



ERODE SENGUNTHAR ENGINEERING COLLEGE

(APPROVED BY AICTE, NEW DELHI & PERMANENTLY AFFILIATED TO ANNA UNIVERSITY, CHENNAI)

ACCREDITED BY NBA, NEW DELHI, NAAC WITH GRADE "A" & IE(I), KOLKATA)

PERUNDURAI, ERODE-638 057.

An Autonomous Institution

BONAFIDE CERTIFICATE

Register No.

Certified that is the Bonafide Record of Work Done

Name of the Student : _____

Branch : _____

Name of the Lab : _____

Faculty Incharge

Head of the Department

Submitted for the End Semester Practical

Held on

Internal Examiner

External Examiner

INDEX

S.NO	DATE	NAME OF THE EXPERIMENT	PAGE NO	MARKS	FACULTY SIGN
DATA VISUALIZATION					
1.		Acquiring and Plotting Data			
2.		Time-Series Analysis – Stock Market			
3.		Visualization Of Massive Dataset – Finance			
4.		Visualization on Streaming Dataset			
5.		Text Visualization Using Web Analytics			
NATURAL LANGUAGE PROCESSING					
1.		Word Analysis & Morphology			
2.		N-Grams			
3.		POS Tagging			
4.		Building Chunker			
5.		Build Chatbot			

DATA VISUALIZATION

Ex no:1

ACQUIRING AND PLOTTING DATA

Date:

Aim:

To acquiring and plotting data in PowerBi with Python libraries

Procedure:

Step 1: Install the necessary Python libraries such as pandas, matplotlib, seaborn, etc. You can use the command `pip install pandas matplotlib seaborn` in your terminal or command prompt.

Step 2: Load the data into a Pandas dataframe. You can use the `read_csv` function from the Pandas library to load a CSV file or `read_excel` function to load an Excel file.

Step 3: Clean and pre-process the data as necessary using Pandas functions such as `dropna`, `fillna`, `groupby`, etc. (If required)

Step 4: Load that data to dataset of Power BI and Check that fields you want to visualize

Step 5: Add the custom visual to your PowerBI report and use it to display the plot.

Program:

In cmd, install required libraries

```
>>pip install seaborn >>pip install matplotlib >>pip install pandas
```

Power BI desktop, Get data >> more >> Search "Python script"

In Python script,

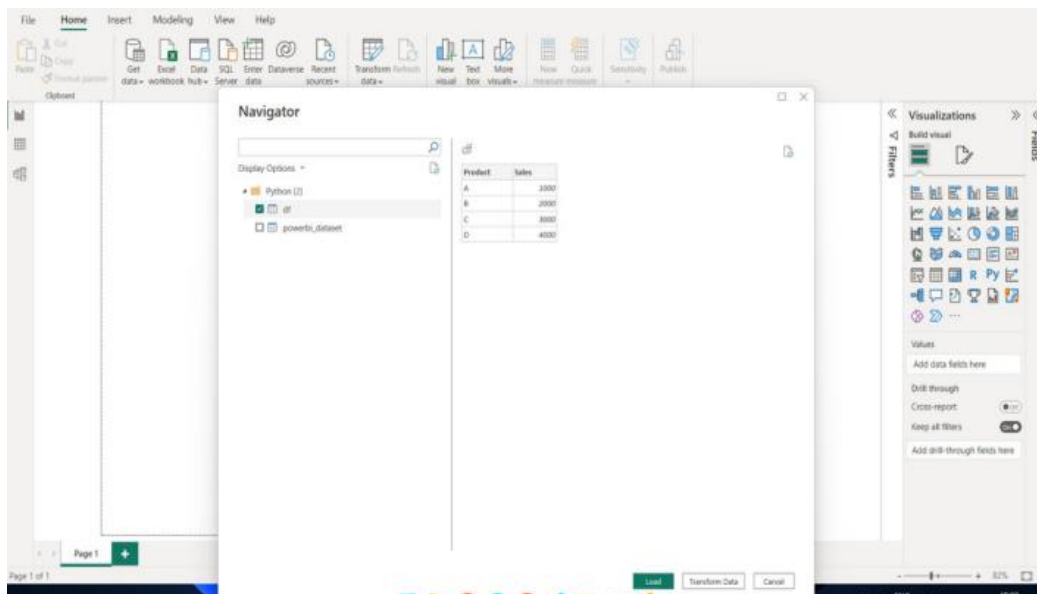
```
import
pandas as
pd#
Sample
data
```

```
data = {'Product': ['A', 'B',  
                  'C', 'D'], 'Sales': [1000,  
                                      2000, 3000, 4000]}
```

Convert the data to a pandas
DataFrame df =
pd.DataFrame(data)

Save the DataFrame as a Power
BI dataset
df.to_csv('powerbi_dataset.csv',
index=False)

Then go to >> Try a sample dataset



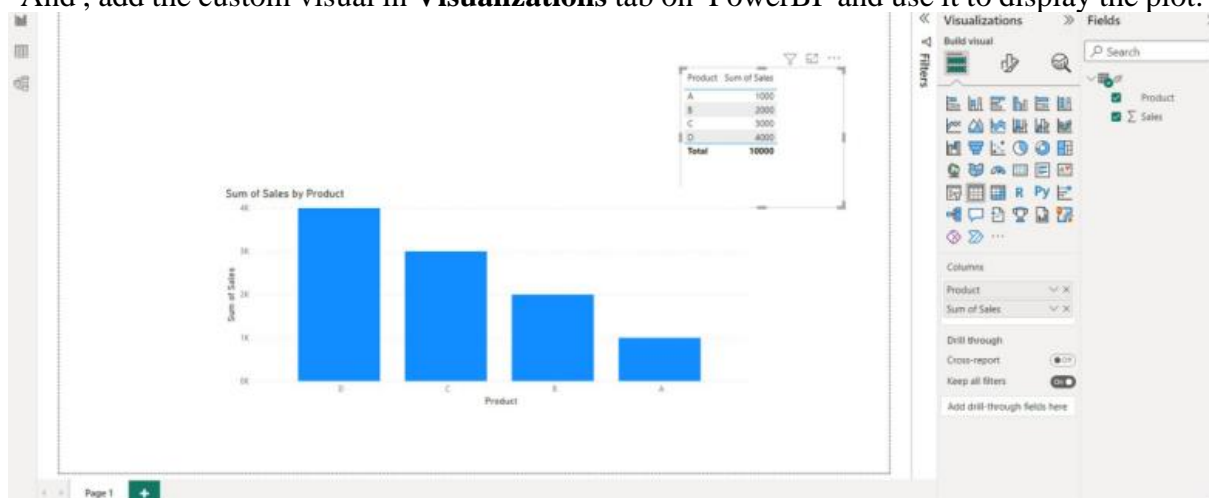
(IMG: for reference)

Select 'df' dataset at Navigator and Click “**Load**” to load

dataIn Fields, Check out the field which you want to

visualize

And , add the custom visual in **Visualizations** tab on PowerBI and use it to display the plot.



OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Acquiring and plotting data in Power BI with Python libraries enables users to leverage the power of Python for advanced data analysis and visualization, while utilizing the reporting capabilities of Power BI.

Ex no:2

TIME-SERIES ANALYSIS – STOCK MARKET

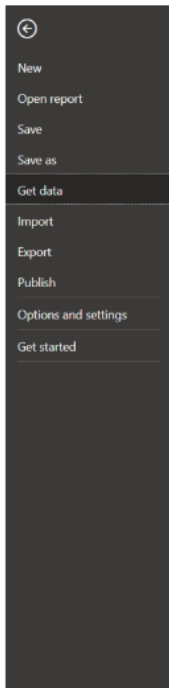
Date:

Aim:

To perform time series analysis using stock market dataset.

Procedure:

1. **Import Data:** Start by importing the stock market data into Power BI. The data can be sourced from various online data providers or through APIs.
2. **Clean and transform the data:** The next step is to clean and transform the data. This may involve removing null values, dealing with outliers, converting data types, and aggregating the data to the desired time intervals (daily, weekly, monthly, etc.).
3. **Create Time-Series Visualization:** In Power BI, you can create a time-series visualization by selecting the "Line and Stacked Column Chart" visualization type and placing the date field on the x-axis and the stock market data on the y-axis.
4. **Analyze Trends and Patterns:** Use the time-series visualization to analyse trends and patterns in the stock market data. You can use features such as trend lines, moving averages, and regression analysis to help identify patterns.
5. **Forecast Future Values:** Power BI has a built-in forecasting feature that can be used to forecast future stock market values based on historical data. To use this feature, select the time-series visualization, click on the "Analytics" tab, and select "Forecast."
6. **Share Insights:** Once you have analysed the data and created visualizations, you can share your insights with others by publishing your report to the Power BI service or by sharing it directly through Power BI Desktop.

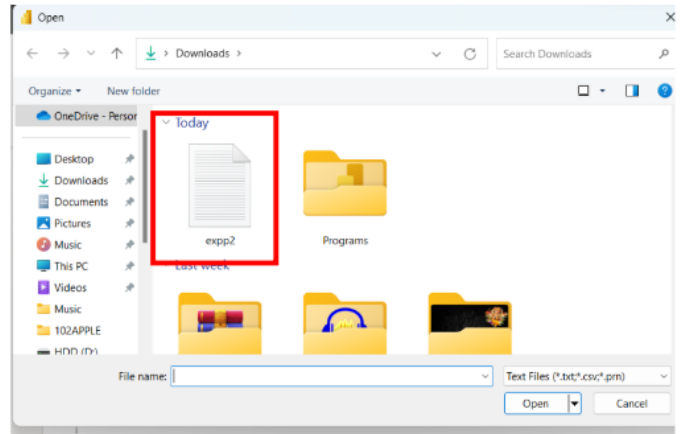


Get data

Most Common

- Excel workbook
- Power BI datasets
- Dataflows
- SQL Server database
- Azure Analysis Services database
- Text/CSV**
- Web
- OData feed
- Blank Query
- Power BI Template Apps

[Get data to get started](#) →

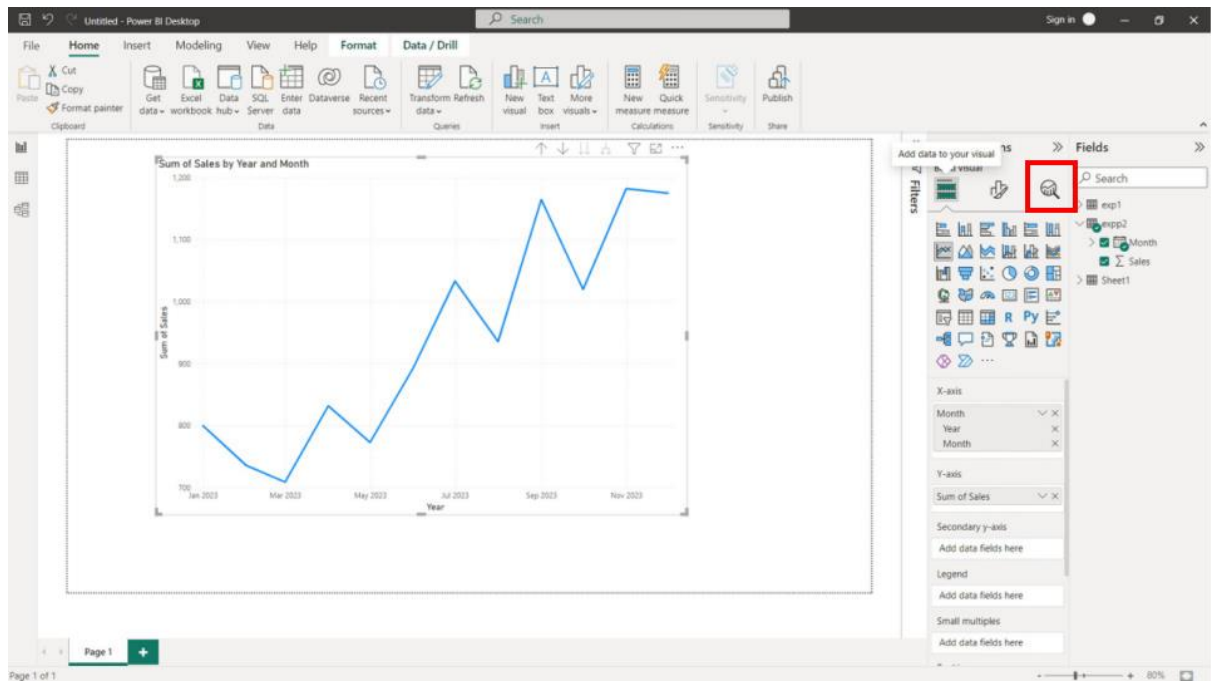
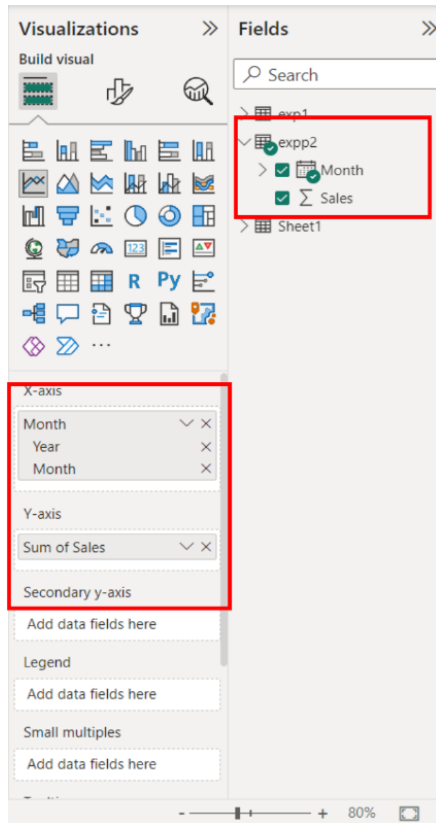


