TASK-1
Task 1: Word Frequency from Text File
❖ Scenario:
You are analyzing log files for keyword frequency.
❖ Task:
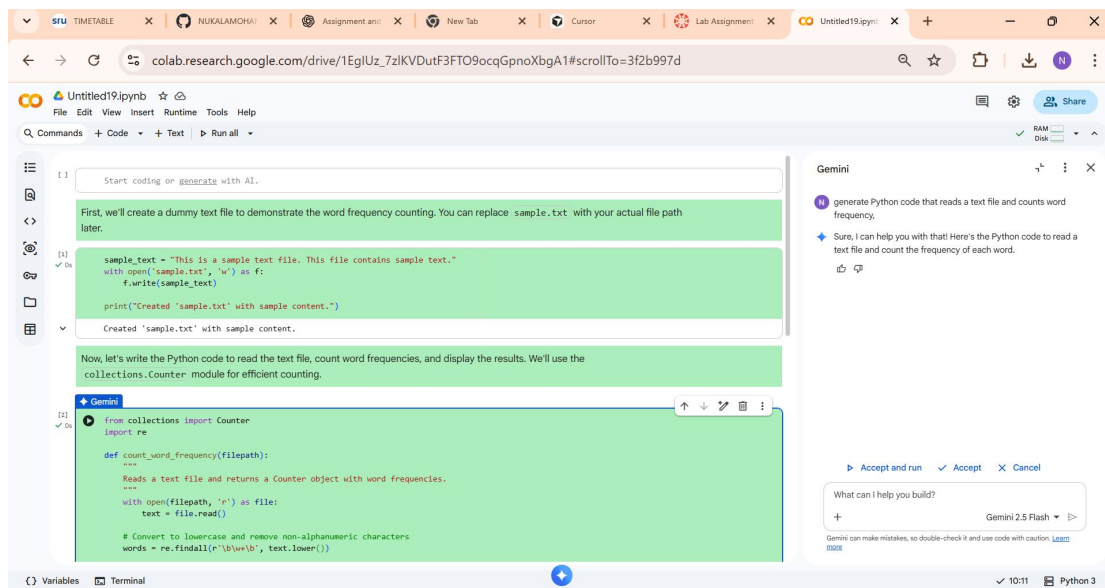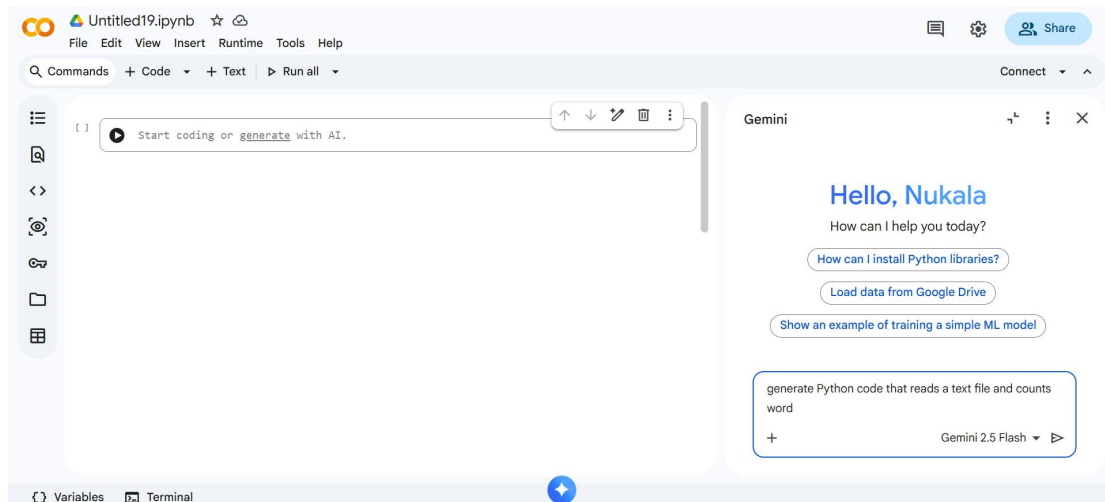Use Gemini to generate Python code that reads a text file and counts word
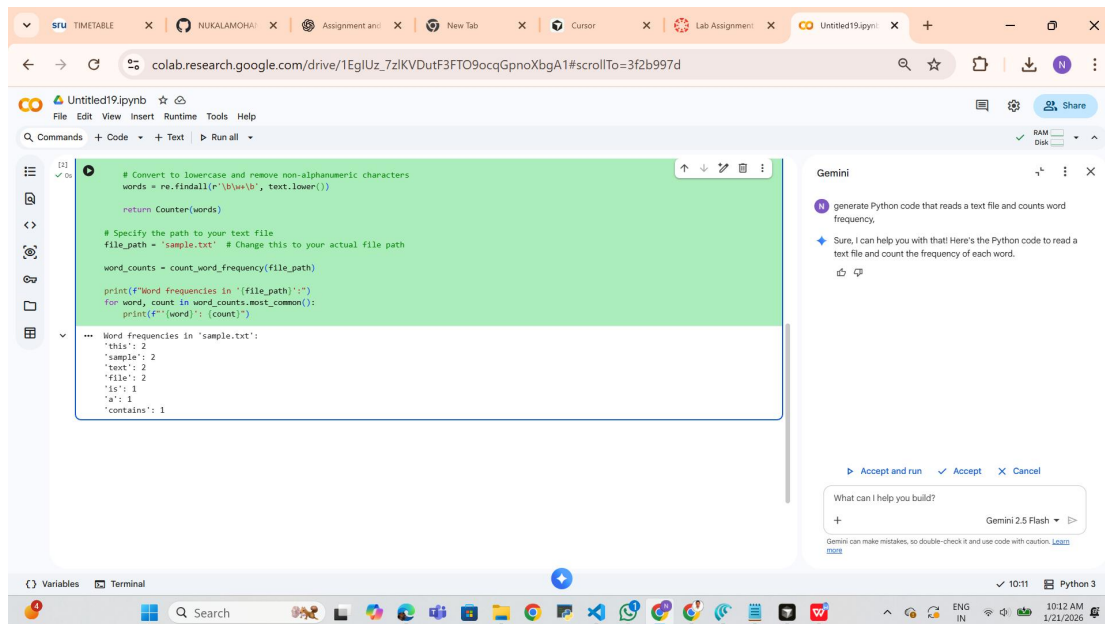frequency, then explains the code.
❖ Expected Output:
➢ Working code
➢ Explanation
➢ Screenshot

The code defines a function to read a text file, sample.txt, for processing.
It converts the entire text to lowercase and extracts all alphanumeric words using regular expressions.
The collections.Counter module is then used to efficiently count the frequency of each extracted word.
The function is called with the sample.txt file, which was pre-populated with example text.
Finally, the code iterates through the word counts and prints each word along with its frequency.

TASK-2
❖ Scenario:
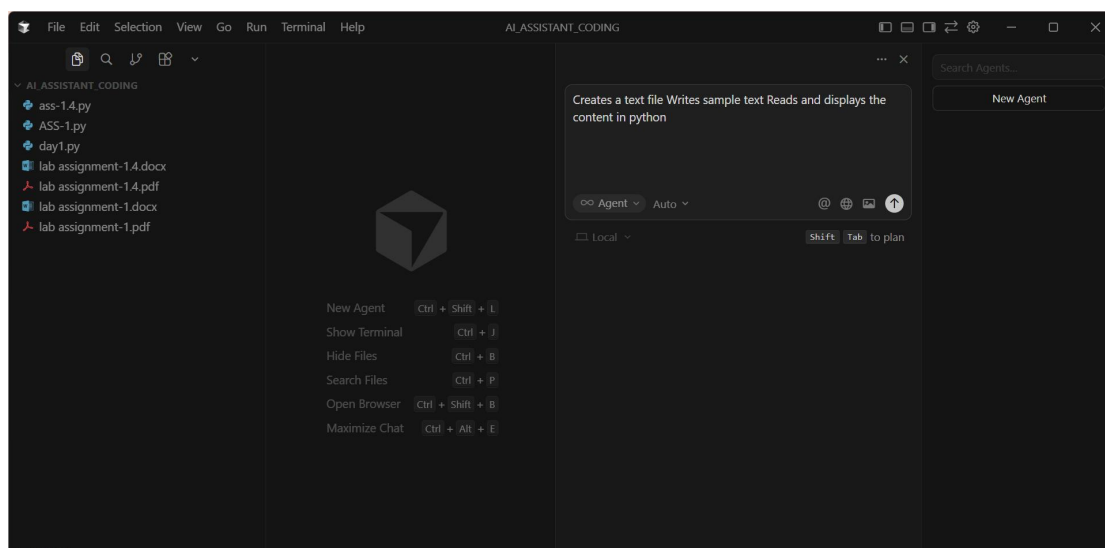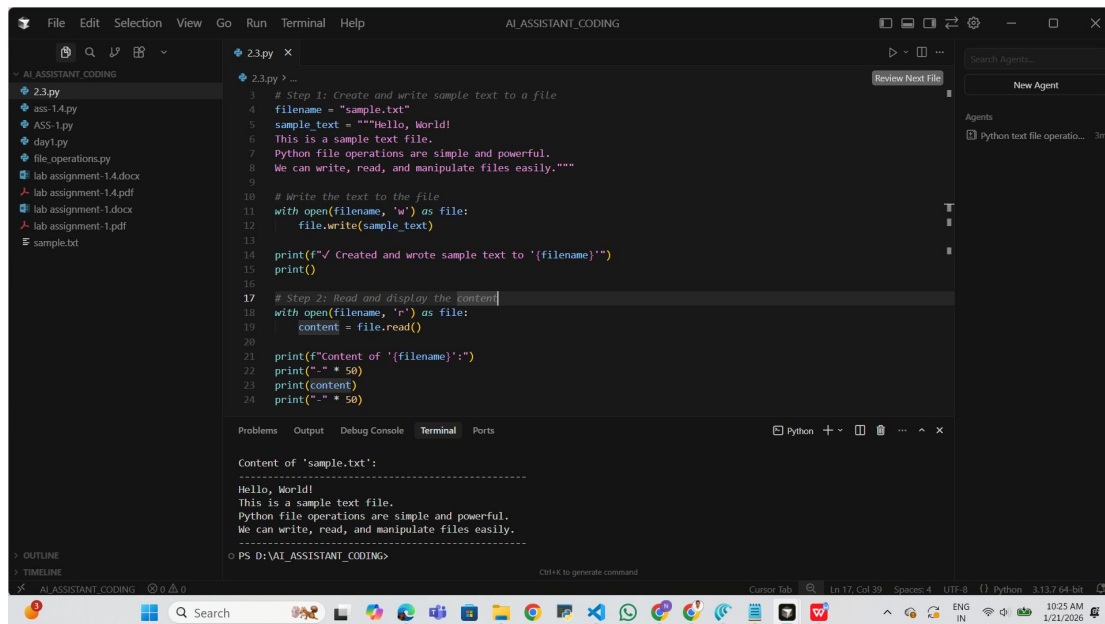You are automating basic file operations.
❖ Task:
Use Cursor AI to generate a program that:
➢ Creates a text file
➢ Writes sample text
➢ Reads and displays the content
❖ Expected Output:
➢ Functional code
➢ Cursor AI screenshots

Task 3: CSV Data Analysis
❖ Scenario:
You are processing structured data from a CSV file.
❖ Task:
Use Gemini in Colab to read a CSV file and calculate mean, min, and max.
❖ Expected Output:
➢ Correct output
➢ Screenshot

Share

RAM
Disk

Gemini

✦ Certainly! Here's a 5-line explanation of the code:

1. The code defines a function to read a text file, `sample.txt`, for processing.
2. It converts the entire text to lowercase and extracts all alphanumeric words using regular expressions.
3. The `collections.Counter` module is then used to efficiently count the frequency of each extracted word.
4. The function is called with the `sample.txt` file, which was pre-populated with example text.
5. Finally, the code iterates through the word counts and prints each word along with its frequency.

👍 👎

Show me the Python code to count word frequency

Run the word frequency counting code

How can I sort the word counts by frequency

read a CSV file and calculate mean, min, and max.

\+   Gemini 2.5 Flash ▾ ▷

Gemini can make mistakes, so double-check it and use code with caution. Learn more

✓ 10:11   🖫 Python 3

ENG
IN   10:34 AM
                      1/21/2026

sru TIMETAB × | NUKALA × | Assignm × | New Tab × | Cursor × | Lab Assi × | CO Untitled × | New Tab × | NZ CSV file × | +

colab.research.google.com/drive/1EgIUz_7zlKVDutF3FTO9ocqGpnoXbgA1#scrollTo=fcc15417

Untitled19.ipynb
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

Start coding or generate with AI.

First, let's create a sample CSV file named `sample_data.csv` that we can use for demonstration. It will contain some numerical data.

```python
import pandas as pd

# Create a sample DataFrame
data = {'Category': ['A', 'B', 'A', 'C', 'B', 'A', 'C'],
        'Value': [10, 25, 15, 30, 20, 12, 35],
        'Quantity': [1, 5, 2, 8, 3, 2, 7]}
df_sample = pd.DataFrame(data)

# Save the DataFrame to a CSV file
df_sample.to_csv('sample_data.csv', index=False)

print("Created 'sample_data.csv' with sample content.")
print(df_sample)
```

```
Created 'sample_data.csv' with sample content.
  Category  Value  Quantity
0        A     10         1
1        B     25         5
2        A     15         2
3        C     30         8
4        B     20         3
5        A     12         2
6        C     35         7
```

Now, let's read the `sample_data.csv` file into a pandas DataFrame and calculate the mean, min, and max for a numerical column, for example, the 'Value' column.

Gemini

N read a CSV file and calculate mean, min, and max.

Okay, I can help with that! Here's the Python code to read a CSV file, then calculate and display the mean, minimum, and maximum values for a numerical column. I'll include a step to create a sample CSV first.

Accept and run   ✓ Accept   ✕ Cancel

What can I help you build?

Gemini 2.5 Flash ▾

Gemini can make mistakes, so double-check it and use code with caution. Learn more

---

sru TIMETAB × | NUKALA × | Assignm × | New Tab × | Cursor × | Lab Assi × | CO Untitled × | New Tab × | NZ CSV file × | +

colab.research.google.com/drive/1EgIUz_7zlKVDutF3FTO9ocqGpnoXbgA1#scrollTo=fcc15417

Untitled19.ipynb
File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all

Files

Now, let's read the `sample_data.csv` file into a pandas DataFrame and calculate the mean, min, and max for a numerical column, for example, the 'Value' column.

Gemini

```python
import pandas as pd

# Specify the path to your CSV file
csv_file_path = 'sample_data.csv'  # Change this to your actual CSV file path

try:
    # Read the CSV file into a pandas DataFrame
    df = pd.read_csv(csv_file_path)

    print(f"Successfully loaded '{csv_file_path}'. Here's the first few rows:")
    display(df.head())

    # Select a numerical column to calculate statistics (e.g., 'Value')
    # You can change 'Value' to any numerical column in your CSV.
    numerical_column = 'Value'

    if numerical_column in df.columns and pd.api.types.is_numeric_dtype(df[numerical_column]):
        # Calculate mean, min, and max
        mean_value = df[numerical_column].mean()
        min_value = df[numerical_column].min()
        max_value = df[numerical_column].max()

        print(f"\nStatistics for column '{numerical_column}':")
        print(f"Mean: {mean_value:.2f}")
        print(f"Min: {min_value:.2f}")
        print(f"Max: {max_value:.2f}")
    else:
        print(f"\nError: Column '{numerical_column}' not found or is not numeric in the CSV file.")
        print("Available columns: ", df.columns.tolist())
```

N read a CSV file and calculate mean, min, and max.

Okay, I can help with that! Here's the Python code to read a CSV file, then calculate and display the mean, minimum, and maximum values for a numerical column. I'll include a step to create a sample CSV first.

Accept and run   ✓ Accept   ✕ Cancel

What can I help you build?

Gemini 2.5 Flash ▾

Gemini can make mistakes, so double-check it and use code with caution. Learn more

Task 4: Sorting Lists – Manual vs Built-in

❖ Scenario:

You are reviewing algorithm choices for efficiency.

❖ Task:

Use Gemini to generate:

➢ Bubble sort

➢ Python's built-in sort()

➢ Compare both implementations.

❖ Expected Output:

➢ Two versions of code

➢ Short comparison

🔍 ☆   ⬚ | ⬇ N ⋮

💬 ⚙ 👥 Share

✓ RAM ▭
Disk ▭ ▼ ^

🗑 ⋮

Gemini ⌐ᴸ ⋮ ✕

N read a CSV file and calculate mean, min, and max.

✦ Okay, I can help with that! Here's the Python code to read a CSV
file, then calculate and display the mean, minimum, and
maximum values for a numerical column. I'll include a step to
create a sample CSV first.

👍 👎

▷ Accept and run   ✓ Accept   ✕ Cancel

give a python code for bubble sort

+   Gemini 2.5 Flash ▼ ▷

Gemini can make mistakes, so double-check it and use code with caution. <u>Learn
more</u>

✓ 10:35   🖥 Python 3

ENG   10:38 AM

Short comparision

Bubble Sort: It's a simple sorting algorithm, but generally inefficient for large datasets (O(n^2) complexity). It sorts by repeatedly stepping through the list, comparing adjacent elements and swapping them.

Python's list.sort() and sorted(): These are highly optimized and implemented in C, using Timsort (a hybrid stable sorting algorithm with O(n log n) average and worst-case complexity). They are significantly faster and more memory-efficient for most use cases. list.sort() sorts in-place, while sorted() returns a new sorted list.