

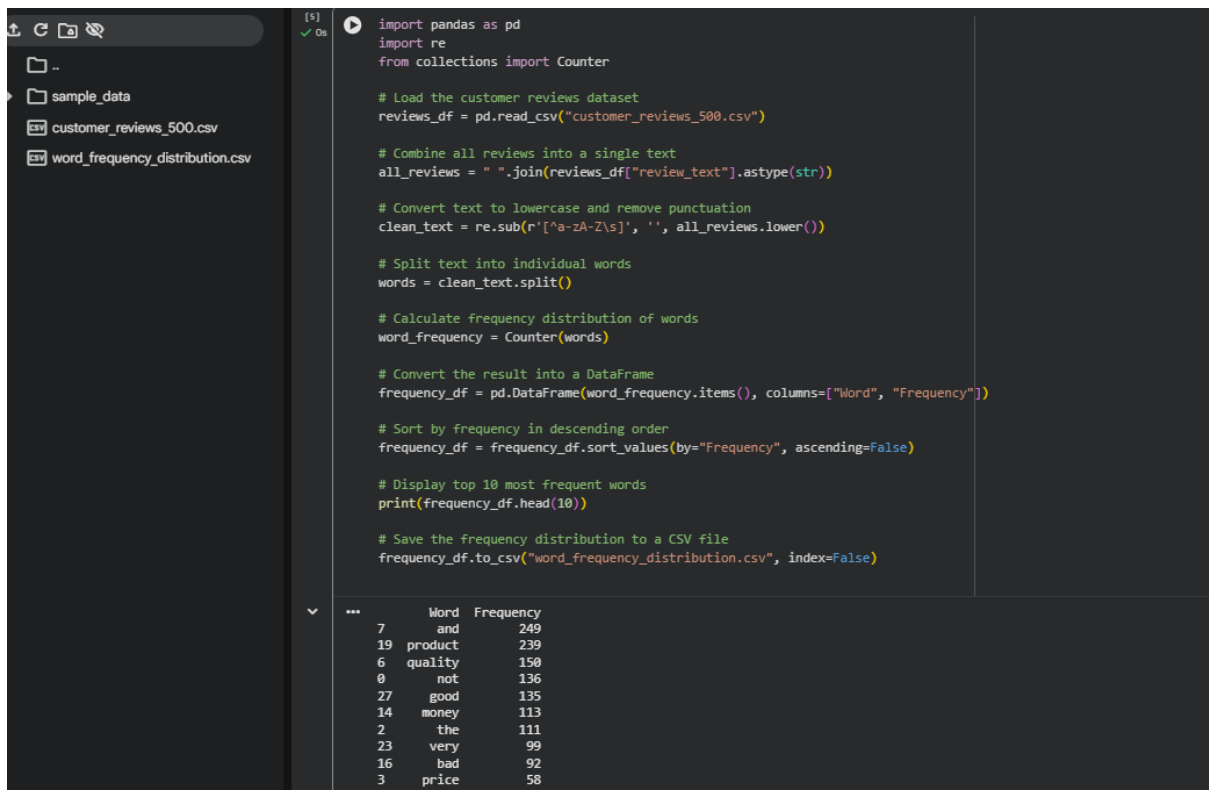
DAY 4 LAB EXPERIMENTS

NAME : LARAVIND

Reg No : 192424080

DATE : 19/12/2025

EXP_16 Write a Python program to compute the frequency distribution of words from a product's customer reviews dataset.



The screenshot shows a Jupyter Notebook interface. On the left, a file explorer displays a folder named 'sample_data' containing two CSV files: 'customer_reviews_500.csv' and 'word_frequency_distribution.csv'. The main area of the notebook contains a Python script that performs the following steps:

- Imports pandas as pd, re, and Counter from collections.
- Loads the customer reviews dataset into a DataFrame named 'reviews_df' using 'customer_reviews_500.csv'.
- Combines all reviews into a single text string using 'review_text'.
- Converts the text to lowercase and removes punctuation using a regular expression.
- Splits the text into individual words.
- Calculates the frequency distribution of words using Counter.
- Converts the result into a DataFrame with columns 'Word' and 'Frequency'.
- Sorts the DataFrame by frequency in descending order.
- Displays the top 10 most frequent words.
- Saves the frequency distribution to a CSV file named 'word_frequency_distribution.csv'.

