```
import pandas as pd
import numpy as np
from datetime import datetime, timedelta

# Set random seed for reproducibility
np.random.seed(42)

# Parameters
n_entries = 1000  # Changed to 1,000 entries
start_date = datetime(2024, 7, 1)
end_date = datetime(2025, 6, 30)
date_range = (end_date - start_date).days

# Define categories
program_types = ['Library Workshop', 'Theatre Event', 'Youth Activity', 'Senior Service', 'Civic Light
Opera']
age_groups = ['Youth', 'Adult', 'Senior']
locations = ['Elmwood Library', 'Hazel Branch Library', 'Starlight Theatre',
             'Maplewood Performing Arts Center', 'Oakleaf Community Center',
             'Willow Park Community Center', 'Silver Elm Senior Center']
grant_funded = ['Yes', 'No']

# Generate data
data = {
    'Program_ID': [f'P{i:04d}' for i in range(1, n_entries + 1)],
    'Program_Type': np.random.choice(program_types, n_entries, p=[0.3, 0.25, 0.2, 0.15, 0.1]),
    'Date': [start_date + timedelta(days=int(np.random.uniform(0, date_range))) for _ in
range(n_entries)],
    'Attendance': [],
    'Participant_Age_Group': [],
    'Satisfaction_Rating': [],
    'Program_Cost': [],
    'Location': [],
    'Grant_Funded': np.random.choice(grant_funded, n_entries, p=[0.3, 0.7]),
    'Economic_Impact': []
}

# Assign values based on Program Type
for program_type in data['Program_Type']:
    if program_type == 'Library Workshop':
        attendance = np.random.randint(50, 150)
        age_group = np.random.choice(['Youth', 'Adult'], p=[0.6, 0.4])
        satisfaction = round(np.random.normal(4.2, 0.5), 1)
        cost = np.random.randint(1000, 2000)
        location = np.random.choice(['Elmwood Library', 'Hazel Branch Library'], p=[0.6, 0.4])
        economic_impact = np.random.randint(2000, 4000)
    elif program_type == 'Theatre Event':
        attendance = np.random.randint(80, 200)
        age_group = 'Adult'
        satisfaction = round(np.random.normal(3.8, 0.6), 1)
        cost = np.random.randint(2000, 4000)
        location = np.random.choice(['Starlight Theatre', 'Maplewood Performing Arts Center'], p=[0.5,
0.5])
        economic_impact = np.random.randint(4000, 8000)
    elif program_type == 'Youth Activity':
        attendance = np.random.randint(40, 100)
        age_group = 'Youth'
        satisfaction = round(np.random.normal(4.4, 0.4), 1)
        cost = np.random.randint(500, 1000)
        location = np.random.choice(['Oakleaf Community Center', 'Willow Park Community Center'],
p=[0.5, 0.5])
        economic_impact = np.random.randint(1000, 2000)
    elif program_type == 'Senior Service':
        attendance = np.random.randint(15, 50)
        age_group = 'Senior'
        satisfaction = round(np.random.normal(4.7, 0.3), 1)
        cost = np.random.randint(300, 800)
        location = 'Silver Elm Senior Center'
        economic_impact = np.random.randint(600, 1600)
    else:  # Civic Light Opera
        attendance = np.random.randint(80, 150)
        age_group = np.random.choice(['Adult', 'Senior'], p=[0.7, 0.3])
        satisfaction = round(np.random.normal(4.0, 0.5), 1)
        cost = np.random.randint(3500, 5000)
        location = 'Maplewood Performing Arts Center'
        economic_impact = np.random.randint(5000, 10000)

    data['Attendance'].append(attendance)
    data['Participant_Age_Group'].append(age_group)
    data['Satisfaction_Rating'].append(satisfaction)
    data['Program_Cost'].append(cost)
    data['Location'].append(location)
    data['Economic_Impact'].append(economic_impact)

# Create DataFrame
df = pd.DataFrame(data)

# Format Date as MM/DD/YYYY
df['Date'] = df['Date'].dt.strftime('%m/%d/%Y')

# Ensure Satisfaction_Rating is between 1 and 5
df['Satisfaction_Rating'] = df['Satisfaction_Rating'].clip(1, 5)

# Sort by Date
df = df.sort_values('Date')
```

```
# Save to CSV
df.to_csv('community_programs_cleaned.csv', index=False)

print(f"Generated {len(df)} entries and saved to community_programs_cleaned.csv")
```