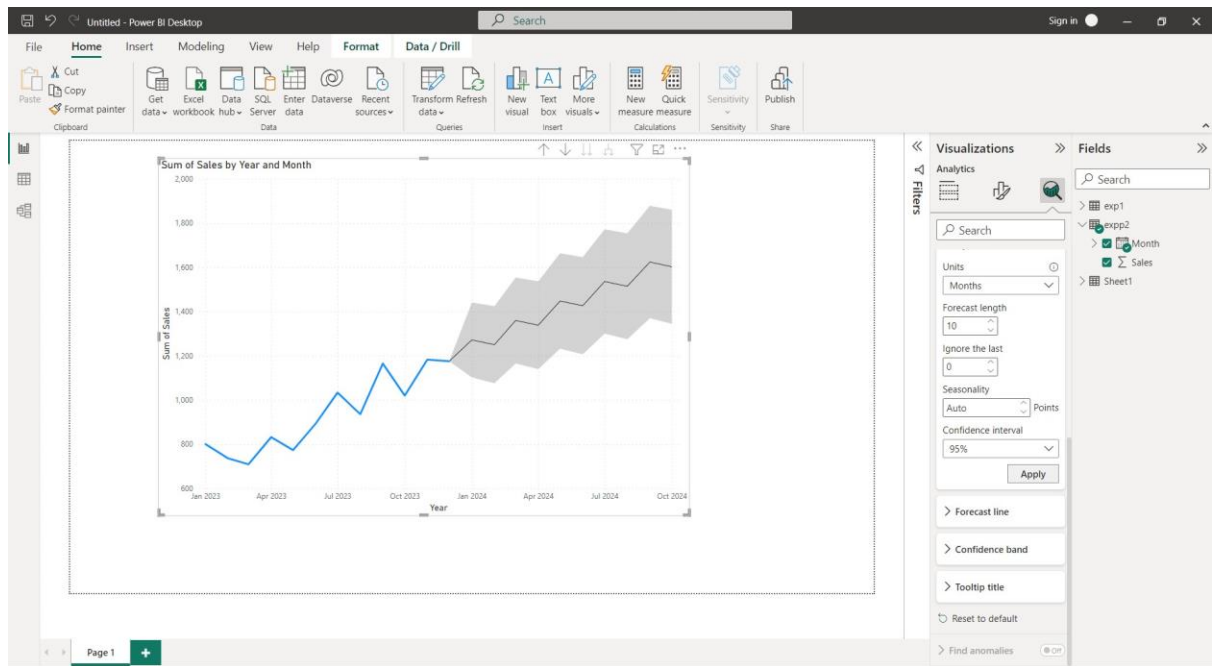
File Origin: 1252: Western European (Windows) | Delimiter: Comma | Data Type Detection: Based on first 200 rows

Month	Sales
01-01-2023	266
01-02-2023	145.9
01-03-2023	183.1
01-04-2023	119.3
01-05-2023	180.3
01-06-2023	168.5
01-07-2023	231.8
01-08-2023	224.5
01-09-2023	192.8
01-10-2023	122.9
01-11-2023	336.5
01-12-2023	185.9
02-01-2023	194.3
02-02-2023	149.5
02-03-2023	210.1
02-04-2023	273.3
02-05-2023	191.4
02-06-2023	287
02-07-2023	226
02-08-2023	303.6

The data in the preview has been truncated due to size limits.

[Extract Table Using Examples](#) **Load** [Transform Data](#) [Cancel](#)





Result:

OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Successfully performed time series analysis on stock market dataset, identifying key trends and patterns using statistical modeling techniques.

Ex no:3 VISUALIZATION OF MASSIVE DATASET – FINANCE

Date:

Aim:

To gain insights from a massive finance dataset and inform business decisions through a visually appealing and interactive dashboard.

Procedure:

1. Data Gathering: Collect the finance dataset from a reliable source and clean it to remove any irrelevant or duplicate data.
2. Data Modelling: Develop a data model that can efficiently handle the large volume of data.
3. Visual Design: Create a visually appealing and interactive dashboard that presents the data in a clear and easy-to-understand manner.
4. Analysis: Explore the dataset using various analytical techniques to uncover trends, patterns, and insights.
5. Reporting: Summarize the findings in a concise report and share it with the relevant stakeholders.

Riby Data Set.csv

File Origin		Delimiter		Data Type Detection				
1252: Western European (Windows) -		Comma -		Based on first 200 rows -				
Region	Country	Item Type	Sales Channel	Order Priority	Order Date	Order ID	Ship Date	Units
Sub-Saharan Africa	South Africa	Fruits	Offline	M	7/27/2012	443368995	7/28/2012	
Middle East and North Africa	Morocco	Clothes	Online	M	9/14/2013	667593514	10/19/2013	
Australia and Oceania	Papua New Guinea	Meat	Offline	M	5/15/2015	940955585	06/04/2015	
Sub-Saharan Africa	Djibouti	Clothes	Offline	H	5/17/2017	880811536	07/02/2017	
Europe	Slovakia	Beverages	Offline	L	10/26/2016	174590194	12/04/2016	
Asia	Sri Lanka	Fruits	Online	L	11/07/2011	830192887	12/18/2011	
Sub-Saharan Africa	Seychelles	Beverages	Online	M	1/18/2013	425793445	2/16/2013	
Sub-Saharan Africa	Tanzania	Beverages	Online	L	11/30/2016	659878194	1/16/2017	
Sub-Saharan Africa	Ghana	Office Supplies	Online	L	3/23/2017	601245963	4/15/2017	
Sub-Saharan Africa	Tanzania	Cosmetics	Offline	L	5/23/2016	739008080	5/24/2016	
Asia	Taiwan	Fruits	Offline	M	02/09/2014	732588374	2/23/2014	
Middle East and North Africa	Algeria	Cosmetics	Online	M	2/18/2011	761723172	2/24/2011	
Asia	Singapore	Snacks	Online	C	1/28/2013	178461303	02/07/2013	
Australia and Oceania	Papua New Guinea	Clothes	Offline	L	6/20/2011	647164094	7/14/2011	
Asia	Vietnam	Personal Care	Online	M	04/04/2010	314505374	05/06/2010	
Sub-Saharan Africa	Uganda	Personal Care	Online	M	6/19/2014	539471471	7/21/2014	
Sub-Saharan Africa	Zimbabwe	Office Supplies	Offline	C	3/28/2011	953361213	04/08/2011	
Sub-Saharan Africa	Ethiopia	Cosmetics	Online	M	07/07/2011	807785928	7/25/2011	
Europe	France	Cosmetics	Online	M	12/07/2015	324669444	1/18/2016	
Central America and the Caribbean	The Bahamas	Personal Care	Online	C	1/19/2011	246248090	2/21/2011	