Below the code, the output of the script is shown as a table:

	Word	Frequency
7	and	249
19	product	239
6	quality	159
0	not	136
27	good	135
14	money	113
2	the	111
23	very	99
16	bad	92
3	price	58

EXP_17 Develop a Python program to preprocess customer feedback from a CSV file, compute word frequencies, display the top N words, and visualize them using a bar chart.

```
Files
  sample_data
    customer_reviews_500.csv
    data.csv
    word_frequency_distribution.csv

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

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

stop_words = {
    "the", "and", "is", "in", "to", "of", "a", "for", "on", "with",
    "this", "that", "it", "as", "was", "are", "be", "by", "an"
}

text = " ".join(df["feedback"].astype(str))

text = text.lower()

text = re.sub(r"[^a-z\s]", "", text)
words = text.split()
filtered_words = [word for word in words if word not in stop_words]

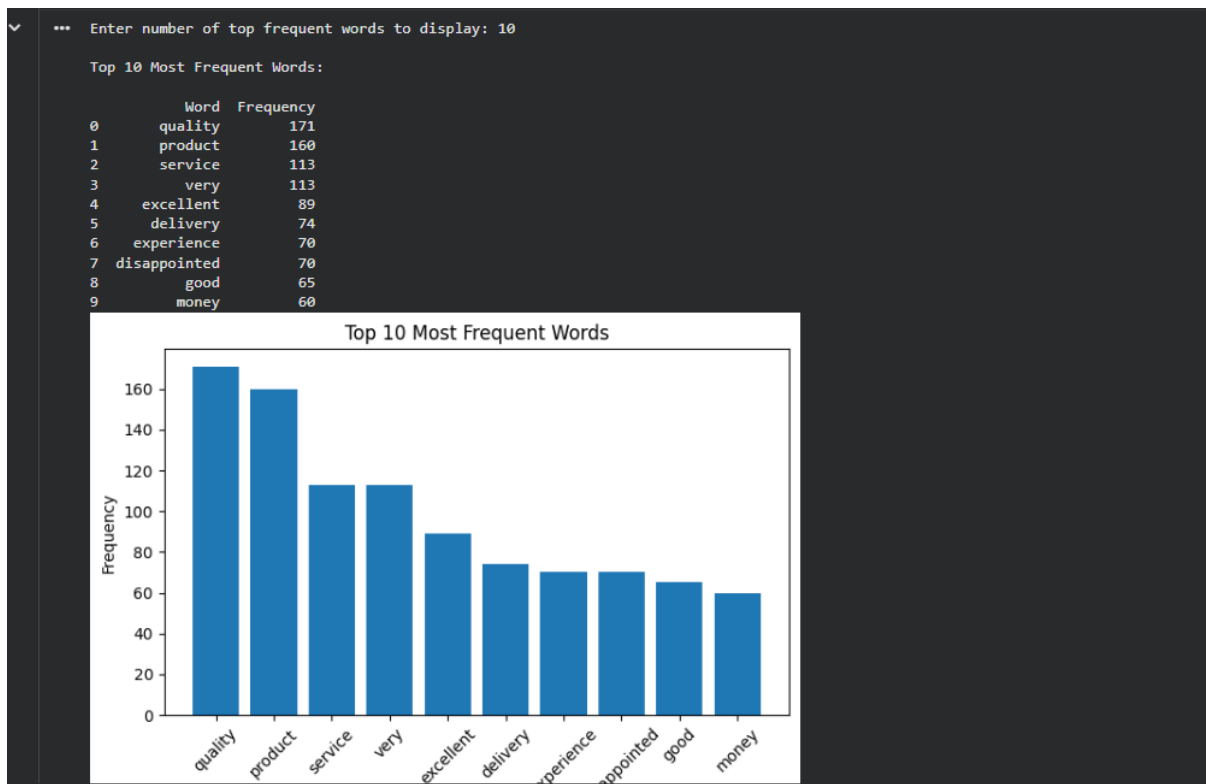
word_freq = Counter(filtered_words)

N = int(input("Enter number of top frequent words to display: "))

top_words = word_freq.most_common(N)

freq_df = pd.DataFrame(top_words, columns=["Word", "Frequency"])
print("\nTop", N, "Most Frequent Words:\n")
print(freq_df)

plt.figure()
plt.bar(freq_df["Word"], freq_df["Frequency"])
plt.xlabel("Words")
plt.ylabel("Frequency")
plt.title("Top {} Most Frequent Words".format(N))
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



EXP_18 Use Pandas to calculate mean, median, and standard deviation of age and body fat data, and visualize them using boxplots, scatter plots, and Q-Q plots.

