This is done by "crawling" web pages for keywords, metadata and related signals that tell search engine it and where to rank content Indexed website should have navigable findable and clearly understood content strategy.

Robots txt file:

A robot txt file is a set of instruction used by website to tell search engine which pages shoud and should not be crawled.

Web Scrapping:-

→ web Scraping refers to collection and structuring the data from web source in a more convenient format. It involves no processing or review of data

web Scrapping can be used to build the data set that are to be used in data mining.

Data mining :-

Data mining refers to analyzing large data sets to reveal useful information and patterns. It doesn't require data processing extraction.

Data mining refers or is a the process of large data to uncover trend and valuable insights.

Request module:

Python request a library or making HTTP request it provide an easy-to-use interface that make working with HTTP very simple, which mean it simplifies the process of studying and receiving the data from websites.

CODE:

```
import requests as rq.
response = rq.get("https://en.wikipedia.org/robots.txt")
data = response.text
Print("robots text for https://en.wikipedia.org")
Print("=========")
Print(data)
```

O/P:

```
robots.text for https://en.wikipedia.org/
==========
```

OUTPUT

```
# robots.txt for https://en.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-
  always too fast. If you're
# irresponsible your access to the site may be
  blocked.
# observing spamming large amount of http://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
Disallow: /
# Wikipedia work bots:
User-agent: IssaBot
Disallow: /
```

Experiment-6

Date: 3/02/24

AIM

Foundations for building data visualizations, Creating first visualization.

PROCEDURE

Step-1: Study about basics of data visualization

Step-2: Study about prerequisites of data visualization

Step-3: Study about different visualization charts for data visualization

Step-4: read the required data into tableau using source connection in Tableau

Step-5: Study different visualization charts available in Tableau on the data

SOURCE CODE

What is Data?

Data refers to raw facts, statistics, or information collected or stored in a structural or unstructured from. Data can take various forms, such as text, numbers, images, videos, and more. It is a fundamental of all information and knowledge and is used in various fields for analysis, decision-making, and

understanding trends and patterns.

Data can be categorized into two main types:

*Structured Data: This type of data is organized into a specific format, such as tables or databases, and is easily searchable and analyzable. Example include spreadsheets, relational databases, and CSV files.

*Unstructured Data: Unstructured data lacks a specific format and can include text documents, social media posts, images, audio recordings, and more. Analyzing unstructured data often requires advanced techniques like natural language processing and image recognition.

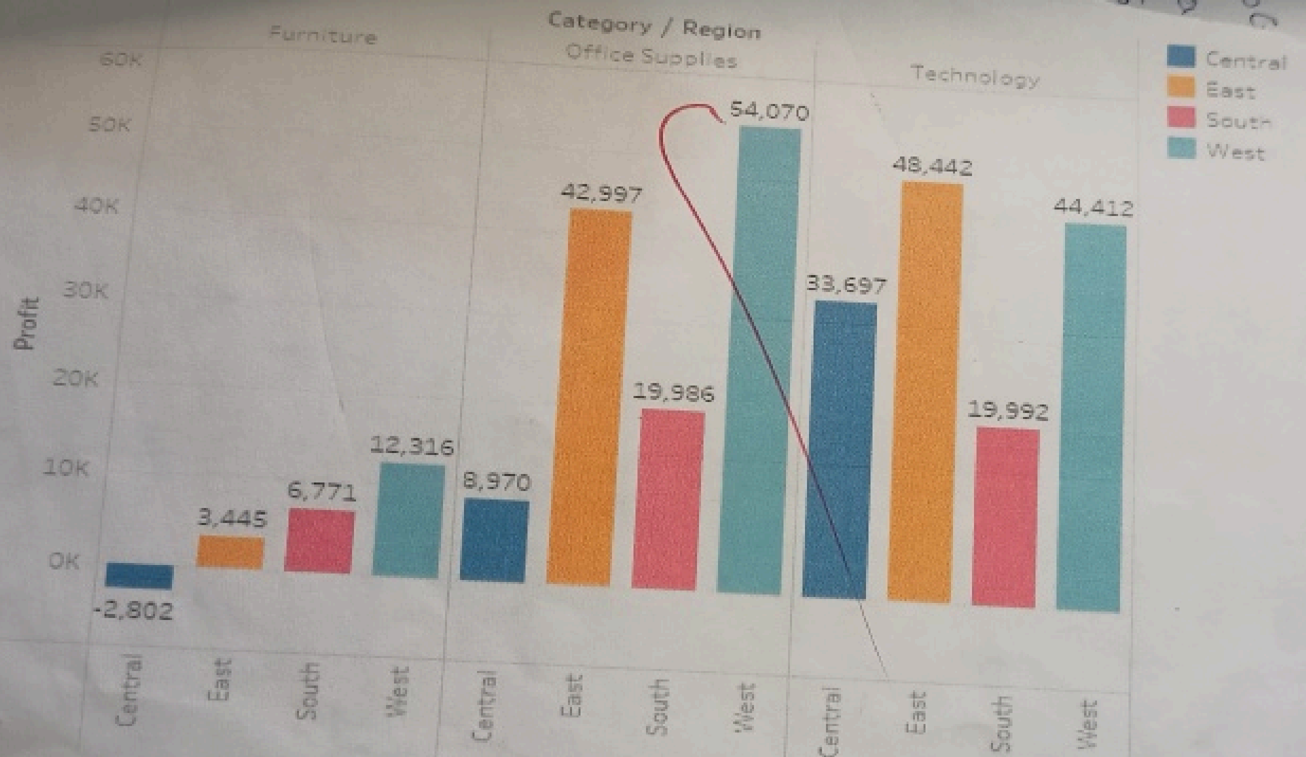
Where to find Data?

