Dimaruct, Jon Paolo
CS2C-BSCS

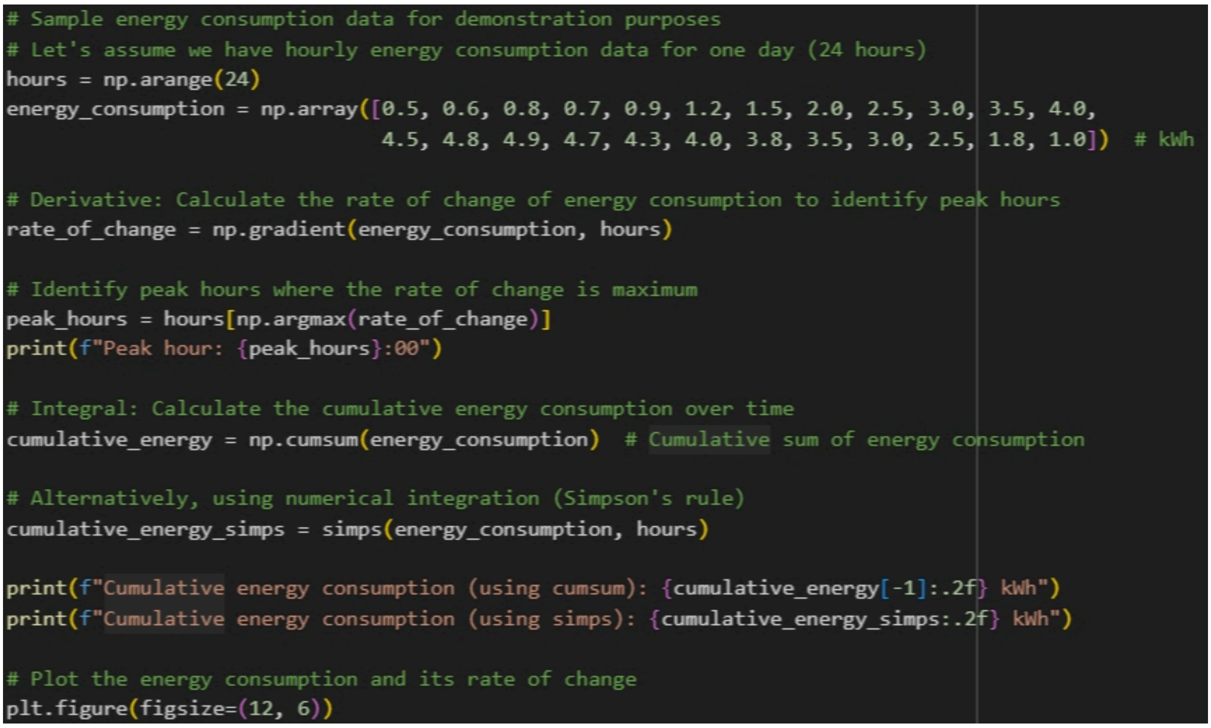
# Calculus PIT

**Data Collection**: Data collection in a smart home, specifically focusing on power meters, involves gathering real-time information about electricity usage for various appliances and devices. Smart plugs or dedicated energy monitoring devices equipped with power meters track the wattage consumed by each appliance at any given moment.

**Data Processing:** In the process of preparing energy consumption data for analysis, data cleaning plays a vital role in ensuring accuracy and reliability. A crucial step in this process involves the removal of outliers or erroneous readings, which are data points that deviate significantly from the expected range.

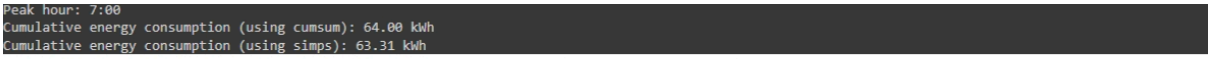
**Model Development**: After gathering and preprocessing the data, we leverage the power of calculus to create predictive models. Derivatives enable us to analyze the rate of change in energy consumption, identifying peak hours and appliance-specific trends. Integrals, on the other hand, help us calculate the cumulative energy consumed over time, providing valuable insights into long-term usage patterns. By implementing these mathematical concepts within a Python-based model.