```
Files [7]
from scipy import stats

df = pd.read_csv("age_bodyfat_18.csv")

print("Statistical Measures:\n")

print("Age:")
print("Mean:", df["Age"].mean())
print("Median:", df["Age"].median())
print("Standard Deviation:", df["Age"].std(), "\n")

print("Body Fat Percentage:")
print("Mean:", df["BodyFat"].mean())
print("Median:", df["BodyFat"].median())
print("Standard Deviation:", df["BodyFat"].std())

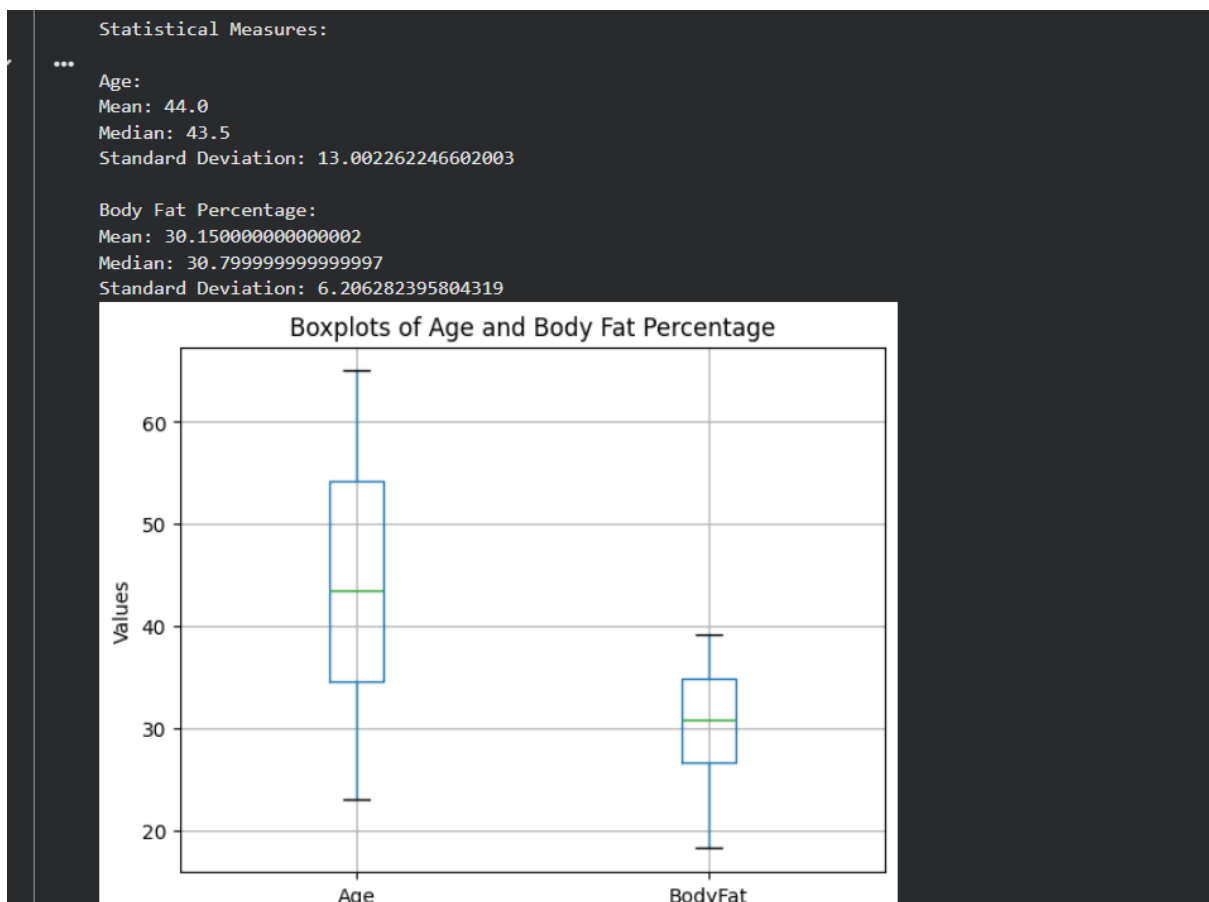
plt.figure()
df.boxplot(column=["Age", "BodyFat"])
plt.title("Boxplots of Age and Body Fat Percentage")
plt.ylabel("Values")
plt.show()