You can find data from various sources, depending on your specific needs:

*Open Data Portals: Many governments and organization provide free access to a wide range of data through open data portals. Example include Data.gov (united States) and data.gov.uk (United kingdom).

*Data Repositories: Academic institutions, research organizations, and data enthusiasts often share datasets on platforms like kaggle, GitHub, and the UCI Machine Learnings Repository.

Exp-6 FV



Sum of Profit for each Region broken down by Category. Color shows details about Region.

*APIs (Application Programming Interfaces): Some websites and services offer APIs that allow you to programmatically access and retrieve data. Examples include Twitter API, Google Maps API, and financial market APIs.

*Web Scrapping: You can extract data from websites using web scraping tools and libraries like Beautiful Soup and Scrapy. However, be mindful of the website's terms of use and legal restrictions.

*Surveys and Survey: You can conduct your own surveys or collect data through questionnaries and interviews.

*IOT Devices: Internet of Things (IOT) devices generate vast amounts of data that can be used for various purposes.

*Comanercial Data Provides: Some Companies Specialize in selling datasets for specific industrie, such as market research, finance, and healtcare.

Foundations for Building Data Visualizations:

Creating effective data visualizations requires a strong foundation in several key areas:

*Data Analysis: Before creating visualizations, you should thoroughly analyze your data to understand its structure, relationships, and any patterns or trends.

OUTPUT

Exploratory data analysis (EDA) techniques can help with this.
*Stastical knowledge: Understanding basic stastics is esseatial for making meaningful interpretations of data. Concepts like mean, median, standard deviation, and corseelation are commonly used in data visualization.
*Domain knowledge: Having knowledge of the speific domain or subject matter related to yourdata is crucial for creating contextually relevant visualizations. It helps you ask the right questions and provide valuable insights.
*Visualization Tools: Familarize yourself with data visualization tools and libraries such as matplotlib, Seaborn, ggplot2, D3.js, and Tableau. Each tool has its strengths and can be used for different types of visualizations.
*Design Principles: Study design priaciples, including color theory, typography and visual hierarchy tocreate visually appealing and effective visualizations. Avoid commoa pitfalls like misleading visualizations.
*Interactivity: Learn how to add interactive elements to your visualizations to engage users and allow them to explore the data. This can be achieved using tools like JavaScript. Python libraries or detected visualization Software.

Creating Your First visualization:

To create your first data visualization, follow these general steps:

* Select Your Data: Choose a dataset that aligns with your goals and interests. Ensure that the data is clean and well-structured.
* Define Your Objective: Clearly define what you want to communicate or explore with your visualization. Are you looking to show trends, comparisons, or distributions?
* Choose the Right Visualization Type: Select a visualization type that suits your data and objectives. Common types include bar charts, line charts, scatter plots, histograms, and pie charts.
* Prepare and Transform Data: Preprocess your data as needed. This may involve aggregating, filtering or transforming the data to fit the chosen visualization.
* Create the Visualization: Use a suitable tool or library to create your visualization. Customize it with labels, colors, and other design elements.
* Interactivity (Optional): If appropriate, add interactive features to your visualization to allow users to interact with the data.
* Test and Iterate: Review your visualization for accuracy and clarity. Seek feedback from others and make improvements as necessary.
* Publish or Share: Once you are satisfied with your visualization, publish it on a platform, embed it in a report, or share it with your intended audience.

Experiment-7

Date:

AIM

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

PROCEDURE

Step-1: Study about overview of Tableau

Step-2: To open the application, click the Tableau icon on your desktop (or in your Start m

Step-3: In the Connect panel at the left side of the Start page, click the Excel link under the a File" heading to the open file selection option.

Step-4: Using the file selection box, select the Excel worksheet that you want to open, and click the Open button to continue

Step-5: Drag the dimension and measure in row and column input field and it will automatic suggest a graph best fitted on data.

Step-6: You can change the graph by clicking on the show me button and select which graph you want.

SOURCE CODE

Procedure:-
Step 1:- Study our review of Tableau.
Step 2:- To open the application click the Tableau icon on your desktop (or in your Start menu).

