DAY 4 LAB EXPERIMENTS

Name: Rakshitha V B

Reg.no: 192324004

Dept: B.Tech AI & DS

Scenario: You are working on a project that involves analyzing customer reviews for a product. You have a dataset containing customer reviews, and your task is to develop a Python program that calculates the frequency distribution of words in the reviews.

Question: Develop a Python program to calculate the frequency distribution of words in the customer reviews dataset?

Solution:

```
import pandas as pd
from collections import Counter
import re

data = {
   "reviews": [
      "This product is very good and useful",
      "The quality of the product is excellent",
      "Very bad experience with this product",
      "The product quality is good and price is reasonable",
      "Excellent product and very good performance"
   ]
}
df = pd.DataFrame(data)
all_text = " ".join(df["reviews"]).lower()
words = re.findall(r'\b\w+\b', all_text)
word_freq = Counter(words)
print(word_freq)
```

```
df = pd.DataFrame(data)
all_text = " ".join(df["reviews"]).lower()
words = re.findall(r'\b\w+\b', all_text)
word_freq = Counter(words)

print(word_freq)

Counter({'product': 5, 'is': 4, 'very': 3, 'good': 3, 'and': 3, 'the': 3, 'this': 2, 'quality': 2, 'excellent': 2, 'useful': 1, 'of': 1, 'bad': 1, 'experience': 1,
```

Scenario: You are a data analyst working for a marketing research company. Your team has collected a large dataset containing customer feedback from various social media platforms. The dataset consists of thousands of text entries, and your task is to develop a Python program to analyze the frequency distribution of words in this dataset. Your program should be able to perform the following tasks:

- Load the dataset from a CSV file (data.csv) containing a single column named "feedback"
- with each row representing a customer comment.
- Preprocess the text data by removing punctuation, converting all text to lowercase, and
- eliminating any stop words (common words like "the," "and," "is," etc. that don't carry
- significant meaning).
- Calculate the frequency distribution of words in the preprocessed dataset.
- Display the top N most frequent words and their corresponding frequencies, where N is
- provided as user input.
- Plot a bar graph to visualize the top N most frequent words and their frequencies.

Question: Create a Python program that fulfills these requirements and helps your team gain insights from the customer feedback data.

Solution:

```
import pandas as pd
import re
import matplotlib.pyplot as plt
from collections import Counter

df = pd.read_csv("data.csv")

text = " ".join(df["feedback"].astype(str))
text = text.lower()
text = re.sub(r'[^\w\s]', '', text)

stop_words = {
    "the","and","is","in","to","of","for","on","with","this","that","it",
    "as","are","was","were","be","been","has","have","had","but","by",
    "from","at","or","an","a","we","you","they","their","our","your"
}
```

```
words = [word for word in text.split() if word not in stop_words]

word_freq = Counter(words)

N = int(input("Enter the value of N: "))

top_words = word_freq.most_common(N)

print("\nTop", N, "Most Frequent Words:")
for word, freq in top_words:
    print(f"{word} : {freq}")

labels, values = zip(*top_words)

plt.bar(labels, values)
plt.xlabel("Words")
plt.ylabel("Frequency")
plt.title(f"Top {N} Most Frequent Words in Customer Feedback")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```python
print("\nTop", N, "Most Frequent Words:")
for word, freq in top_words:
    print(f"{word} : {freq}")
```

```
Top 8 Most Frequent Words:
product : 8
quality : 6
very : 6
excellent : 4
experience : 4
service : 4
delivery : 4
performance : 3
```

Scenario: Suppose a hospital tested the age and body fat data for 18 randomly selected adults with the following result.

Question:

- Calculate the mean, median and standard deviation of age and %fat using Pandas.
- Draw the boxplots for age and %fat.
- Draw a scatter plot and a q-q plot based on these two variables

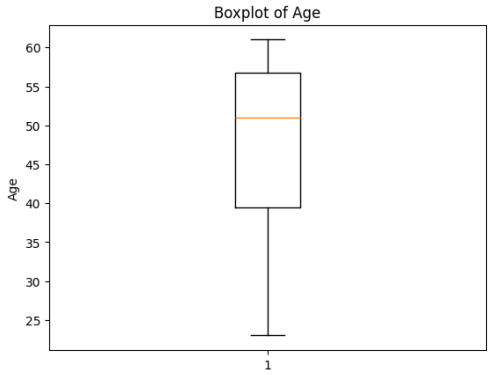
Solution:

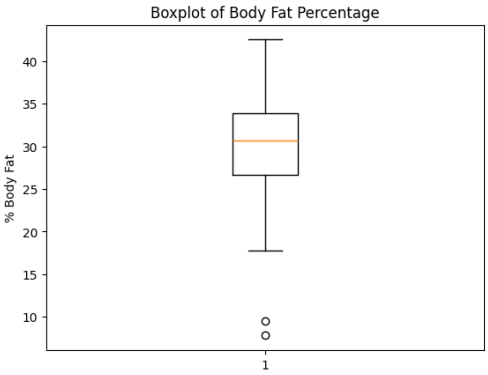
```
import pandas as pd

import matplotlib.pyplot as plt

import scipy.stats as stats


data = {

   "age": [23, 23, 27, 27, 39, 41, 47, 49, 50, 52, 54, 54, 56, 57, 58, 58, 60, 61],

   "fat": [9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 27.2, 31.2,

         34.6, 42.5, 28.8, 33.4, 30.2, 34.1, 32.9, 41.2, 35.7]

}


df = pd.DataFrame(data)


print("Mean:\n", df.mean())

print("\nMedian:\n", df.median())

print("\nStandard Deviation:\n", df.std())


plt.boxplot(df["age"])

plt.title("Boxplot of Age")

plt.ylabel("Age")

plt.show()


plt.boxplot(df["fat"])

plt.title("Boxplot of Body Fat Percentage")

plt.ylabel("% Body Fat")

plt.show()
```

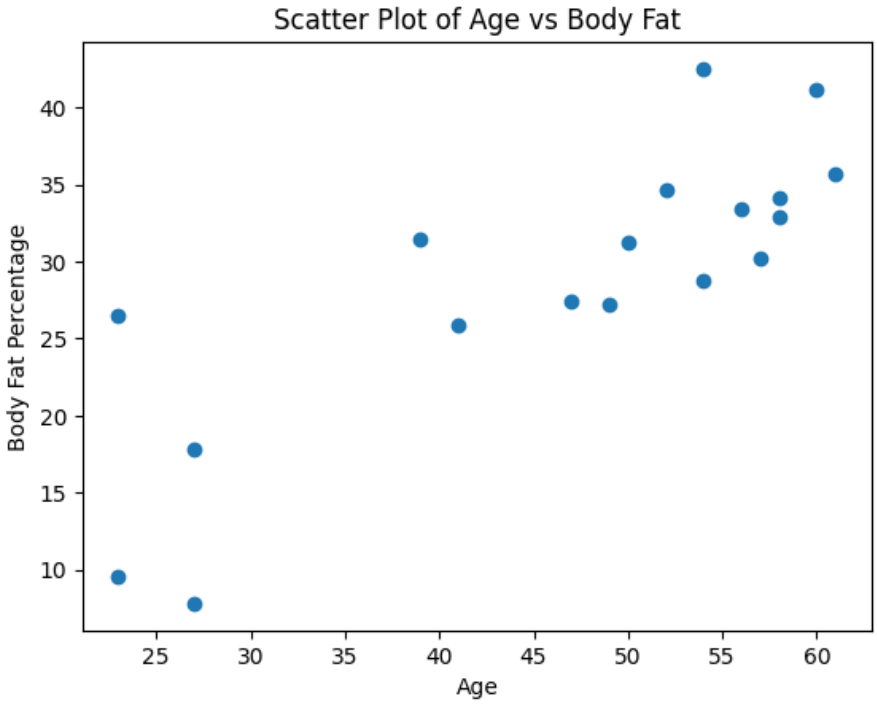
```
plt.scatter(df["age"], df["fat"])
plt.xlabel("Age")
plt.ylabel("Body Fat Percentage")
plt.title("Scatter Plot of Age vs Body Fat")
plt.show()

stats.probplot(df["age"], dist="norm", plot=plt)
plt.title("Q-Q Plot of Age")
plt.show()

stats.probplot(df["fat"], dist="norm", plot=plt)
plt.title("Q-Q Plot of Body Fat Percentage")
plt.show()
```

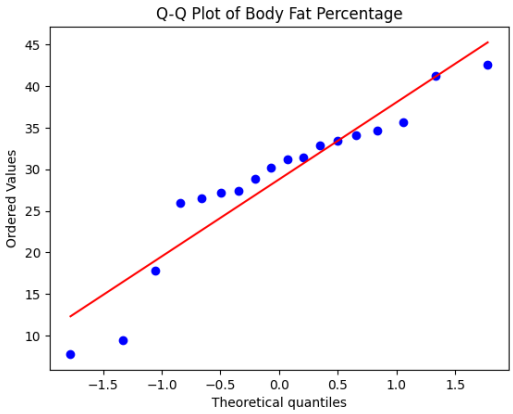
```
print("Mean:\n", df.mean())
print("\nMedian:\n", df.median())
print("\nStandard Deviation:\n", df.std())
```