Extract Table Using Examples

Load

Transform Data

Cancel

Extract Table Using Examples

Load

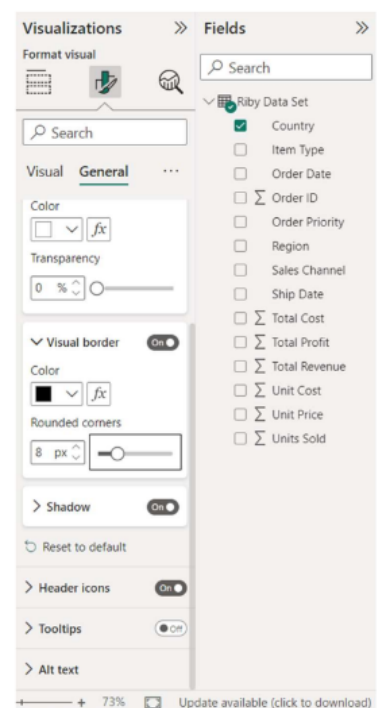
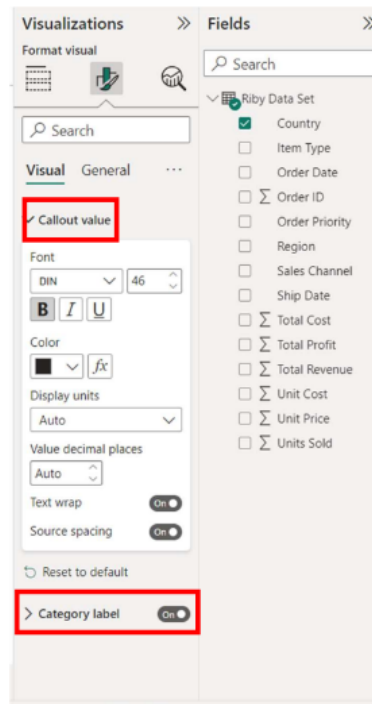
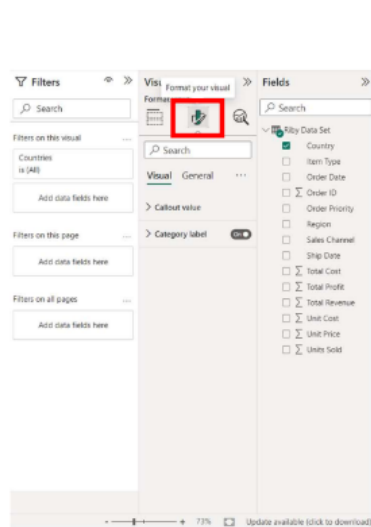
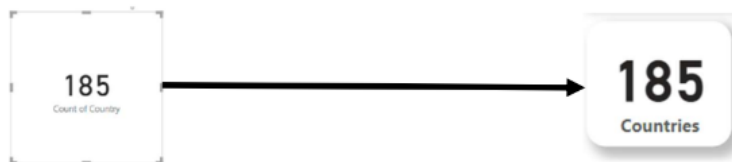
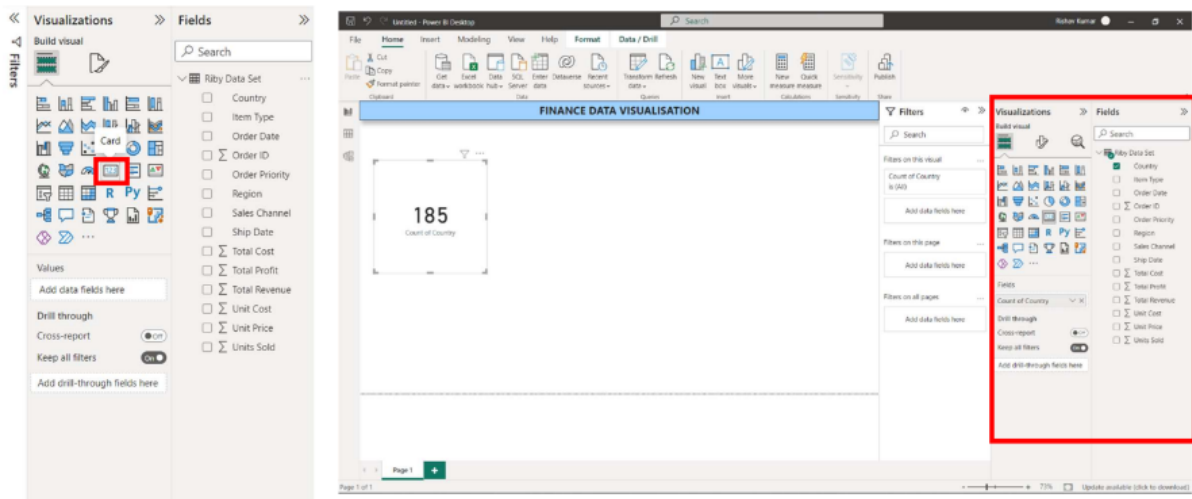
Transform Data

