```
!pip install scikit-fuzzy
```

```
Requirement already satisfied: scikit-fuzzy in /usr/local/lib/python3.12/dist-packages (0.5.0)
```

```
# Import necessary libraries
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
import warnings

# Suppress a known UserWarning from scikit-fuzzy's internal functions
# This is required for cleaner output in non-interactive environments like Colab.
warnings.filterwarnings("ignore", category=UserWarning)

# --- 1. Restaurant Tipping System (Mamdani FIS) ---

def restaurant_tipping():
    """
    Implements a Fuzzy Inference System to determine the tip percentage
    based on service and food quality using custom Membership Functions (MFs).
    """
    print("--- Running Restaurant Tipping System ---")

    # 1. Define Antecedents (Inputs) and Consequent (Output)
    # Universe of Discourse (UoD) for Service and Food: 0 to 10
    service = ctrl.Antecedent(np.arange(0, 11, 1), 'Service Quality')
    food = ctrl.Antecedent(np.arange(0, 11, 1), 'Food Quality')
    # UoD for Tip: 0 to 25%
    tip = ctrl.Consequent(np.arange(0, 26, 1), 'Tip Percentage', defuzzify_method='centroid')

    # 2. Define Custom Membership Functions (MFs)

    # Service MFs (Improved definitions)
    service['poor'] = fuzz.trimf(service.universe, [0, 0, 5])
    service['average'] = fuzz.trimf(service.universe, [3, 5, 7])
    service['excellent'] = fuzz.trimf(service.universe, [5, 10, 10])

    # Food MFs
    food['bad'] = fuzz.trimf(food.universe, [0, 0, 5])
    food['decent'] = fuzz.trimf(food.universe, [3, 5, 7])
    food['great'] = fuzz.trimf(food.universe, [5, 10, 10])

    # Tip MFs
    tip['low'] = fuzz.trimf(tip.universe, [0, 0, 10])     # Max 10%
    tip['medium'] = fuzz.trimf(tip.universe, [5, 15, 20]) # Peaking at 15%
    tip['high'] = fuzz.trimf(tip.universe, [15, 25, 25])  # Min 15%

    # 3. Define Fuzzy Rules (Nuanced Logic)
    # If service OR food is poor/bad, give a low tip
    rule1 = ctrl.Rule(service['poor'] | food['bad'], tip['low'])
    # If both are decent, give a medium tip
    rule2 = ctrl.Rule(service['average'] & food['decent'], tip['medium'])
    # If service OR food is excellent/great, give a high tip
    rule3 = ctrl.Rule(service['excellent'] | food['great'], tip['high'])
    # Specific rule: If service is excellent but food is bad, still default to low tip
    rule4 = ctrl.Rule(service['excellent'] & food['bad'], tip['low'])

    # 4. Create Control System and Simulation
    tipping_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4])
    tipping_sim = ctrl.ControlSystemSimulation(tipping_ctrl)

    # 5. Input values (Test Case: 6.5 Service, 9.8 Food)
    tipping_sim.input['Service Quality'] = 6.5
    tipping_sim.input['Food Quality'] = 9.8

    # 6. Compute the result
    try:
        tipping_sim.compute()
    except Exception as e:
        print(f"Error computing Tipping System: {e}")
        return

    # 7. Print and visualize the result
    calculated_tip = tipping_sim.output['Tip Percentage']
```

```
    print(f"Input - Service: 6.5, Food: 9.8")
    print(f"Calculated Tip: {calculated_tip:.2f}%")

    # Visualize the results (The problematic line is REMOVED)
    tip.view(sim=tipping_sim)


# --- 2. ABS Brake System Problem (Mamdani FIS) ---

def abs_brake_system():
    """
    Implements a Fuzzy Inference System for ABS to determine the brake pressure
    based on the speed and wheel slip.
    """
    print("\n--- Running ABS Brake System ---")

    # 1. Define Antecedents (Inputs) and Consequent (Output)
    # UoD for Speed: 0 to 100 km/h
    speed = ctrl.Antecedent(np.arange(0, 101, 1), 'Car Speed')
    # UoD for Wheel Slip: 0 to 10 (10 is max slip/wheel lock)
    slip = ctrl.Antecedent(np.arange(0, 11, 1), 'Wheel Slip')
    # UoD for Brake Pressure: 0 to 100
    brake = ctrl.Consequent(np.arange(0, 101, 1), 'Brake Pressure', defuzzify_method='centroid')

    # 2. Define Custom Membership Functions (MFs)

    # Speed MFs
    speed['slow'] = fuzz.trimf(speed.universe, [0, 0, 40])
    speed['medium'] = fuzz.trimf(speed.universe, [20, 50, 80])
    speed['fast'] = fuzz.trimf(speed.universe, [60, 100, 100])

    # Slip MFs (Optimal slip for max braking is around 5)
    slip['low'] = fuzz.trimf(slip.universe, [0, 0, 4])
    slip['optimal'] = fuzz.trimf(slip.universe, [2, 5, 8])
    slip['high'] = fuzz.trimf(slip.universe, [6, 10, 10])

    # Brake Pressure MFs
    brake['soft'] = fuzz.trimf(brake.universe, [0, 0, 40])
    brake['moderate'] = fuzz.trimf(brake.universe, [20, 50, 80])
    brake['hard'] = fuzz.trimf(brake.universe, [60, 100, 100])

    # 3. Define Fuzzy Rules (ABS Logic)
    # Rule 1: High speed, low slip -> Need hard braking to slow down
    rule1 = ctrl.Rule(slip['low'] & speed['fast'], brake['hard'])
    # Rule 2: Optimal slip -> Maintain control with moderate pressure
    rule2 = ctrl.Rule(slip['optimal'], brake['moderate'])
    # Rule 3: High slip (locking) -> Release pressure to regain traction
    rule3 = ctrl.Rule(slip['high'], brake['soft'])
    # Rule 4: Slow speed -> Prevent wheel lock, use soft pressure
    rule4 = ctrl.Rule(speed['slow'], brake['soft'])

    # 4. Create Control System and Simulation
    braking_ctrl = ctrl.ControlSystem([rule1, rule2, rule3, rule4])
    braking_sim = ctrl.ControlSystemSimulation(braking_ctrl)

    # 5. Input values (Test Case: 70 Speed, 3 Slip)
    braking_sim.input['Car Speed'] = 70
    braking_sim.input['Wheel Slip'] = 3

    # 6. Compute the result
    try:
        braking_sim.compute()
    except Exception as e:
        print(f"Error computing ABS System: {e}")
        return

    # 7. Print and visualize the result
    calculated_brake = braking_sim.output['Brake Pressure']
    print(f"Input - Speed: 70, Slip: 3")
    print(f"Calculated Brake Pressure: {calculated_brake:.2f}%")

    # Visualize the results (The problematic line is REMOVED)
    brake.view(sim=braking_sim)

# --- Main Execution ---

if __name__ == '__main__':
```

```
# Run the systems
restaurant_tipping()
abs_brake_system()

# Display all generated plots
plt.show()
```

```
--- Running Restaurant Tipping System ---
Input - Service: 6.5, Food: 9.8
Calculated Tip: 21.66%

--- Running ABS Brake System ---
Input - Speed: 70, Slip: 3
Calculated Brake Pressure: 59.99%
```