**PYTHON CODE:**

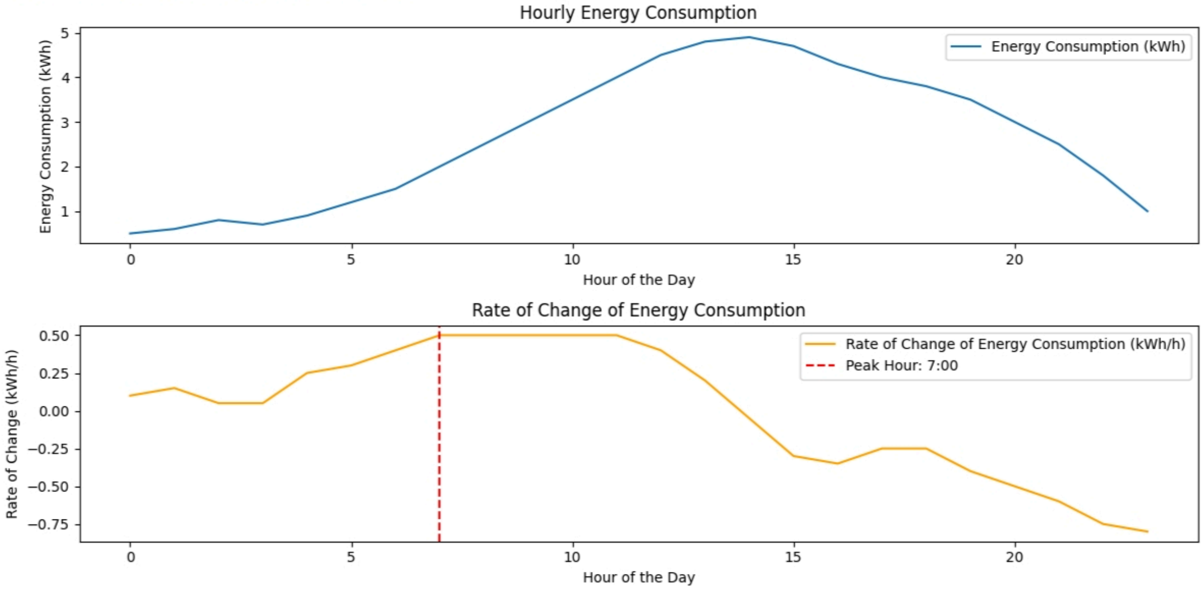


```python
# Sample energy consumption data for demonstration purposes
# Let's assume we have hourly energy consumption data for one day (24 hours)
hours = np.arange(24)
energy_consumption = np.array([0.5, 0.6, 0.8, 0.7, 0.9, 1.2, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0,
                               4.5, 4.8, 4.9, 4.7, 4.3, 4.0, 3.8, 3.5, 3.0, 2.5, 1.8, 1.0])  # kWh

# Derivative: Calculate the rate of change of energy consumption to identify peak hours
rate_of_change = np.gradient(energy_consumption, hours)

# Identify peak hours where the rate of change is maximum
peak_hours = hours[np.argmax(rate_of_change)]
print(f"Peak hour: {peak_hours}:00")

# Integral: Calculate the cumulative energy consumption over time
cumulative_energy = np.cumsum(energy_consumption)  # Cumulative sum of energy consumption

# Alternatively, using numerical integration (Simpson's rule)
cumulative_energy_simps = simps(energy_consumption, hours)

print(f"Cumulative energy consumption (using cumsum): {cumulative_energy[-1]:.2f} kWh")
print(f"Cumulative energy consumption (using simps): {cumulative_energy_simps:.2f} kWh")

# Plot the energy consumption and its rate of change
plt.figure(figsize=(12, 6))
```

THE OUTPUT:



```
Peak hour: 7:00
Cumulative energy consumption (using cumsum): 64.00 kWh
Cumulative energy consumption (using simps): 63.31 kWh
```



Explanation:

1. **Energy Consumption Data**: The script uses a sample hourly energy consumption dataset for one day.
2. **Rate of Change (Derivative)**:
   - The `np.gradient` function is used to compute the rate of change of energy consumption.
   - The hour with the maximum rate of change is identified as the peak hour.
3. **Cumulative Energy (Integral)**:
   - The `np.cumsum` function calculates the cumulative sum of energy consumption over time.
   - Additionally, Simpson's rule (`simps` function from `scipy.integrate`) is used to integrate the energy consumption data numerically for validation.
4. **Plotting**:
   - The script plots the energy consumption data and its rate of change, highlighting the peak hour for better visualization.

## Conclusion:

By combining these analyses, we gain insights into when the most energy is consumed and how the usage changes throughout the day. Identifying peak hours helps in understanding high-demand periods, while cumulative energy consumption gives an overall picture of energy usage, useful for long-term planning and optimization.