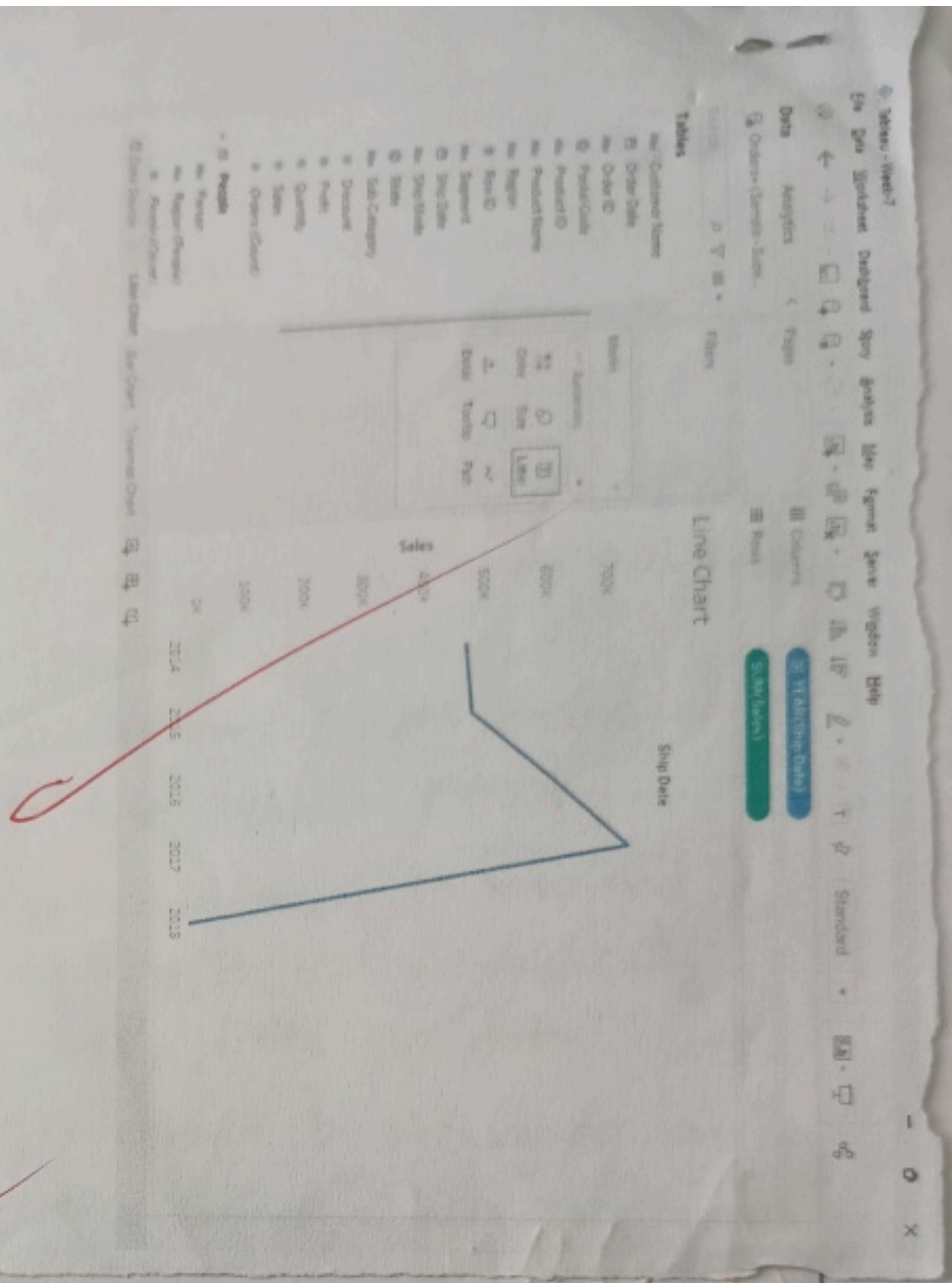
step3:- In the conne
start page. Cl
"To a file"
option.

step4:- Using the fi
work sheet
click then

step5:- Make relatio
common field

step6:- Drag the dim
column input
suggest a gr

step7:- You can ch
"show me" bo
you want.



a. Line Chart:

1. from the "data Source Pane", drag and drop the date field to the columns shelf and a numeric field (ex: sales, revenue) to the Rows Shelf.

b. Bar Chart:

1. drag and drop a categorical field (ex: product category, region) to the columns Shelf and a numeric field to the Rows shelf.

Step4:- In the connect panel at the left side of the start page. Click the excel line under the "To a file" heading to the open file selection option.

Step5:- Using the file selection box, select the Excel work sheet that you want to open and then click then open button to continue.

Step6:- Make relationship between sheets based on common fields in data source page.

Step6:- Drag the dimension and measure in row and column input field and it will automatically suggest a graph best fitted on data.