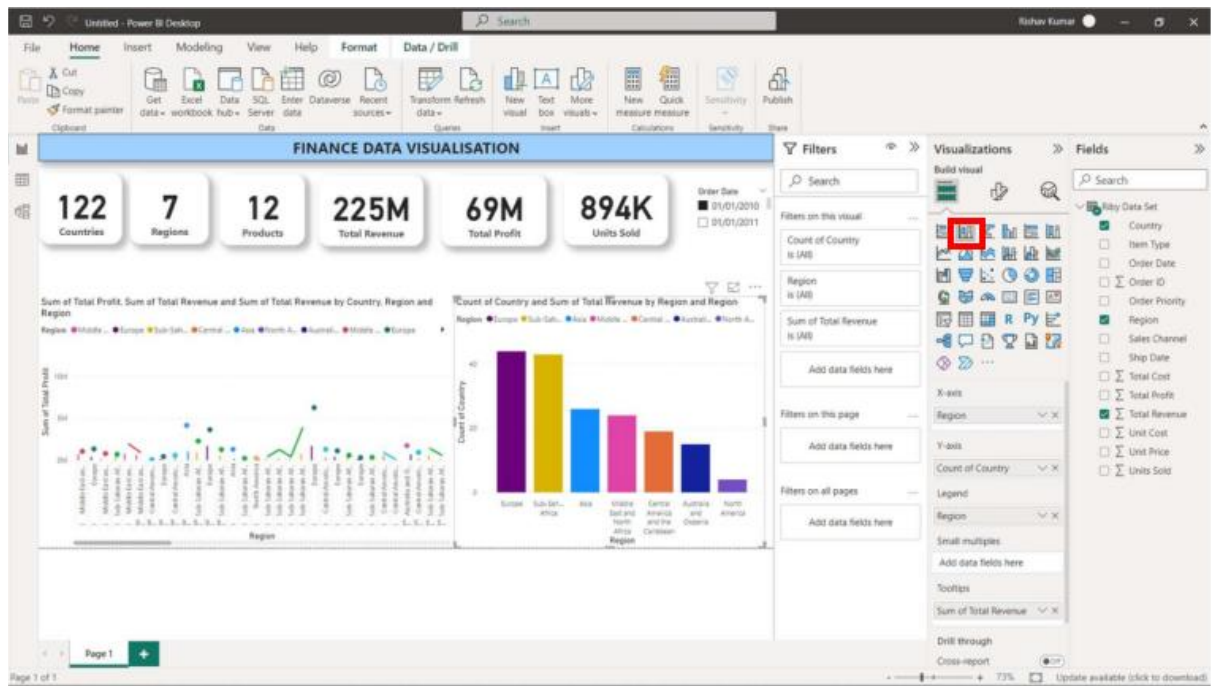
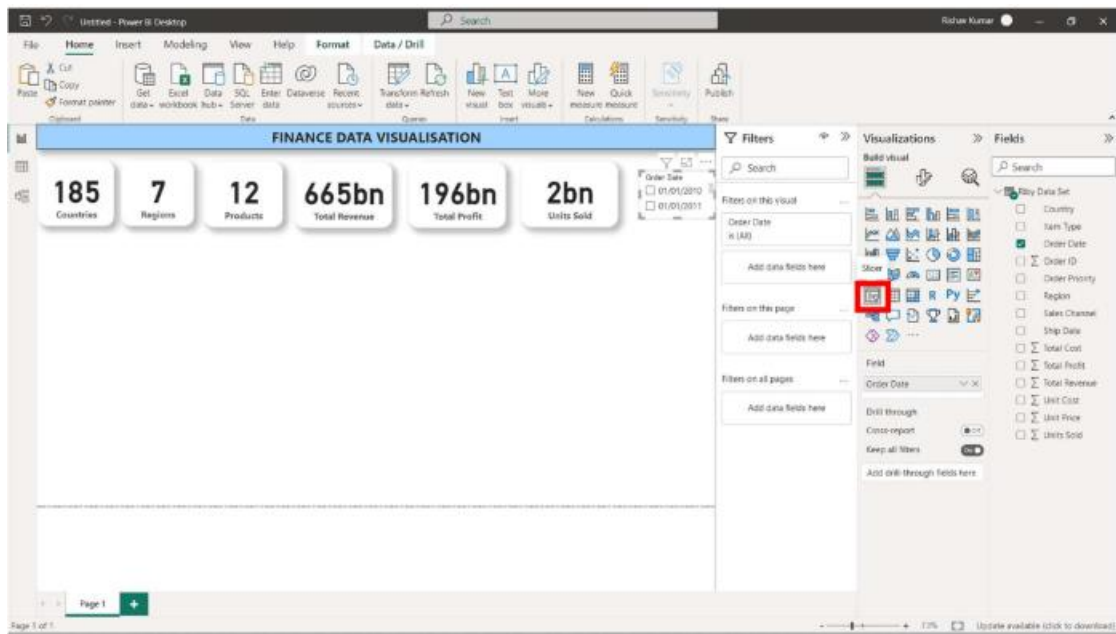
Cancel

Load

Riby Data Set
43 MB from Riby Data Set.csv

Cancel





OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

The finance dataset was analysed with precision, and the resulting insights were used to create a visually appealing and interactive dashboard, which was successfully deployed to inform business decisions.

Ex no:4

VISUALIZATION ON STREAMING DATASET

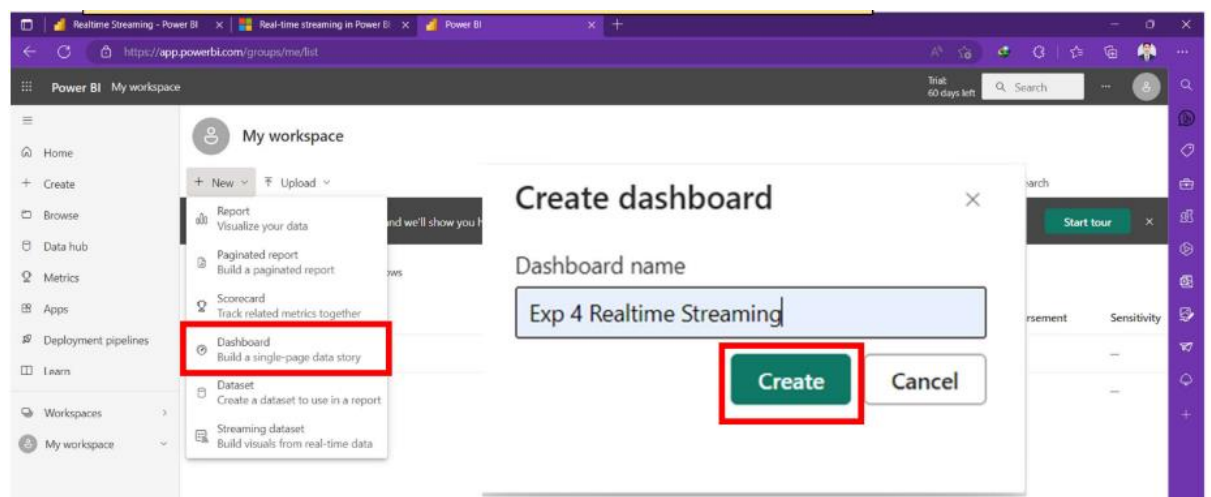
Date:

