

```
In [1]: !pip install numpy pandas matplotlib psutil
```

```
Requirement already satisfied: numpy in /opt/conda/lib/python3.10/site-packages (1.22.4)
Requirement already satisfied: pandas in /opt/conda/lib/python3.10/site-packages (1.4.3)
Requirement already satisfied: matplotlib in /opt/conda/lib/python3.10/site-packages (3.5.2)
Requirement already satisfied: psutil in /opt/conda/lib/python3.10/site-packages (5.9.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /opt/conda/lib/python3.10/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.10/site-packages (from pandas) (2022.1)
Requirement already satisfied: cyclor>=0.10 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (1.4.3)
Requirement already satisfied: fonttools>=4.22.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (4.34.4)
Requirement already satisfied: pyparsing>=2.2.1 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (3.0.9)
Requirement already satisfied: pillow>=6.2.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (9.2.0)
Requirement already satisfied: packaging>=20.0 in /opt/conda/lib/python3.10/site-packages (from matplotlib) (21.3)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.10/site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

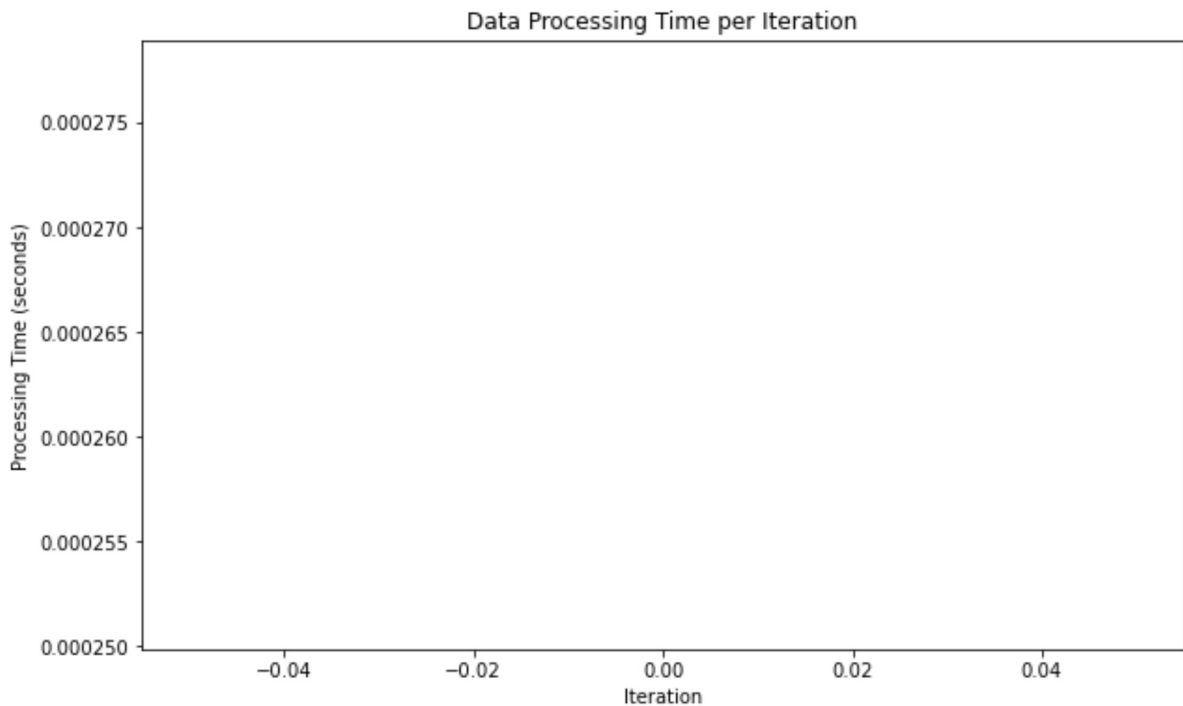
```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import psutil
import time
```

```
In [3]: def generate_sensor_data(num_sensors, num_readings):
        return np.random.rand(num_sensors, num_readings)
```

```
In [4]: def process_data(data):
        return np.mean(data, axis=1)
```

```
In [5]: num_sensors = 1000
num_readings = 100
num_iterations = 50
processing_times = []
for _ in range(num_iterations):
    data = generate_sensor_data(num_sensors, num_readings)
    start_time = time.time()
    processed_data = process_data(data)
    end_time = time.time()
    processing_times.append(end_time - start_time)
```

