

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 with a dark theme. On the left, there's a file tree with 'sample_data' containing 'customer_reviews_500.csv' and 'word_frequency_distribution.csv'. The main area contains the following Python code:

```
[S] ✓ 0s
import pandas as pd
import re
from collections import Counter

# Load the customer reviews dataset
reviews_df = pd.read_csv("customer_reviews_500.csv")

# Combine all reviews into a single text
all_reviews = " ".join(reviews_df["review_text"].astype(str))

# Convert text to lowercase and remove punctuation
clean_text = re.sub(r'[^\w\s]', '', all_reviews.lower())

# Split text into individual words
words = clean_text.split()

# Calculate frequency distribution of words
word_frequency = Counter(words)

# Convert the result into a DataFrame
frequency_df = pd.DataFrame(word_frequency.items(), columns=["Word", "Frequency"])

# Sort by frequency in descending order
frequency_df = frequency_df.sort_values(by="Frequency", ascending=False)

# Display top 10 most frequent words
print(frequency_df.head(10))

# Save the frequency distribution to a CSV file
frequency_df.to_csv("word_frequency_distribution.csv", index=False)
```

Below the code, the output is displayed in a table:

	Word	Frequency
7	and	249
19	product	239
6	quality	150
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.

The screenshot shows a Jupyter Notebook interface with two main sections: a file browser on the left and a code editor/terminal on the right.

File Browser: Shows files in the current directory: .., sample_data, customer_reviews_500.csv, data.csv, and word_frequency_distribution.csv.

Code Editor: Displays the following Python script:

```
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("Top {} Most Frequent Words:\n".format(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()
```

Output Terminal: Shows the user input and the resulting DataFrame:

```
... Enter number of top frequent words to display: 10
```

```
Top 10 Most Frequent Words:
```

	Word	Frequency
0	quality	171
1	product	160
2	service	113
3	very	113
4	excellent	89
5	delivery	74
6	experience	70
7	disappointed	70
8	good	65
9	money	60

Figure: A bar chart titled "Top 10 Most Frequent Words" showing the frequency of words. The x-axis lists the words: quality, product, service, very, excellent, delivery, experience, disappointed, good, and money. The y-axis represents Frequency, ranging from 0 to 160. The bars are blue.

Word	Frequency
quality	171
product	160
service	113
very	113
excellent	89
delivery	74
experience	70
disappointed	70
good	65
money	60

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.

The screenshot shows a Jupyter Notebook interface with two main sections: a file browser on the left and a code editor/terminal on the right.

File Browser: Shows a list of files in the current directory, including sample_data, ab_test_conversion_rates.csv, age_bodyfat_18.csv, blood_pressure_reduction_50.csv, customer_reviews_500.csv, data.csv, and word_frequency_distribution.csv.

Code Editor/Terminal: Displays Python code using Pandas and SciPy to calculate statistical measures and generate plots.

```
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()
```