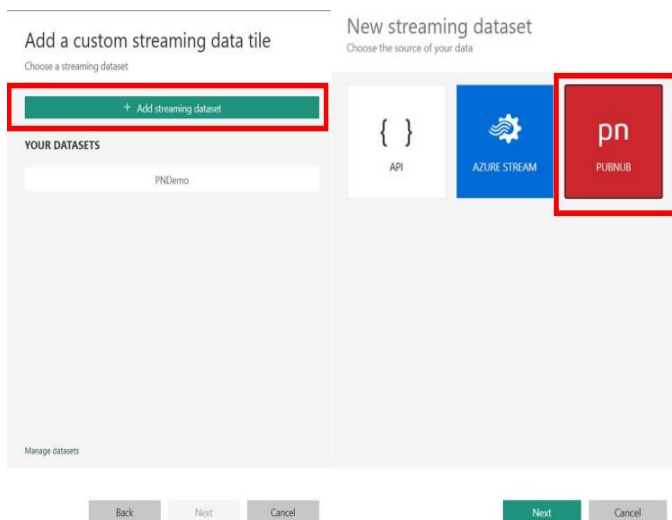
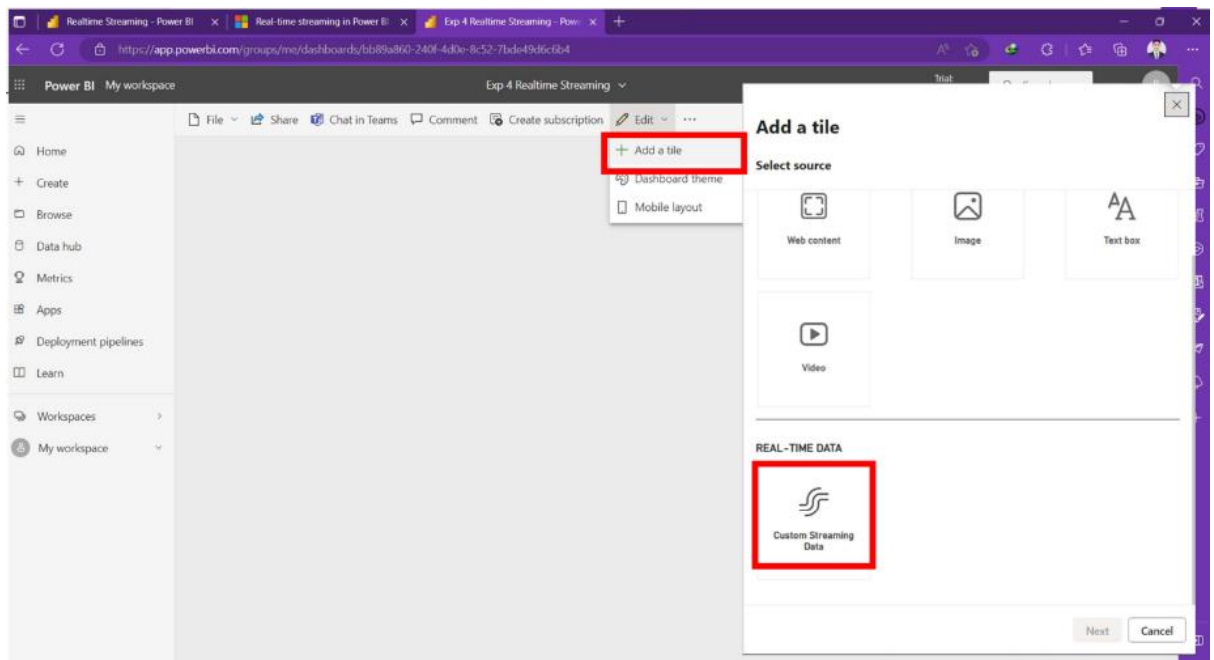
Aim:

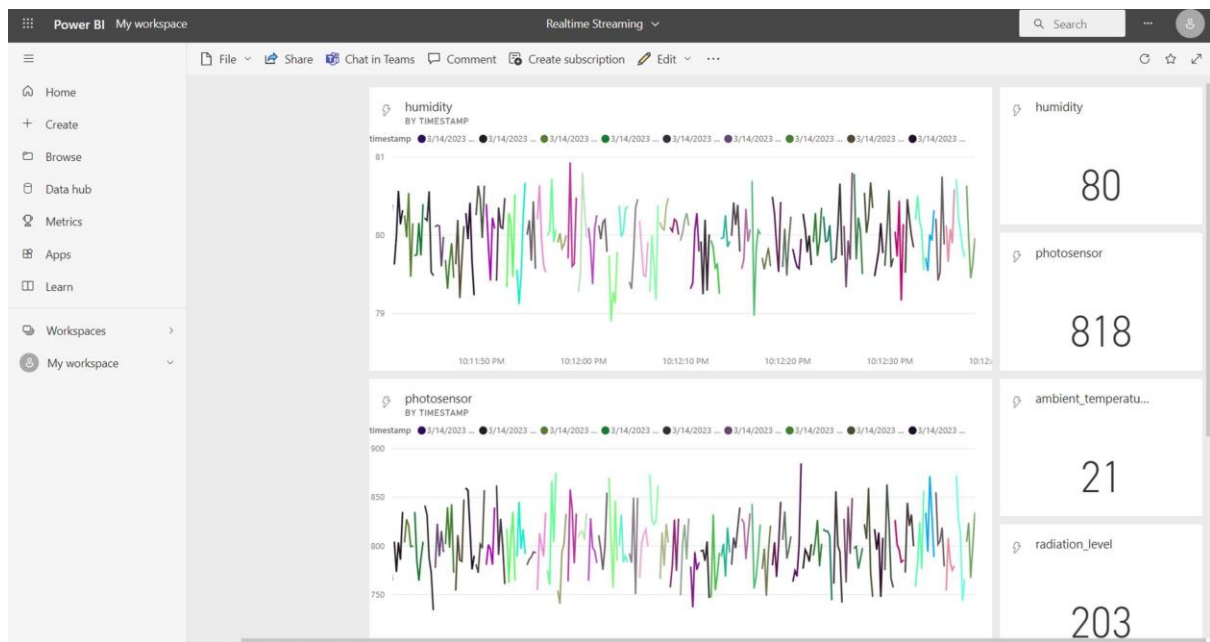
The aim of visualizing a streaming dataset in Power BI is to create real-time data visualizations that update dynamically as new data is received from a streaming source. This can help users quickly identify trends, patterns, and anomalies in their data.

Procedure:

1. Connect to your streaming data source: In Power BI, go to the "Home" tab and click on "Get Data". Select "Streaming" under the "Other" category and choose your streaming data source.
2. Configure the streaming dataset: In the "Configure Streaming Dataset" window, enter the required information about your data source, such as the streaming URL, schema, and data types.
3. Create visuals: Once your streaming dataset is configured, you can create visualizations by dragging and dropping fields onto the canvas. You can choose from various visualization types, including charts, maps, and tables.
4. Publish your report: Once you have created your visuals, you can publish your report to the Power BI service. This will allow you to view and share your real-time data visualizations with others.







OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Successfully visualize streaming data in Power BI to create real-time data visualizations that provide valuable insights into patterns, trends, and anomalies, and make data-driven decisions on the fly.

Ex no:5 TEXT VISUALIZATION USING WEB ANALYTICS

Date:

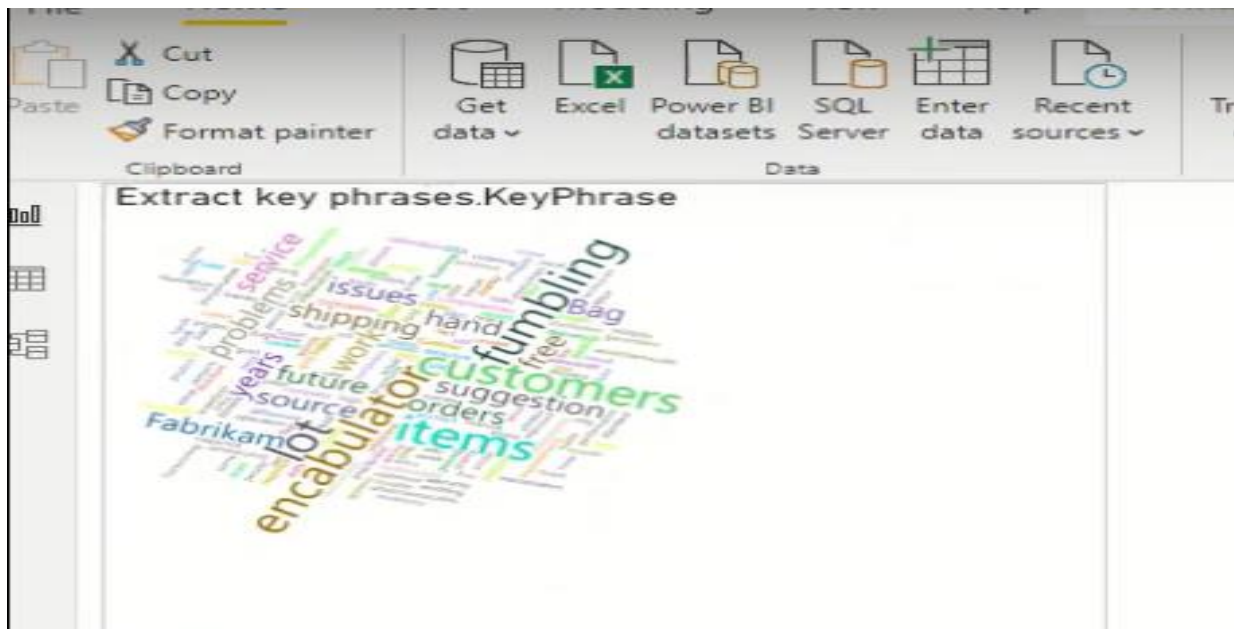
Aim:

The aim of this project is to use web analytics data to create a visualization in Power BI that provides insights into website performance.

Procedure:

1. **Data collection:** The first step is to collect web analytics data from your website. This can be done using tools such as Google Analytics or Adobe Analytics.
2. **Data preparation:** Once you have collected the data, you will need to clean and prepare it for analysis. This may involve removing duplicates, fixing errors, and formatting data in a way that is compatible with Power BI.
3. **Data modeling:** The next step is to create a data model in Power BI that will allow you to analyze and visualize the data effectively. This involves creating relationships between different tables and defining calculations and measures that will be used in the visualization.
4. **Visualization design:** With the data model in place, you can now start designing your visualization. This may involve creating charts, graphs, and tables that display the data in a meaningful way. You may also want to add interactive elements such as filters and slicers that allow users to explore the data in more detail.
5. **Report creation:** Once you have designed your visualization, you can create a report in Power BI that brings everything together. This report can be shared with others in your organization, allowing them to gain insights into website performance.

	A ₀	A ₀ name	A ₀ userid	A ₀ emailaddress	A ₀ datetime	A ₀ topic	A ₀ subject	A ₀ comment
1	0	Donald Erie	derie	don@example.com	09:04:00	shipping	Insufficient packaging	I ordered three widgets last week and just received them. I am VERY dis-
2	0	Jake Bering	jakeb	jake.bering@example.com	09:37:00	fulfillment	Wrong items again	Can't believe you fools shipped me the wrong items AGAIN. If you won't
3	0	Ann Huron	anhnr	ahuron@example.com	10:11:00	praise	Fantastic!	Wow! I had NO IDEA that reciprocating grambles such high quality ev-
4	1203	David Columbia	davidc	davidc78@example.com	10:15:00	fulfillment	Re: Wrong Items again	Jake, I've had similar problems tracking down another source thingam...
5	0	Maria Michigan	mariam	maria@example.com	10:44:00	other	Photo on framistan brochure	That hand model needs a manicure, stat!
6	0	Juan Brazos	juann	juan1@example.com	11:01:00	techsupport	Bag of Holding	I put my wedding ring in a Bag Holding I purchased from you guys (for ...
7	1205	Jake Bering	jakeb	jake.bering@example.com	12:23:00	fulfillment	Re: Wrong Items again	Dad, sorry, it's whatchamacallit I'm looking for. I should have been ...
8	0	Susan Colorado	susanc	suscol@example.com	13:15:00	techsupport	Problems with retro-encabulator	Hi folks. According to your encabulator brochure, the hydroptic mar...
9	1208	Ron Puget	ronrp	ronnyp@example.com	13:57:00	fulfillment	Re: Wrong Items again	Jake, it's probably none my business, but you will probably get a bette...
10	1206	Elena Pecos	elenab	pecos@example.com	14:14:00	other	Re: Photo on framistan brochure	Maria, I'm pretty sure the folks at Fabrikam use photos actual custome...
11	1207	Darius Williams	dariusw	go-dawgs@example.com	15:03:00	techsupport	Bag of Holding	Buan, are you sure it's not a Bag Transmuting? They look very similar, b...
12	1209	Fabrikam Support	fsupport	support@Fabrikam.com	15:16:00	techsupport	Re: Problems with retro-encabulator	Susan, the hydroptic marzelveans MUST be fitted properly to the run...
13	1209	Roy Ontario	royo	roy_it@example.com	15:29:00	techsupport	Re: Problems with retro-encabulator	Have you tried turning it off and back on again? The encabulator runs ...
14	1210	Jake Bering	jakeb	jake.bering@example.com	15:33:00	fulfillment	Re: Wrong Items again	Ron, you're quite right: it is none your business! The terms I used are q...
15	1214	Susan Colorado	susanc	suscol@example.com	15:57:00	techsupport	Re: Problems with retro-encabulator	Roy, thanks for the suggestion. It did help a little, but did not entirely e...
16	1203	Fabrikam Sales	fsales	sales@fabrikam.com	16:04:00	fulfillment	Re: Wrong Items again	Jake, I'm very sorry that you've had this problem for so long. Our syste...
17	0	Wen Niagara	wenn	wen@example.com	16:19:00	productinfo	When are the new models coming out?	I am in need some doohickies in the near future. If I remember correc...
18	1209	Susan Colorado	susanc	suscol@example.com	16:47:00	techsupport	Re: Problems with retro-encabulator	Thanks. One the marzelveans was installed backward, so it's my fault ...
19	1217	Jake Bering	jakeb	jake.bering@example.com	17:12:00	fulfillment	Re: Wrong Items again	That's all well and good for ME, and I do appreciate it, but what about ...
20	0	Riya Champlain	riyab	riya@example.com	18:49:00	suggestion	Gizmo colors	Hi, have you ever considered making gizmos in colors other than chan...



OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Successfully use web analytics data to create a Power BI visualization that provides insights into website performance, allowing for data-driven decisions and improved user experience.

NATURAL LANGUAGE PROCESSING

Ex No: 1

WORD ANALYSIS & MORPHOLOGY

Date:

Aim:

The aim of this program is to analyze and examine the morphology of words using the Python Natural Language Toolkit (NLTK) library.

Procedure:

1. Import necessary libraries, such as the NLTK library and the word_tokenize function.
2. Define the input text to be analyzed.
3. Tokenize the input text into individual words using the word_tokenize function from the NLTK library.
4. Apply morphology analysis using the PorterStemmer algorithm from the NLTK library.
5. Store the stemmed words in a new list.
6. Print out the original text, the tokenized words, and the stemmed words.

Program:

```
import nltk

from nltk.tokenize import word_tokenize

# input text
text = "The quick brown fox jumps over the lazy dog."

# tokenizing the text into words
words = word_tokenize(text)

# applying morphology analysis
morph = nltk.PorterStemmer()

stemmed_words = [morph.stem(word) for word in words]

# output
print("Original Text: ", text)
print("Tokenized Words: ", words)
print("Stemmed Words: ", stemmed_words)
```

Output:

Original Text: The quick brown fox jumps over the lazy dog.

Tokenized Words: ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog', '.']

Stemmed Words: ['the', 'quick', 'brown', 'fox', 'jump', 'over', 'the', 'lazi', 'dog', '.']

OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Thus the Program for Word Analysis & Morphology written and executed successfully.

Ex No: 2

N-GRAMS

Date:

Aim:

The aim of this code is to generate the bi-gram and tri-gram sequences of words from a given input sentence.

Procedure:

1. Import the required module - "ngrams" from the NLTK package.
2. Define a string variable "s" that contains the input sentence.
3. Split the sentence into individual words using the "split()" function and store it in a variable called "w".
4. Generate the bi-gram sequences of words using the "ngrams()" function from the NLTK package, by passing the word list "w" and the value "2" to specify the length of the sequence as a parameter. Store the result in a variable called "bi_gram".
5. Generate the tri-gram sequences of words using the same "ngrams()" function, but by passing the value "3" as a parameter. Store the result in a variable called "tri_gram".
6. Print the bi-gram and tri-gram sequences using the "print()" function, along with appropriate messages.

Program:

```
from nltk.util import ngrams

s = "This My laptop it's very comfortable for me"

w = s.split()

bi_gram = list(ngrams(w, 2))

tri_gram = list(ngrams(w, 3))

print("Bi-gram:", bi_gram)

print("Tri-gram:", tri_gram)
```

Output:

Bi-gram: [('This', 'My'), ('My', 'laptop'), ('laptop', "it's"), ("it's", 'very'), ('very', 'comfortable'), ('comfortable', 'for'), ('for', 'me')]

Tri-gram: [('This', 'My', 'laptop'), ('My', 'laptop', "it's"), ('laptop', "it's", 'very'), ("it's", 'very', 'comfortable'), ('very', 'comfortable', 'for'), ('comfortable', 'for', 'me')]

OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Thus the Program for N-grams written and executed successfully.

Ex No: 3

POS TAGGING

Date:

Aim:

To perform POS tagging on a given sentence using NLTK in Python.

Procedure:

1. First, we import the **nltk** library and specifically the **word_tokenize** function which tokenizes the input sentence into individual words.
2. We define the input sentence as a string variable.
3. We tokenize the sentence into individual words using **word_tokenize**.
4. We then use the **pos_tag** function from **nltk** to perform POS tagging on the tokenized words.
5. Finally, we print out the POS tagged words.

Program:

```
import nltk  
from nltk.tokenize import word_tokenize  
sentence = "The quick brown fox jumps over the lazy dog."  
tokens = word_tokenize(sentence)  
pos_tags = nltk.pos_tag(tokens)  
print(pos_tags)
```

Output:

[('The', 'DT'), ('quick', 'JJ'), ('brown', 'NN'), ('fox', 'NN'), ('jumps', 'VBZ'), ('over', 'IN'), ('the', 'DT'), ('lazy', 'JJ'), ('dog', 'NN'), (',', ',')]

OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Thus the Program for POS Tagging written and executed successfully.

Ex No: 4

BUILDING CHUNKER

Date:

Aim:

The aim of the above code is to parse a given input sentence using regular expressions and chunk it into phrases.

Procedure:

1. Import the necessary module nltk which provides natural language processing functionalities.
2. Define a regular expression-based grammar pattern that specifies the structure of the phrases to be extracted.
3. Initialize the nltk.RegexpParser() object with the defined grammar pattern.
4. Provide an input sentence that needs to be parsed and chunked into phrases.
5. Tokenize the input sentence into individual words using nltk.word_tokenize().
6. Tag each word with its corresponding Part-of-Speech (POS) tag using nltk.pos_tag().
7. Parse the tagged sentence using the initialized parser object, which creates a parse tree with phrases as nodes and words as leaves.
8. Traverse the parsed tree and extract the Noun Phrases using the subtrees() method.
9. Check the label of each subtree and extract the ones labeled as "NP".
10. Print the extracted Noun Phrases.

Program:

```
import nltk

# Define a grammar pattern using regular expressions
grammar_pattern = r"""
    NP: {<DT>?<JJ>*<NN>} # chunking Noun Phrases
    VP: {<VB.*><NP|PP|CLAUSE>+<$>} # chunking Verb Phrases
    PP: {<IN><NP>} # chunking Prepositional Phrases
    CLAUSE: {<NP><VP>} # chunking Clauses
    ""

# Initialize the parser with the grammar pattern
parser = nltk.RegexpParser(grammar_pattern)
```

```

# Input sentence
s = "The quick brown fox jumped over the lazy dog."

# Tokenize the input sentence
tokens = nltk.word_tokenize(s)

# Tag the tokens with Part-of-Speech tags
pos_tags = nltk.pos_tag(tokens)

# Parse the tagged sentence using the parser
tree = parser.parse(pos_tags)

# Traverse the parsed tree and print the leaves
for subtree in tree.subtrees():
    if subtree.label() == 'NP':
        print(' '.join(word for word, tag in subtree.leaves()))

```

Output:

```

The
quick brown fox
the lazy dog

```

OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Thus the Program for Building Chunker written and executed successfully.

Ex No: 5

BUILD CHATBOT

Date:

Aim:

The given code implements a simple chatbot using the NLTK library in Python. The chatbot responds to a few pre-defined patterns of user input.

Procedure:

1. First, the code imports the necessary modules from the NLTK library - 'nltk' and 'Chat' from 'nltk.chat.util'.
2. The code defines a set of input-output patterns using regular expressions and corresponding responses. For example, if the user inputs 'Hi there!', the chatbot will respond with 'Hello!'.
3. These input-output patterns are stored in a list of tuples, where each tuple contains a regular expression pattern and a response for that pattern.
4. The code initializes a Chat object using the input-output patterns and a dictionary of reflections (used to map user pronouns to chatbot pronouns).
5. Finally, the code starts a conversation with the chatbot using the 'converse()' method of the Chat object.

Program:

```
import nltk

from nltk.chat.util import Chat, reflections

input_output_patterns = [
    (r'Hi there!', ['Hello!']),
    (r'How are you?', ['I am doing well, thank you. How are you?']),
    (r'What is your name?', ['My name is Chatbot.']),
    (r'Quit|Bye|Thank you', ['Goodbye!', 'Bye! Nice talking to you.'])
]

chatbot = Chat(input_output_patterns, reflections)

chatbot.converse()
```

Output:

> Hi there!

Hello!

> What is your name?

My name is Chatbot.

> How are you?

I am doing well, thank you. How are you?

> Quit

Goodbye!

OBSERVATION	(20)	
RECORD	(15)	
VIVA VOCE	(10)	
TOTAL	(45)	

Result:

Thus the Program for Build ChatBot written and executed successfully.