```
In [6]: plt.figure(figsize=(10, 6))
plt.title('Data Processing Time per Iteration')
plt.xlabel('Iteration')
plt.ylabel('Processing Time (seconds)')
plt.plot(processing_times)
plt.show()
```

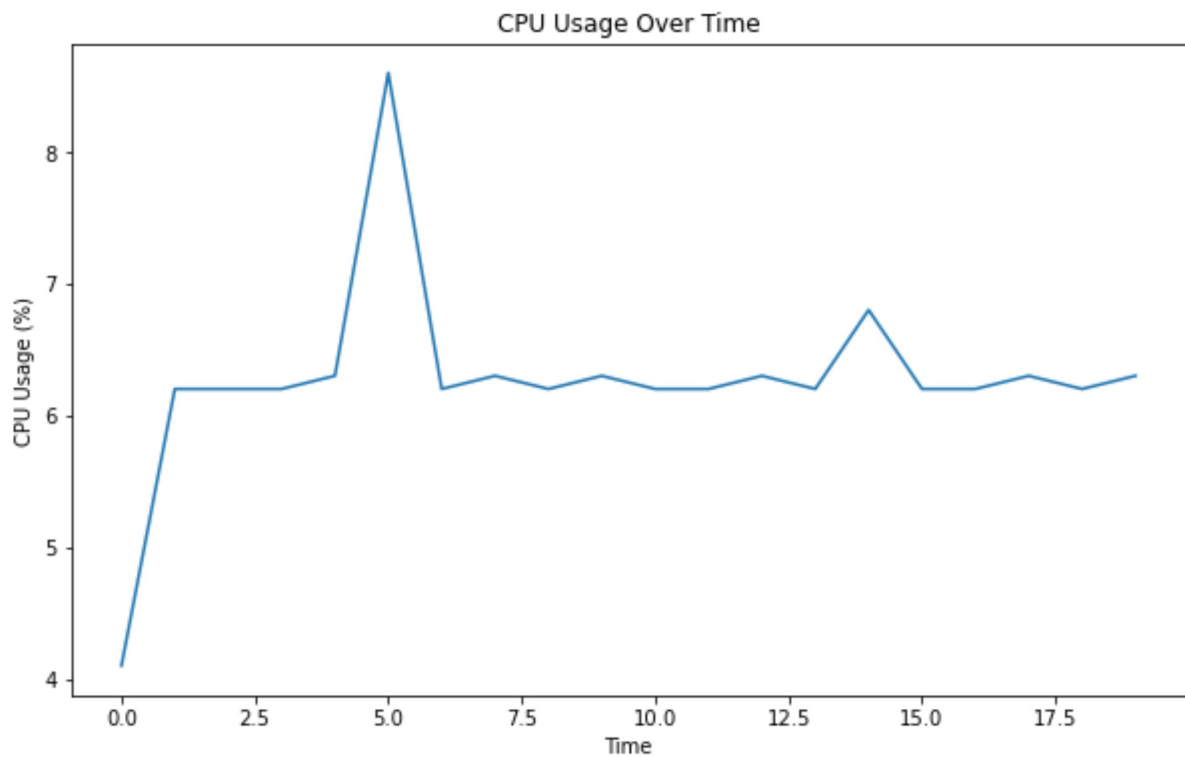


```
In [7]: def simulate_cpu_usage(duration):
    start_time = time.time()
    while time.time() - start_time < duration:
        pass

    def simulate_memory_usage(size_mb):
        return ' ' * (size_mb * 1024 * 1024)
```

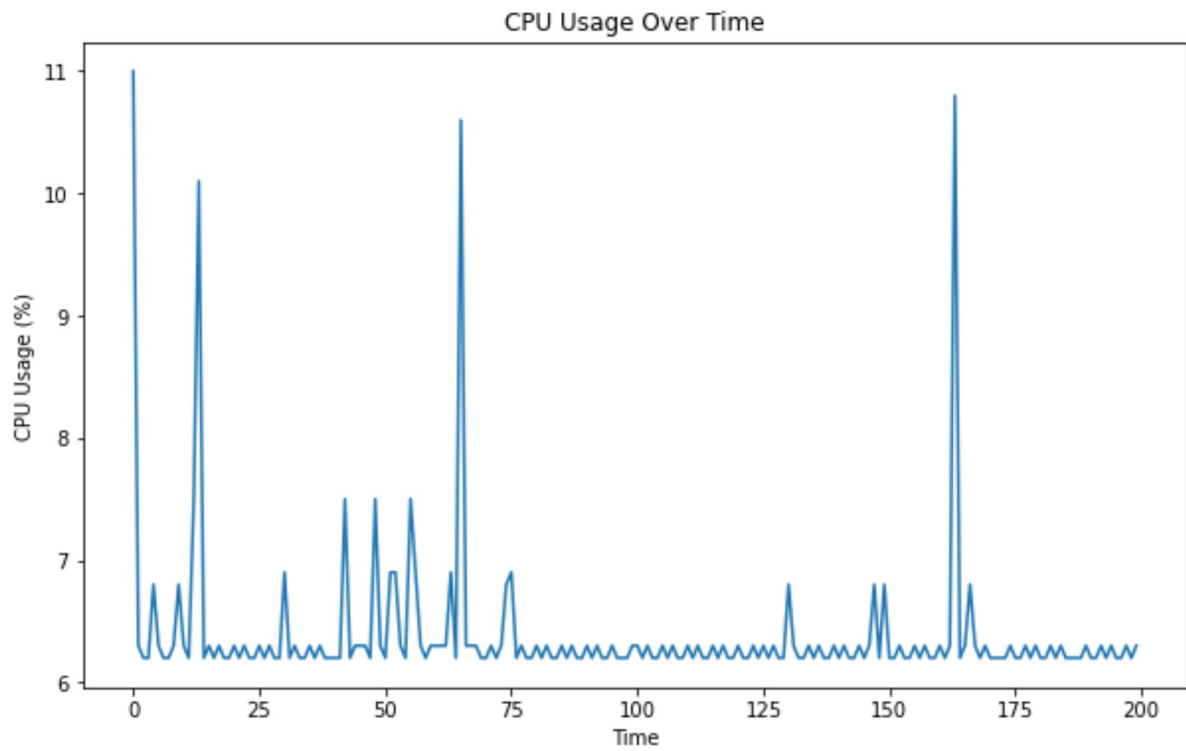
```
In [8]: cpu_usage = []
for _ in range(20):
    simulate_cpu_usage(0.1)
    cpu_usage.append(psutil.cpu_percent())