Step7:- You can change the graph by clicking on the "Show me" button and select which ever graph you want..

a. Line Char:

1. from the "data Source Pane", drag and drop the date field to the columns shelf and a numeric field (ex. sales, revenue) to the Rows shelf.

b. Bar Char:

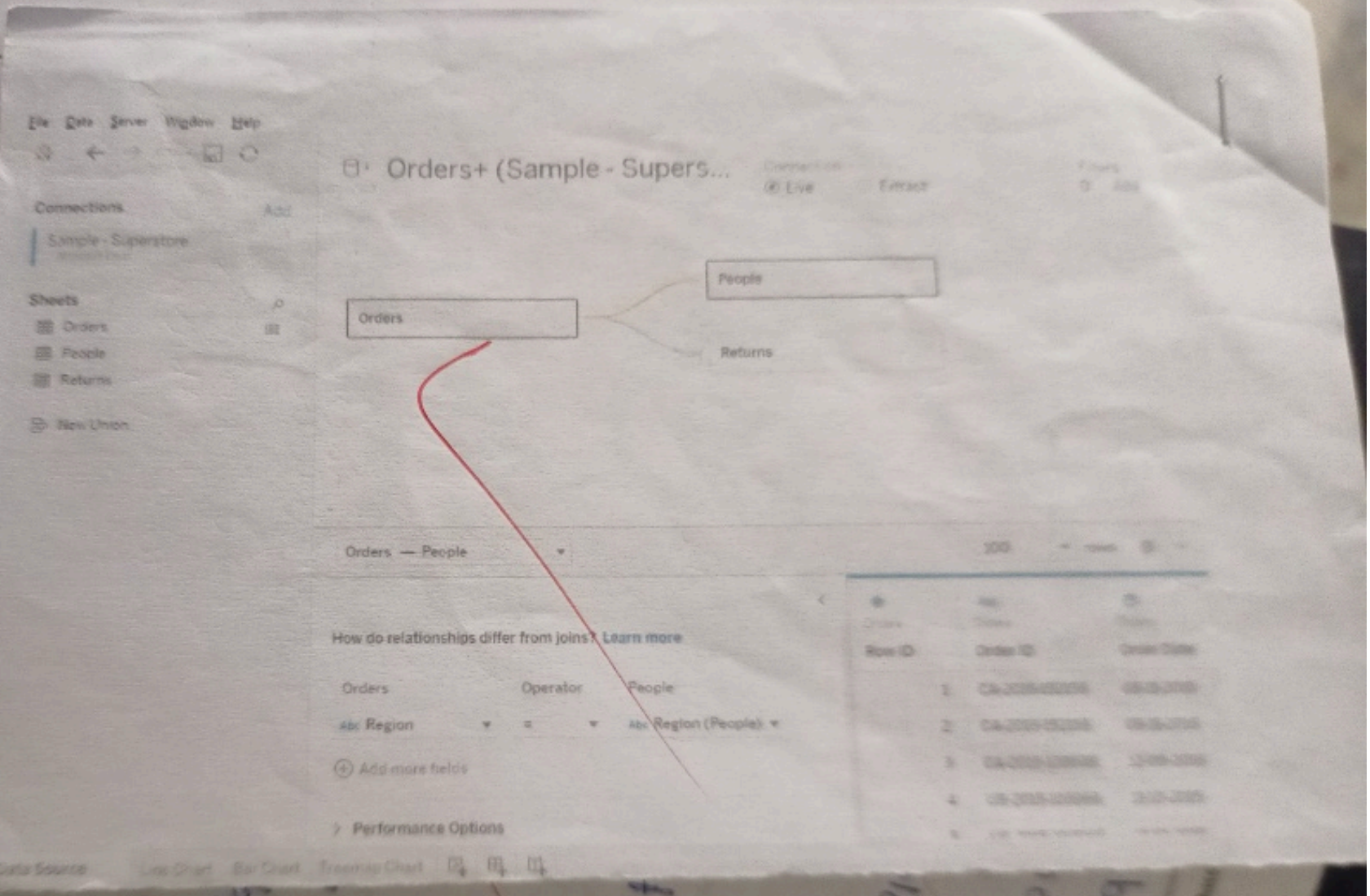
1. drag and drop a categorical field (ex: product category, region) to the columns shelf and a numeric field to the Rows shelf.

2. Then tableau will create a bar chart. You can adjust the orientation and formatting as needed. to display Labels on the bars click on Lables under marks and Select "Show mark Labels".

3. Tree Map:

1. drag and drop a categorical field to the colomns shelf.
2. drag and drop a numeric field to the size shelf under marks.
3. Tableau will create treemap visualization You can forthey customize it by adjusting colours and labels.

21/9.