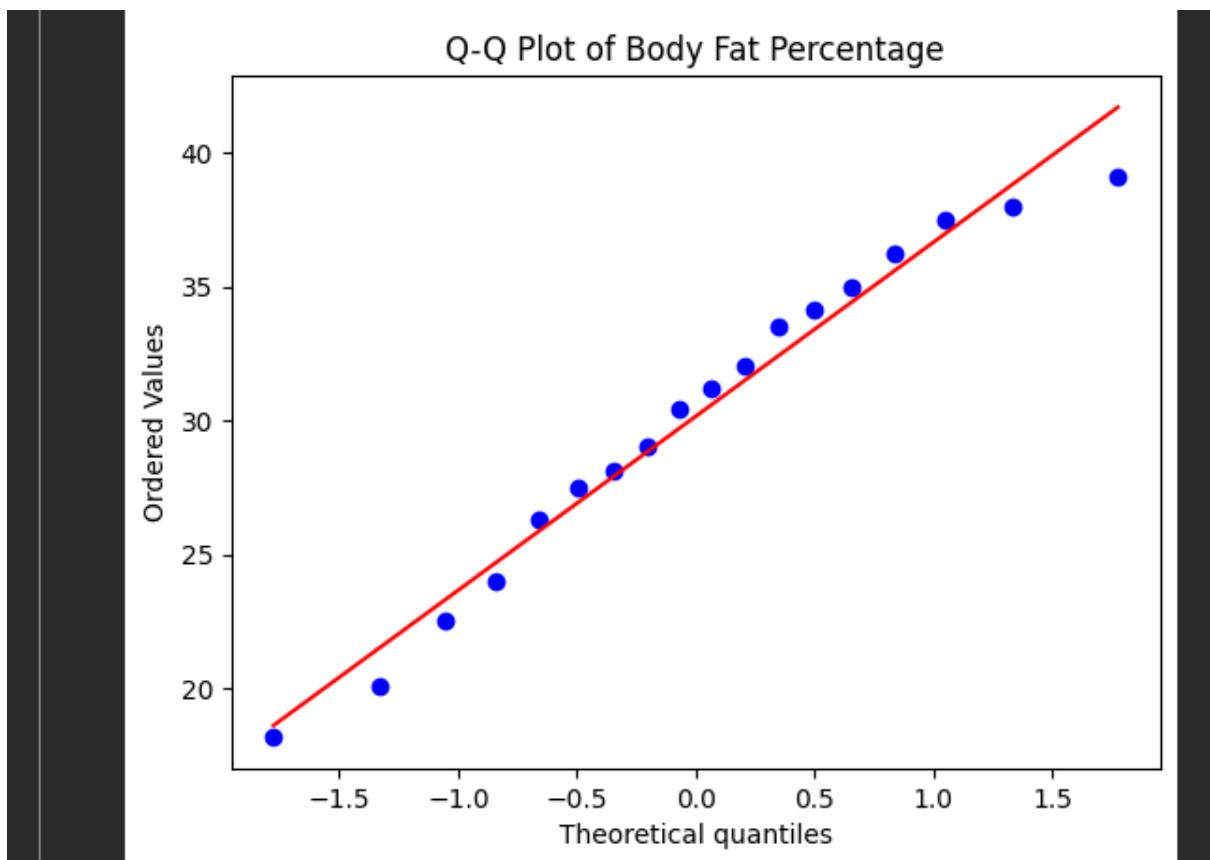
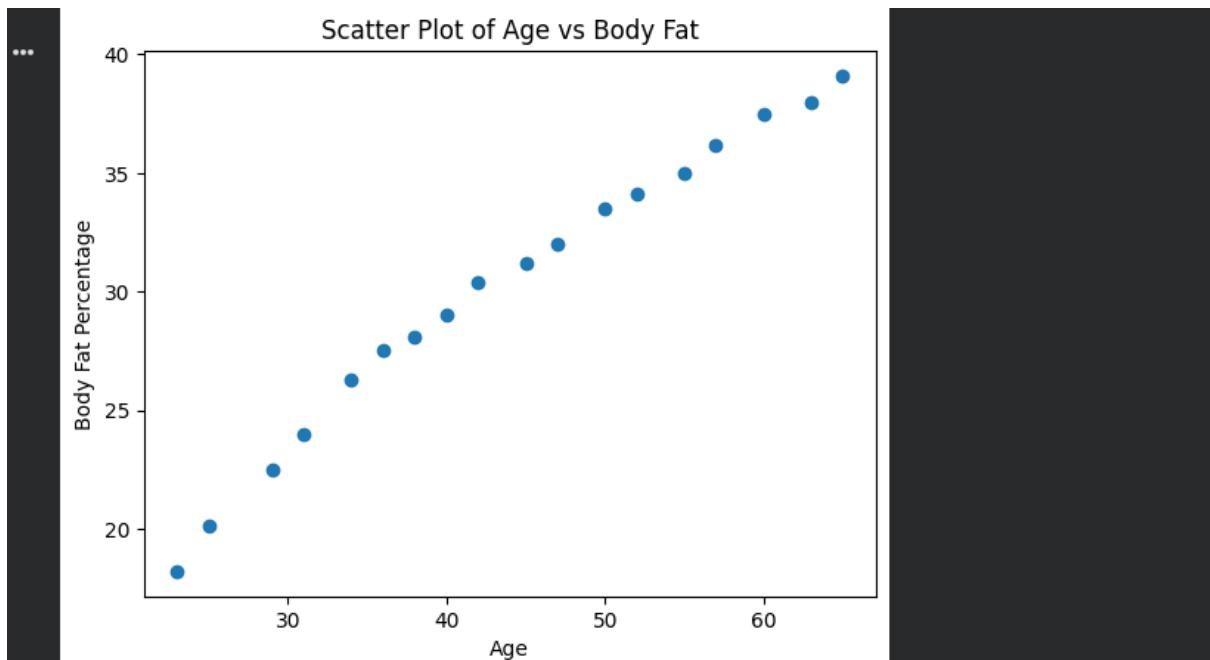
Output: The terminal shows the calculated statistical measures for Age and Body Fat Percentage.

```
Statistical Measures:
...
Age:
Mean: 44.0
Median: 43.5
Standard Deviation: 13.002262246602003

Body Fat Percentage:
Mean: 30.150000000000002
Median: 30.799999999999997
Standard Deviation: 6.206282395804319
```

Plots: Two boxplots are generated: one for Age and one for Body Fat Percentage. The Age boxplot has a median of approximately 44, an IQR from about 35 to 55, and whiskers extending from 23 to 68. The Body Fat Percentage boxplot has a median of approximately 31, an IQR from about 27 to 35, and whiskers extending from 18 to 40.

The figure contains two side-by-side boxplots. The left boxplot is titled "Boxplots of Age and Body Fat Percentage". The x-axis is labeled "Age" and the y-axis is labeled "Values". The boxplot shows a median of approximately 44, an IQR from about 35 to 55, and whiskers extending from 23 to 68. The right boxplot shows a median of approximately 31, an IQR from about 27 to 35, and whiskers extending from 18 to 40. The y-axis for both is labeled "Values" and ranges from 20 to 60.



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

The screenshot shows a Jupyter Notebook interface. On the left, there's a file tree with files like sample_data, age_bodyfat_18.csv, blood_pressure_reduction_50.csv, customer_reviews_500.csv, data.csv, and word_frequency_distribution.csv. The main area has a code cell numbered [11] with the following Python code:

```
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)
```

Output from the cell shows the confidence intervals for both groups:

```
... 95% Confidence Interval for Drug Group: (np.float64(9.76658498706352), np.float64(12.925350543671751))
95% Confidence Interval for Placebo Group: (np.float64(1.991428176754361), np.float64(4.283933121289505))
```

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

The screenshot shows a Jupyter Notebook interface. On the left, there's a file tree with files like sample_data, ab_test_conversion_rates.csv, age_bodyfat_18.csv, blood_pressure_reduction_50.csv, customer_reviews_500.csv, data.csv, and word_frequency_distribution.csv. The main area has a code cell numbered [12] with the following Python code:

```
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.")
```

Output from the cell shows the results of the t-test:

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