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

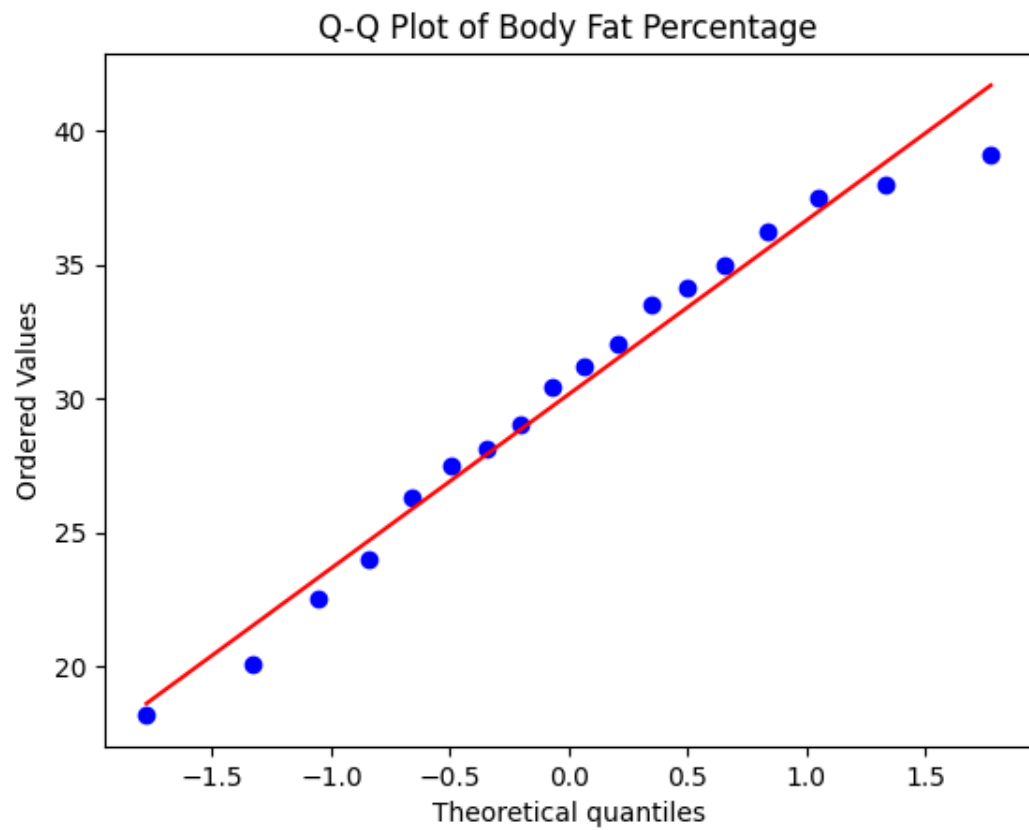
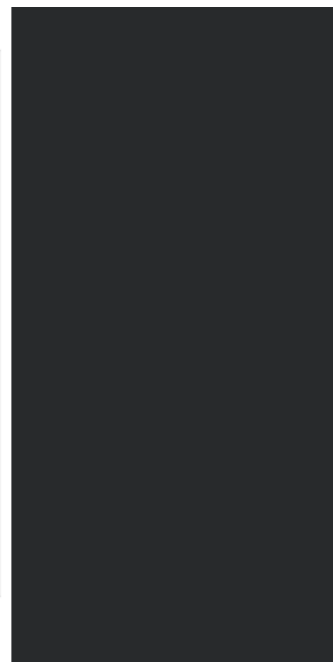
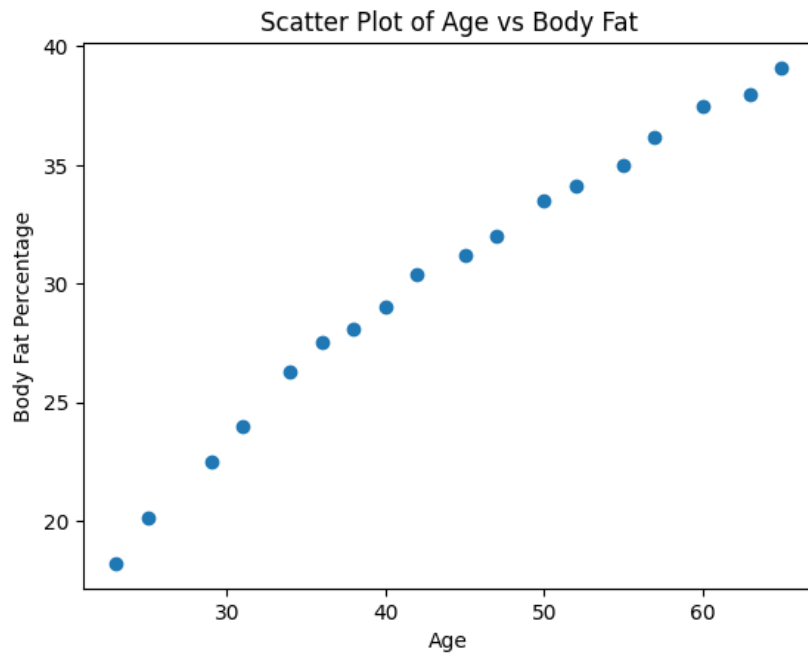
plt.figure()
stats.probplot(df["BodyFat"], dist="norm", plot=plt)
plt.title("Q-Q Plot of Body Fat Percentage")
plt.show()
```

Statistical Measures:

Age:
Mean: 44.0



...



EXP_19 Calculate the 95% confidence intervals for the mean blood pressure reduction in both drug and placebo groups from a clinical trial.

