DSA0410 – Fundamentals of Data Science

Day - 4

Name : Abinesh H

Reg.no.: 192424387

16. 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?

```
#exp16
import pandas as pd
import re
from collections import Counter

# Sample customer reviews dataset
data = {
    "review": [
        "This product is very good and useful",
        "Good quality product with good performance",
        "Not satisfied with the product quality",
        "Excellent product and very good value",
        "Product quality is average"
    ]
}
df = pd.DataFrame(data)

# Combine all reviews into one text
text = " ".join(df["review"]).lower()

# Remove punctuation and split into words
words = re.findall(r'\b\w+\b', text)

# Calculate frequency distribution
word_freq = Counter(words)

# Display result
print(word_freq)
```

```
Counter({'product': 5, 'good': 4, 'quality': 3, 'is': 2, 'very': 2, 'and': 2,
```

17. 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.

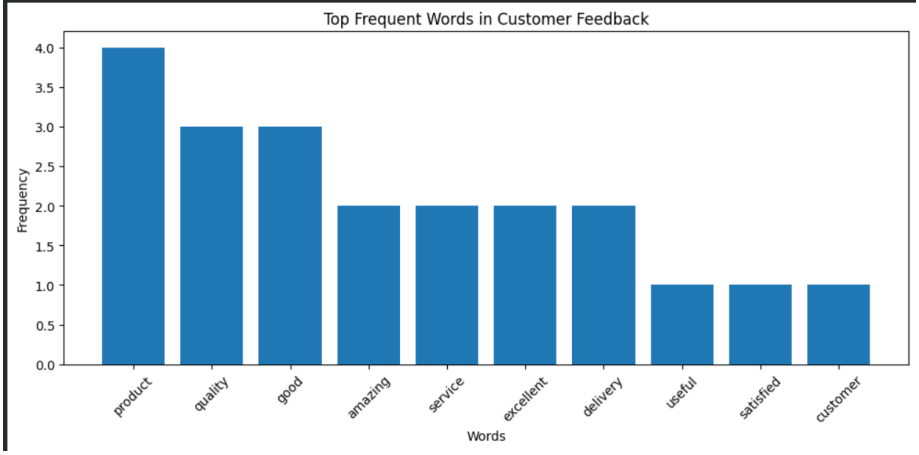
 Display the top N most frequent words and their corresponding frequencies, where N is provided as user input.

 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.

```python
#exp17
import pandas as pd
import re
import matplotlib.pyplot as plt
from collections import Counter
import nltk
from nltk.corpus import stopwords
# Download stopwords (run once)
nltk.download('stopwords')
# ------------------------------
# SAMPLE DATASET (IN CODE)
# ------------------------------
data = {
    "feedback": [
        "This product is amazing and very useful",
        "I am not satisfied with the customer service",
        "Excellent quality and fast delivery",
        "The product quality is good but delivery is slow",
        "Very poor experience with this product",
        "Amazing service and excellent support",
        "Not happy with the product performance",
        "Good value for money and good quality"
    ]
}
df = pd.DataFrame(data)
# Get English stop words
stop_words = set(stopwords.words("english"))
# Text preprocessing function
def preprocess_text(text):
    text = text.lower()                          # Convert to lowercase
    text = re.sub(r'[^\w\s]', '', text)          # Remove punctuation
    words = text.split()                         # Tokenize
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
Enter number of top frequent words: 10

Top 10 Most Frequent Words:

product : 4
quality : 3
good : 3
amazing : 2
service : 2
excellent : 2
delivery : 2
useful : 1
satisfied : 1
customer : 1
```



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

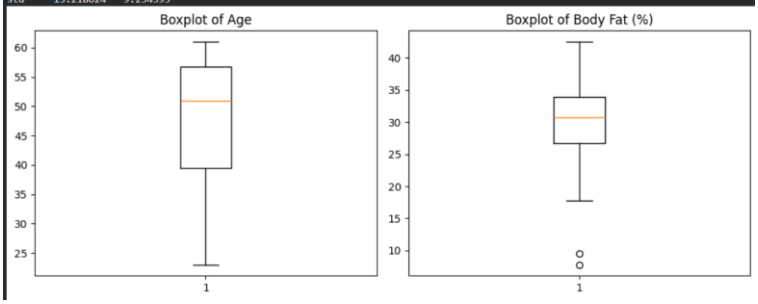
age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