File Data Server Window Help
Orders+ (Sample - Supers...
Live
Extract
Connections
Add
Sample - Superstore
Sheets
Orders
People
Returns
New Union
Orders
People
Returns
Orders — People
How do relationships differ from joins? Learn more
Orders
Operator
People
Abc Region
=
Abc Region (People)
Add more fields
Performance Options
Data Source

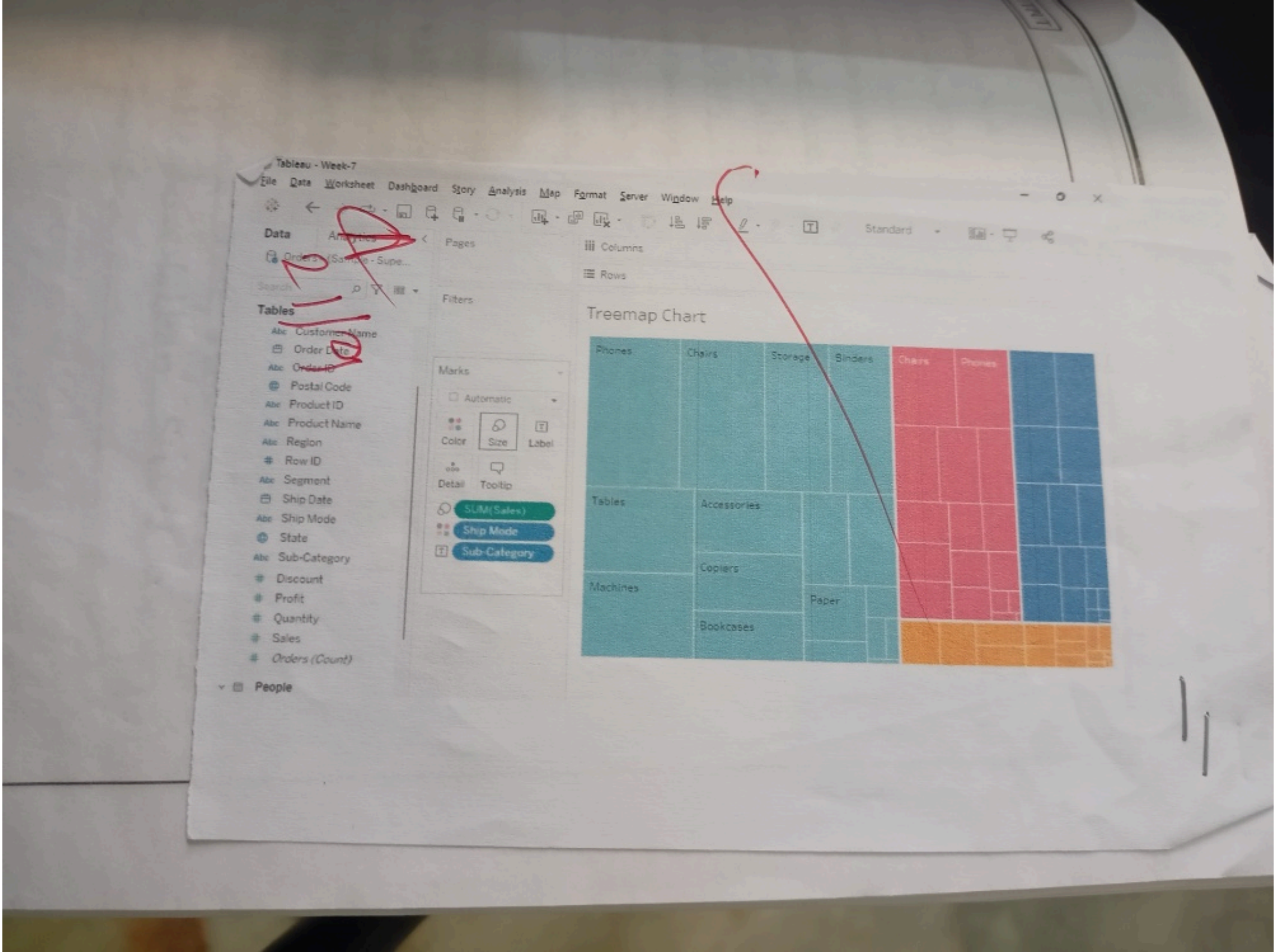
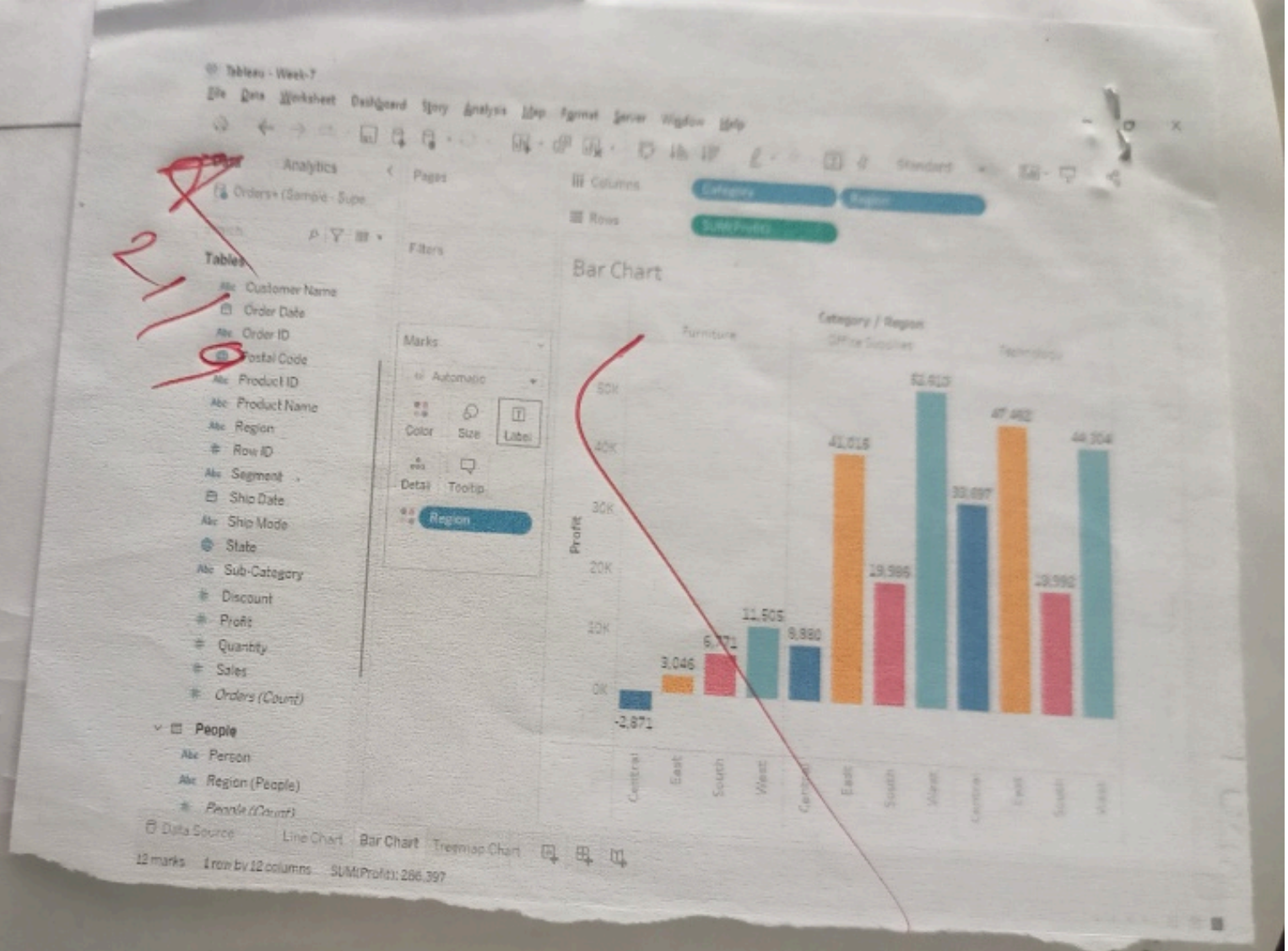


Tableau - Week-7
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Data
Pages
Columns
Rows
Filters
Tables
Customer Name
Order Date
Order ID
Postal Code
Product ID
Product Name
Region
Row ID
Segment
Ship Date
Ship Mode
State
Sub-Category
Discount
Profit
Quantity
Sales
Orders (Count)
People
Marks
Automatic
Color
Size
Label
Detail
Tooltip
SUM(Sales)
Ship Mode
Sub-Category
Treemap Chart
Phones
Chairs
Storage
Binders
Chairs
Phones
Tables
Accessories
Copiers
Machines
Paper
Bookcases



VIVA QUESTIONS

1. What is tableau?
Ans. Tableau people

2. What are the
Ans. These based Excel

3. What are
Ans. Basch Bobb

4. How ca
Ans. Shas

5. What
Ans. Yes

SOURCE CODE

Procedure:

Step1: Create the calculated field.

a. in a work sheet in Tableau, select Analysis > Create calculated field.

b. in the Calculation Editor that opens, give the Calculated a name.

Step 2:- Enter your Calculation.

a. in the calculation editor, enter a formula. For example, you can create a Calculation field to Calculate.

Profit margin as ((Sum [Prif]) / Sum (Sales)) * 100.

b. when you finished, click ok the new calculated field is added to the Data Pane.

Step 3: Study the overview of Sum, AVG and Aggregate function.

There are 3 main types of calculations you can create calculated field.

1. Basic calculations: (Row level Calculations) of at the visualization level of Detail (an aggregate calculation).
2. Lop Expressions: Level of Details.

Table calculations:

Table calculations allow you to transform value at the level of detail of the visualization only. SUM & AVG (average) function.

The SUM function in Tableau calculation the total sum of a numeric field.

You can use it to find the sum of values in a colomn or as part of a more complex calculated field using the sum function.

AVG- Function calculates the Average (mean) values of a numeric field like sum. You can use it dragging a numeric field into the avg shelf. or creating a calculated field with the Average function.

Aggregate function:- Tableau Provides a variety of aggregate function that allow you to perform calculations on group of data. Common aggregate function include SUM, AVG, Count, Min (minimum value), Max (maximum value). This function are particularly useful when you want granularity (example :- by category, region or time period).

Create one more calculated field.

% Returned Sales by overall Sales.

(Region wise return Sales) / (Region -Sales)

Region wise return Sales.

IIF (IFNULL([Returned) 'no')='Yes',
(Sales), NULL).

Region -Sales.

{fixed [Person], [Region]: Sum([Sales])}.

28/9

Experiment-8

Date:

rview of SUM, AVG and aggregate features, creating custom calcu

ated field

in Tableau, select Analysis > Create Calculated Field

ion Editor that opens, give the calculated field a name

on Editor, enter a formula.

o enter formula:

M([Sales])

click OK.

l field is added to the Data pane

of SUM, AVG and aggregate functions

SOURCE CODE

Procedure:

Step1: Create the calculated field.

a. in a work sheet in Tableau, select Analysis >Create calculated field.

b. in the calculation Editor that opens, give the calculated a name.

Step2:- Enter your calculation.

a. in the calculation editor, enter a formula. For example, you can create a calculation field to calculate.

Profit margin as ((Sum[Psif])/Sum(Csales)) *100.

b. when you finished, click ok the new calculated field is added to the Data Pane.

Step3: Study the overview of Sum, AVG and Aggregate function.

There are 3 main types of calculations you can create calculated field.

1. Basic calculations: Row level calculations) of at the visualization level of Detail can aggregate calculation).

2. Lop Expressions: Level of Details.

Experiment-9

Date: 19/10/24

AIM

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

PROCEDURE

I. Applying new data calculations to visualizations

Step-1: 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 Subcategory.

Step-2: 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.

SOURCE CODE

1. Drag and Drop Calculated Field.

To apply your newly calculated fields to a visualization simply drand and drop them into the appropriate shelves in your worksheet.

For example you can drag and 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. Filters with calculated fields.

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

Step-1: Display a worksheet or dashboard.

Step-2: From the Format menu, choose the part of the view that you want to format, such as Font, Borders, or Filters.

Step-3: To format interactive controls go to **Format > Worksheet > Interactive Controls**

Step-4: To control the background color of the worksheet, pane, and headers. to go **Format > Worksheet > Shading.**

Step-5: To format Filters and Sets, formatting by either going into Format > Filters and Sets

Step-6: To access highlighter formatting by either going into Format > Highlighters

SOURCE CODE

Formating visualizations:

i) Format Pane:

On the left side as the tableau interface you find the format pane. It allows you to format various aspects of your visualization, such as fonts, colors, 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:

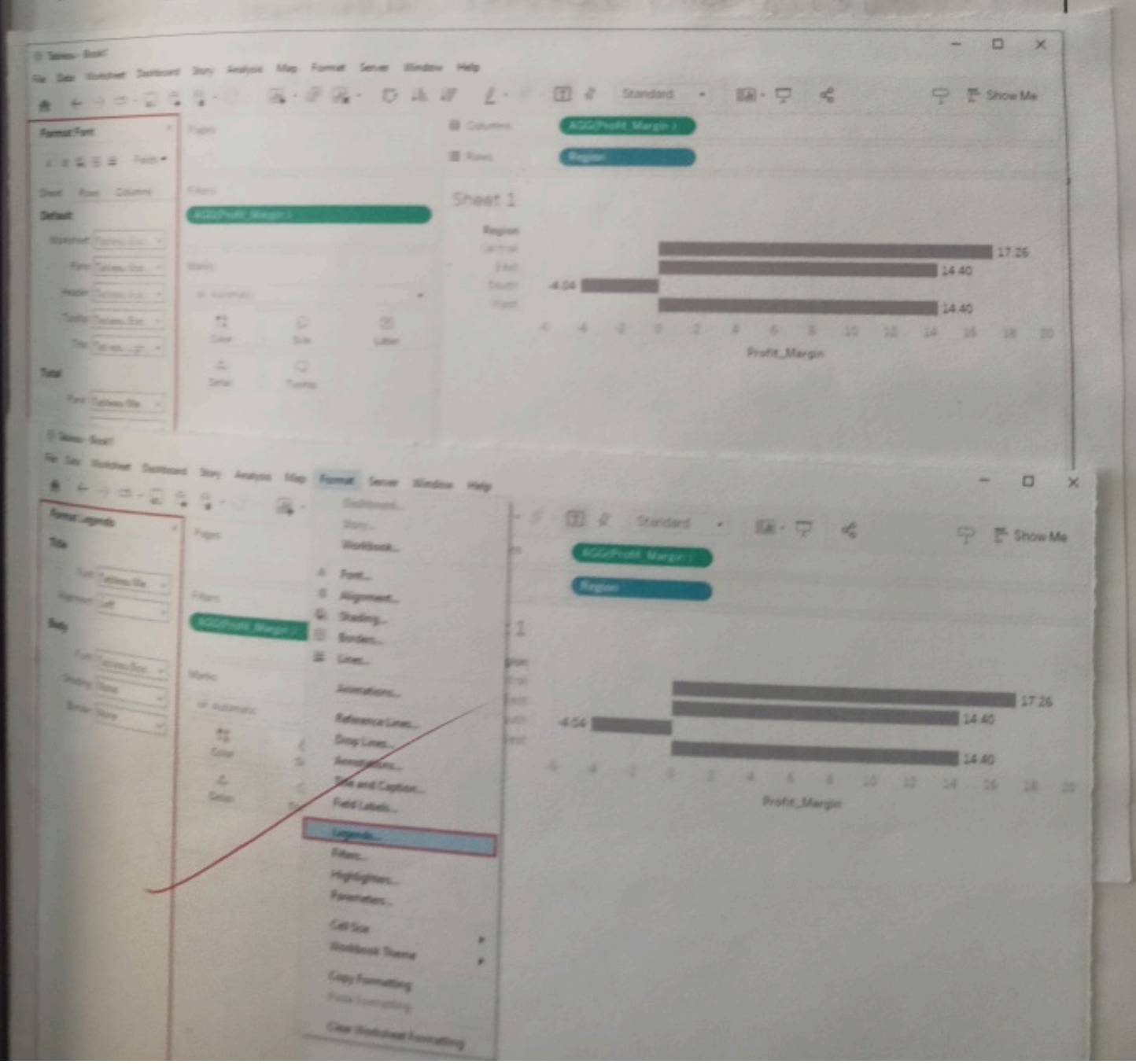
The mark card, located above your visualization offers formatting options specific to the type of marks Card to access these options and modify how your data is represented.

3. Axis and Gridlines:

You can format axis lables, fitlesand gridlines to improve the readability of your visualization. Rig

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 visualization. You can change colors, labels, and the position of legends to match your data.



Formatting Tools and Menus

Tableau provide 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 Work Sheet 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 dashboard the Dashboard menu allows you to format the entire dashboard layout, including background, size and title.

of The View:

Every view has a table in some form, which may include rows, columns, headers, axes, pa
cells, and marks. Views can optionally include tooltips, titles, captions, field labels, and le

PROCEDURE

Step-1: To show and unshow headers:

a. Right-click (control-click on Mac) the headers in the view and select Show Hea

b. Select the field in the view whose headers you want to show and select Show Hea
on the field menu.

Step-2: To show and unshow axis:

a. Right-click (control-click on MOUSE) the axis in the view and select Show Hea
to clear the check mark next to this option.

b. Right-click (control-click on MOUSE) the measure in the view whose axis you w
to show and select Show Header on the field menu.

Step-3: To Disable tooltip commands:

a. If you don't want users to be able to access tooltip commands, you can disable th

Click Tooltip on the Marks card or select **Worksheet >Tooltip.**

b. In the Edit Tooltip dialog box, clear the Include command buttons check box

Step-4: To show or hide titles in a worksheet

a. From the toolbar menu, click **Worksheet > Show Title.**

b. On the toolbar, **click the drop-down arrow on the Show/Hide** Cards button
select **Title from the context menu.**

Toggle the check mark on or off to show or hide the title.

SOURCE CODE

formatting specific parts of the view
Tableau lets you format specific elements of your

visualization:

1. Annotations:
You can add annotations to your visualization to highlight important points or provide additional context. format these annotations using the options available when you right-click on an annotations.

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.

the bottom & select hi
menu ③ Edit the button to Specify the toast obj
board and add a title button
s in calculated fields in tableau? to Show or hide.
a way to combine multiple dimension members
gle, higher level category 1. Create a group
ps in calculate fields. 3. Use groups in
archy in tableau? aggregations 4. Editing group.
erarchy in tableau, we have to drag
ne fields onto each other in the data

eau?

Tableau is a visualization, or viz
reate to explore data.

ableau?

a visual representation of one or more data
source. It allows you to modify a viz
alization type, user experience, and
of analysis all in one pane.

26/10

Experiment-10

Date: 26/10/24

AIM

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

PROCEDURE

I. Editing and Formatting Axes

Step-1: Double-click an axis to open the Edit Axis dialog box and change the axis configuration and formatting

Step-2: To select the marks associated with the axis, right-click the axis and select Marks.

Step-3: 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**.

Step-4: Try remaining options of axis

SOURCE CODE

Editing and formating 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 of 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 following 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 value, scale, or range according to your needs.

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 field)

Calculated field can be used if you need to create customized logic for manipulating certain data type of data values. There are a large- large of functions available Tableau Data.

Data provides enable you to rearrange the columns and rows in a report so you can view data from different perpectives.