```
import pandas as pd
import numpy as np
from scipy import stats

df = pd.read_csv("blood_pressure_reduction_50.csv")

drug_group = df[df["Group"] == "Drug"]["BP_Reduction"]
placebo_group = df[df["Group"] == "Placebo"]["BP_Reduction"]

def confidence_interval(data, confidence=0.95):
    mean = np.mean(data)
    std_err = stats.sem(data)
    margin = std_err * stats.t.ppf((1 + confidence) / 2, len(data) - 1)
    return mean - margin, mean + margin

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(9.76658498706352), np.float64(12.925358543671751))
95% Confidence Interval for Placebo Group: (np.float64(1.991428176754361), np.float64(4.283933312120505))
```

EXP_20 Analyze A/B test conversion rate data to determine whether there is a statistically significant difference between website designs A and B.

```
import pandas as pd
from scipy import stats

df = pd.read_csv("ab_test_conversion_rates.csv")
design_A = df[df["Design"] == "A"]["Conversion_Rate"]
design_B = df[df["Design"] == "B"]["Conversion_Rate"]

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("Result: There IS a statistically significant difference between Design A and Design B.")
else:
    print("Result: There is NO statistically significant difference between Design A and Design B.")
```

```
T-statistic: -8.551459943248732
P-value: 3.292961428703664e-15
Result: There IS a statistically significant difference between Design A and Design B.
```