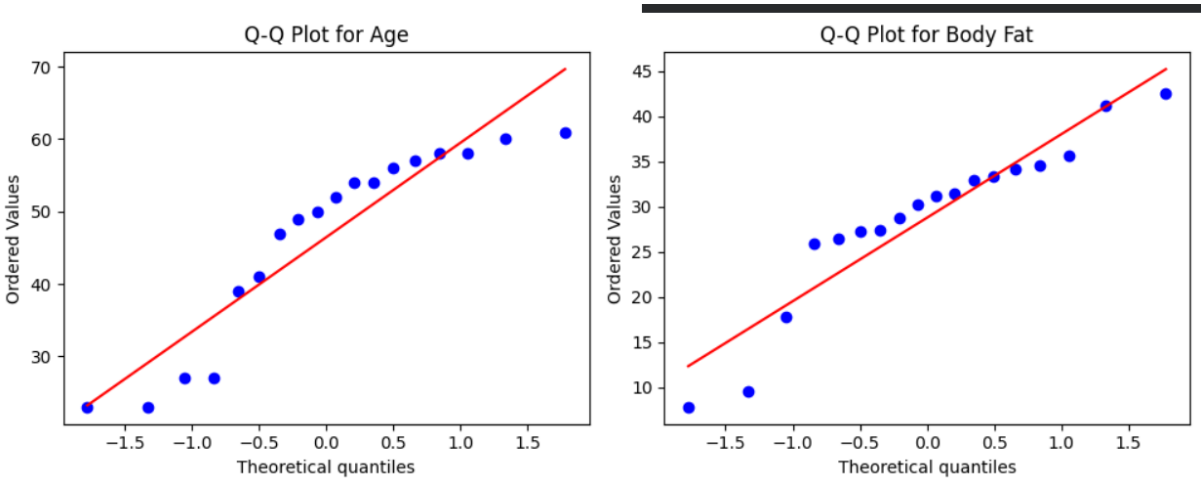
Question:

- 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

```python
#exp18
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
data = {
    "Age": [23, 23, 27, 27, 39, 41, 47, 49, 50,
            52, 54, 54, 56, 57, 58, 58, 60, 61],
    "BodyFat": [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(df)
stats_df = df.agg(['mean', 'median', 'std'])
print("\nStatistical Summary:\n")
print(stats_df)
plt.figure(figsize=(10,4))

plt.subplot(1,2,1)
plt.boxplot(df["Age"])
plt.title("Boxplot of Age")

plt.subplot(1,2,2)
plt.boxplot(df["BodyFat"])
plt.title("Boxplot of Body Fat (%)")

plt.tight_layout()
plt.show()
plt.figure(figsize=(6,5))
plt.scatter(df["Age"], df["BodyFat"])
plt.xlabel("Age")
```

```
    Age  BodyFat
0    23      9.5
1    23     26.5
2    27      7.8
3    27     17.8
4    39     31.4
5    41     25.9
6    47     27.4
7    49     27.2
8    50     31.2
9    52     34.6
10   54     42.5
11   54     28.8
12   56     33.4
13   57     30.2
14   58     34.1
15   58     32.9
16   60     41.2
17   61     35.7

Statistical Summary:

              Age    BodyFat
mean    46.444444  28.783333
median  51.000000  30.700000
std     13.218624   9.254395
```





19. 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?

```python
#exp19
import numpy as np
import pandas as pd
from scipy import stats
# Sample blood pressure reduction data (mmHg)

drug_group = np.array([
    12, 15, 14, 16, 13, 17, 18, 14, 15, 16,
    14, 15, 17, 16, 18, 19, 20, 14, 15, 16,
    13, 14, 15, 16, 17
])

placebo_group = np.array([
    2, 3, 1, 4, 2, 3, 5, 1, 2, 3,
    2, 3, 4, 2, 3, 4, 1, 2, 3, 2,
    1, 2, 3, 4, 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, df=n-1)
    margin_error = t_value * (std / np.sqrt(n))
    return mean - margin_error, mean + margin_error
drug_ci = confidence_interval(drug_group)
placebo_ci = confidence_interval(placebo_group)
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(14.75994531134246), np.float64(16.36005468865754))
95% Confidence Interval for Placebo Group: (np.float64(2.1128749019216277), np.float64(3.0071250980783724))
```

20. 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?"

```python
#exp20
import numpy as np
from scipy import stats
# Conversion rates (%) for two website designs
design_A = np.array([
    3.2, 3.5, 3.1, 3.4, 3.6, 3.3, 3.5, 3.2, 3.4, 3.3,
    3.6, 3.7, 3.4, 3.5, 3.3
])

design_B = np.array([
    3.8, 4.0, 3.9, 4.1, 3.7, 3.9, 4.2, 4.0, 3.8, 3.9,
    4.1, 4.3, 3.9, 4.0, 4.2
])
t_stat, p_value = stats.ttest_ind(design_A, design_B)

print("T-statistic:", t_stat)
print("P-value:", p_value)
alpha = 0.05

if p_value < alpha:
    print("Reject the null hypothesis.")
    print("There is a statistically significant difference between the two designs.")
else:
    print("Fail to reject the null hypothesis.")
    print("There is no statistically significant difference between the two designs.")
```

```
T-statistic: -9.520967883121864
P-value: 2.8124794963115707e-10
Reject the null hypothesis.
There is a statistically significant difference between the two designs.
```