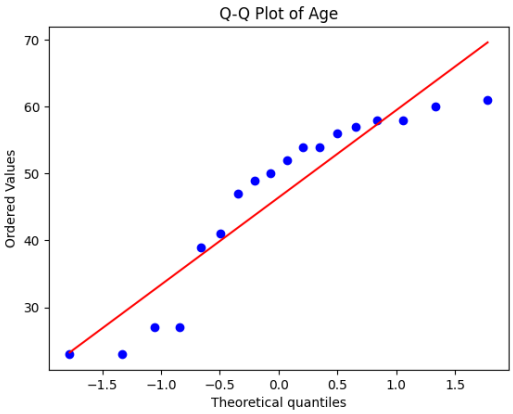
```
Mean:
 age    46.444444
fat    28.783333
dtype: float64

Median:
 age    51.0
fat    30.7
dtype: float64

Standard Deviation:
 age    13.218624
fat     9.254395
dtype: float64
```


Boxplot of Age
Age
60
55
50
45
40
35
30
25
1


Boxplot of Body Fat Percentage
% Body Fat
40
35
30
25
20
15
10
1


Scatter Plot of Age vs Body Fat
Body Fat Percentage
40
35
30
25
20
15
10
25
30
35
40
45
50
55
60
Age


Q-Q Plot of Body Fat Percentage
Ordered Values
45
40
35
30
25
20
15
10
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical quantiles


Q-Q Plot of Age
Ordered Values
70
60
50
40
30
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical quantiles

Scenario: You are a medical researcher investigating the effectiveness of a new drug in reducing blood pressure. You conduct a clinical trial with a sample of 50 patients who were randomly assigned to receive either the new drug or a placebo. After measuring their blood pressure levels at the end of the trial, you obtain the data for both groups. Now, you want to determine the confidence intervals for the mean reduction in blood pressure for both the drug and placebo groups.

Question: "What is the 95% confidence interval for the mean reduction in blood pressure for patients who received the new drug? Also, what is the 95% confidence interval for the mean reduction in blood pressure for patients who received the placebo?

Solution:

```
import numpy as np
import scipy.stats as stats
drug_reduction = np.array([
   12, 15, 14, 10, 13, 16, 18, 14, 15, 17,11, 13, 16, 14, 15, 17, 12, 14, 16, 18, 13, 15, 14, 16, 17 ])
placebo_reduction = np.array([ 2, 3, 1, 4, 2, 3, 2, 1, 3, 4, 2, 1, 3, 2, 4, 1, 2, 3, 1, 2, 3, 2, 1, 3, 2])


def confidence_interval(data, confidence=0.95):
   mean = np.mean(data)
   std = np.std(data, ddof=1)
   n = len(data)
   t_value = stats.t.ppf((1 + confidence) / 2, n - 1)
   margin = t_value * (std / np.sqrt(n))
   return mean - margin, mean + margin

drug_ci = confidence_interval(drug_reduction)
placebo_ci = confidence_interval(placebo_reduction)

print("95% Confidence Interval for Drug Group:", drug_ci)
print("95% Confidence Interval for Placebo Group:", placebo_ci)
```

```
drug_ci = confidence_interval(drug_reduction)
placebo_ci = confidence_interval(placebo_reduction)

print("95% Confidence Interval for Drug Group:", drug_ci)
print("95% Confidence Interval for Placebo Group:", placebo_ci)
```

```
95% Confidence Interval for Drug Group: (np.float64(13.732507645657936), np.float64(15.467492354342063))
95% Confidence Interval for Placebo Group: (np.float64(1.8755601314517765), np.float64(2.684439868548223))
```

Scenario: You are a data scientist working for an e-commerce company. The marketing team has conducted an A/B test to evaluate the effectiveness of two different website designs (A and B) in terms of conversion rate. They randomly divided the website visitors into two groups, with one group experiencing design A and the other experiencing design B. After a week of data collection, you now have the conversion rate data for both groups. You want to determine whether there is a statistically significant difference in the mean conversion rates between the two website designs.

Question: "Based on the data collected from the A/B test, is there a statistically significant difference in the mean conversion rates between website design A and website design B?"

Solution:

import numpy as np

from scipy.stats import ttest_ind

design_A = np.array([3.2, 3.5, 3.1, 3.4, 3.6, 3.3, 3.2, 3.5, 3.4, 3.3])

design_B = np.array([3.8, 4.0, 3.9, 4.1, 3.7, 4.0, 3.9, 4.2, 3.8, 4.1])

t_stat, p_value = ttest_ind(design_A, design_B)

print("T-statistic:", t_stat)

print("P-value:", p_value)

```
import numpy as np
from scipy.stats import ttest_ind

design_A = np.array([3.2, 3.5, 3.1, 3.4, 3.6, 3.3, 3.2, 3.5, 3.4, 3.3])
design_B = np.array([3.8, 4.0, 3.9, 4.1, 3.7, 4.0, 3.9, 4.2, 3.8, 4.1])

t_stat, p_value = ttest_ind(design_A, design_B)

print("T-statistic:", t_stat)
print("P-value:", p_value)
```

```
T-statistic: -8.485281374238573
P-value: 1.04885469518974e-07
```