plt.figure(figsize=(10, 6))
plt.plot(cpu_usage)
plt.title('CPU Usage Over Time')
plt.xlabel('Time')
plt.ylabel('CPU Usage (%)')
plt.show()
```

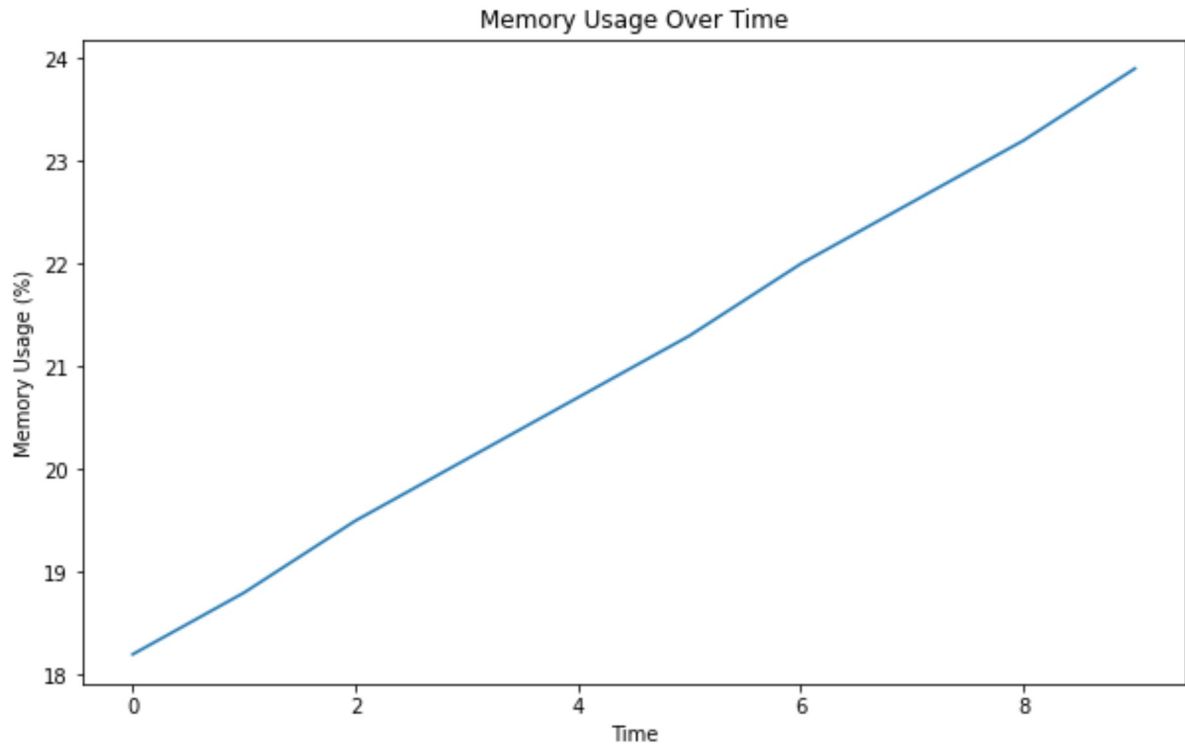


```
In [9]: cpu_usage = []
        for _ in range(200):
            simulate_cpu_usage(0.1)
            cpu_usage.append(psutil.cpu_percent())

        plt.figure(figsize=(10, 6))
        plt.plot(cpu_usage)
        plt.title('CPU Usage Over Time')
        plt.xlabel('Time')
        plt.ylabel('CPU Usage (%)')
        plt.show()
```



```
In [10]: memory_usage = []  
for i in range(10):  
    _ = simulate_memory_usage(100 * i) # Allocate memory in 100MB increments  
    memory_usage.append(psutil.virtual_memory().percent)  
plt.figure(figsize=(10, 6))  
plt.plot(memory_usage)  
plt.title('Memory Usage Over Time')  
plt.xlabel('Time')  
plt.ylabel('Memory Usage (%)')  
plt.show()
```



In []:

In []:

In []: