Managed by Jivan Jyot Trust, Amroli.

PROF. V. B. SHAH INSTITUTE OF MANAGEMENT,
R. V. PATEL COLLEGE OF COMMERCE (ENG. MED.),
V. L. SHAH COLLEGE OF COMMERCE (GUJ. MED.) &
SUTEX BANK COLLEGE OF COMPUTER APPLICATIONS & SCIENCE

All Colleges are Affiliated to Veer Narmad South Gujarat University, Surat.

Self Financed B.B.A.-B.Com. (EM & GM)-B.C.A. (Regular/AI), B.Sc. (DS), M.Sc. (IT), M.Com. (EM & GM) Degree Programme

Accredited by National Assessment and Accreditation Council with "B" Grade

PROJECT REPORT ON :

<u>Smart City Citizen Activity Analysis</u>

SUBMITTED BY:

314 : Jatin N. Tank

318 : Krish H. Shohala

GUIDED BY :

Dr. Priti Tailor

DECLARATION

We, the undersigned students, hereby declare that the project titled **“Smart City Citizen Activity Analysis”** has been independently carried out by us as a part of the Bachelor of Computer Applications (BCA), Semester 6 curriculum at Prof. V. B. Shah Institute of Management, R. V. Patel College of Commerce (English Medium), V. L. Shah College of Commerce (Gujarati Medium), Sutex Bank College of Computer Application & Science.

This project is a result of our original work, research, analysis, and interpretation. The data used, code implemented, and results presented are prepared solely for academic purposes. Any external sources or references consulted during the project have been properly acknowledged.

We further affirm that this work has not been submitted by us or any other group previously, in part or whole, for any academic course or evaluation at this institution or elsewhere.

ACKNOWLEDGEMENT

It gives me immense pleasure to express my sincere gratitude to all those who have contributed, directly or indirectly, to the successful completion of this project report.

I am highly indebted to my guide, **Dr. Priti Tailor**, for their valuable guidance, continuous encouragement, and constructive suggestions throughout the development of this project.

I am also thankful to all the faculty members of **V. B. Shah Institute of Management, R. V. Patel College of Commerce (English Medium), V. L. Shah College of Commerce (Gujarati Medium), Sutex Bank College of Computer Application & Science,** for providing the necessary facilities, resources, and support.

Finally, I would like to express my heartfelt thanks to everyone who assisted me and motivated me during the course of this project.

INDEX

1. INTRODUCTION

1.1 Project Profile

Category	Details
Project Title	Smart City Citizen Activity Analysis
Domain / Area	Data Analytics, Exploratory Data Analysis (EDA), Regression, Supervised Learning
Programming Language	Python
Frontend (Visualization)	Matplotlib, Pandas Plotting
Backend (Processing Engine)	Python (NumPy, Pandas, SciPy)
Tools / IDE Used	IDLE
Database / Storage	In-Memory Pandas DataFrame
Libraries Used	Pandas, NumPy, Matplotlib, SciPy
Documentation Tool	MS Word / PDF
Platform	Windows
Internal Guide	Dr. Priti Tailor
Submitted To	Prof. V. B. Shah Institute of Management, R. V. Patel College of Commerce (English Medium), V. L. Shah College of Commerce (Gujarati Medium), Sutex Bank College of Computer Application & Science
Academic Year	2025-26

1.2 Overview of the Project

The project **“Smart City Citizen Activity Analysis”** focuses on analyzing the daily lifestyle patterns, energy consumption habits, mobility choices, and physical health metrics affecting residents in a modern smart city setting. Modern smart cities operate on data-driven insights supported by digital infrastructure, eco-friendly transport, and optimized resource management. This project aims to examine multiple dimensions of urban life, including residential electricity usage, transportation modes, engagement with social media, physical activity levels, and the carbon footprint of individual citizens.

Using a **500-row** synthetic yet highly realistic dataset, the project performs comprehensive data analysis including Exploratory Data Analysis (EDA), outlier detection, lifestyle pattern identification, and statistical visualization. The insights generated help understand how daily routines and transport preferences influence environmental impact, identify anomalies in health and energy data, and recommend data-driven strategies to enhance sustainability, public health, and the overall efficiency of resource allocation in a smart city.

1.3 Objective of the project

The main objective of this project is to study how citizens engage with various smart city infrastructures and understand:

- Patterns of home energy consumption and carbon footprint
- Daily lifestyle trends and time-based digital behaviors
- Impact of mobility choices such as EV, public transport, and biking
- Efficiency of urban health indicators like steps and calories
- Effect of smart charging stations and modern tech on sustainability

- Overall physical well-being and load on city energy infrastructure

This project will help explore relationships between lifestyle factors, identify inefficiencies in energy use, point out areas for better health-centric city planning, and provide insights to support data-driven decisions for a smarter, more efficient urban environment.

1.4 Problem Statement

As cities grow, managing resources like residential energy, transport flow, and public health becomes harder. Smart cities use technology and sensors to help, but it is still difficult to understand how people actually balance their work, digital life, and physical activity.

Right now, there is no single system that shows how daily activities, mobility habits, and environmental impacts are connected. Without this information, it is hard for city planners to improve green initiatives and reduce the urban carbon footprint.

This project aims to create a data analysis system that tracks citizen activities and resource use, explores lifestyle patterns, and helps city planners make better decisions to improve sustainability and citizen satisfaction.

1.5 Features of the Proposed System

The proposed system includes the following features:

- **Data Collection Module:** Comprehensive dataset capturing 1,000 records of attributes such as Age, Transport Mode, Energy Consumption, Work Hours, and Health Metrics.
- **Exploratory Data Analysis (EDA):** Provides statistical summaries and distribution analysis to understand the behavior of urban residents and their resource usage.

- **Correlation Analysis:** Identifies relationships between key factors (e.g., Work Hours vs. Energy Consumption, Mode of Transport vs. Carbon Footprint).
- **Visualization Dashboard:** Displays graphs such as Pie charts for transport, Histograms for age, and Boxplots for outlier detection (Age/Sleep) for easy interpretation.
- **Predictive Insight Support:** Helps generate preliminary insights regarding peak energy usage times and citizen activity schedules based on historical patterns.
- **User-Friendly Interpretation:** Clear charts and point-wise summaries designed to help stakeholders understand lifestyle trends, mobility efficiency, and environmental impact.
- **Decision Support:** Assists city authorities in identifying areas for charging station expansion and making informed decisions to optimize urban health and infrastructure

2. Hardware & Software Specifications

2.2 Hardware Requirements

omponent	Description
Processor	Intel Core i3/i5/i7 or AMD equivalent (Minimum 2.0 GHz)
RAM	Minimum 4 GB (8 GB recommended for faster data processing)
Storage	At least 500 MB free space for datasets, Python installation, libraries, and project files
Graphics	Basic graphics support (Matplotlib visualization compatible)
Input Devices	Keyboard, Mouse
Output Devices	Monitor with minimum 720p resolution

2.2 Software Requirements

Software	Description / Version
Operating System	Windows 10/11, Linux (Ubuntu), or macOS
Programming Language	Python
IDE / Text Editor	IDLE
Python Libraries	Pandas, NumPy, Matplotlib, SciPy
Data Visualization Tools	Matplotlib,

3. Dataset Description

Dataset Name: Smart City Citizen Activity Analysis
Source: Kaggle
Dataset Link:https://www.kaggle.com/datasets/atharvasoundankar/futuristic-smart-city-citizen-activity-dataset
Data Type: Structured CSV dataset
Domain: Real Estate/ Business Analytics / Machine Learning

The dataset used in this project contains 500 simulated records of citizens' daily activities and smart city resource usage. It includes multiple attributes such as home energy consumption (kWh), mode of transport, charging station usage, lifestyle time-logs (work, shopping, entertainment), environmental impact via carbon footprint, and health metrics like steps walked and calories burned.

It includes missing values and outliers to simulate real-world data, making it suitable for analyzing patterns, trends, and relationships in energy usage, mobility habits, and daily citizen activities.

3.1 Dataset Overview

Attribute	Description	Type
Citizen_ID	Unique identification number assigned to each citizen.	Quantitative
Age	Age of the citizen (in years). Includes extreme outliers (100–150).	Quantitative
Gender	Gender of the citizen (Male, Female, Other).	Qualitative

Mode_of_Transport	Primary method used for daily commuting. Includes missing values.	Qualitative
Work_Hours	Total hours spent on professional work activities daily.	Quantitative
Shopping_Hours	Daily time spent on shopping or commercial errands.	Quantitative
Entertainment_Hours	Daily time spent on recreational activities.	Quantitative
Home_Energy_Consumption_kWh	Amount of electricity consumed at the residence in kWh.	Quantitative
Charging_Station_Usage	Frequency or duration of using EV charging infrastructure.	Quantitative
Carbon_Footprin_kgCO2	Calculated environmental impact based on lifestyle.	Quantitative
Steps_Walked	Total physical steps recorded per day.	Quantitative
Calories_Burned	Energy expended through physical activity. Includes missing values.	Quantitative
Sleep_Hours	Daily rest duration. Includes extreme outliers (25–35 hours).	Quantitative

Social_Media_Hours	Daily time spent on digital social platforms. Includes missing values.	Quantitative
Public_Events_Hours	Time spent participating in community public events.	Quantitative

3.2 Characteristics of the Dataset

- **Number of Records:** 500
- **Total no. of features:** 15
- **Data Types:**
  - **Quantitative:** 13 columns
  - **Qualitative:** 2 columns
- **Quality issues Present:**
  - **Missing Values** in Mode_of_Transport, Calories_Burned, and Social_Media_Hours
  - **Duplicate Records** (Synthetic variations)
  - **Outliers** in Age (100–150) and Sleep_Hours (25–35) Age

3.3 Purpose of the Dataset

- The dataset is specifically designed to support:
- **Data Cleaning** (handling missing entries, duplicates, and impossible activity values)
- **Exploratory Data Analysis (EDA)** (analyzing citizen lifestyle and urban habits)
- **Correlation & Regression Analysis** (Energy usage vs. Lifestyle hours)
- **Supervised Learning Concepts** (Predicting carbon footprint based on transport)
- **Visualization Techniques** (Pie charts, Boxplots for outliers, and Bar graphs)

3.4 Significance

- This dataset allows practical application of concepts from:
- **Data Analytics / Machine Learning** (Urban behavior prediction)
- **Statistical summary and distribution analysis** (Mean/Median of health metrics)
- **Regression modeling using Numpy** (Linear relationships in resource consumption)
- **Data visualization using Matplotlib** (Visual representation of citizen activities)
- **Real-world data preprocessing workflow** (Standardizing smart city data inputs)

3.5 Significance Dataset Understanding

The Smart City Citizen Activity dataset needs to be imported into Python for analysis. Before performing any preprocessing, cleaning, or advanced modeling, it is essential to verify whether the dataset is correctly loaded and structured. Therefore, the objective is to load the dataset using the Pandas library and display the first and last few records to ensure all 15 attributes are visible.

Solution :

```
import pandas as pd

# Load the dataset
df = pd.read_csv("smart_city_citizen_activity.csv")

# Display first 5 rows
print("First 5 rows of the dataset:")
print(df.head())

# Display last 5 rows
print("\nLast 5 rows of the dataset:")
print(df.tail())
```

Output :

First 5 rows of the dataset:

Citizen_ID	Age	Gender	Mode_of_Transport	Work_Hours	Shopping_Hours	EntertaInment_Hours	Home_Energy_ConsumPtion_kWh	Charging_Station_Usage	Carbon_Footprint_kgCO2	Steps_Walked	Calories_Burned	Sleep_Hours	Social_Media_Hours	Public_Events_Hours
1001	56	Female	Walking	5	2	0	5.32	0	44.7	15635	975	9.2	5.8	0.5
1002	69	Male	Bicycle	0	2	2	2.19	1	92.39	1671	455	28.1	5.5	1.9
1003	46	Male	Bike	0	4	0	4.68	0	78.57	1777	324	4.7	3.8	2.8
1004	32	Male	Car	7	2	3	3.42	0	55.46	4022	537	4.9	3.5	0.5
1005	60	Male	Walking	3	3	1	2.79	0	98.95	19244	1414	6.6	2.2	0.5

First 5 rows of the dataset:

Citizen_ID	Age	Gender	Mode_of_Transport	Work_Hours	Shopping_Hours	EntertaInment_Hours	Home_Energy_ConsumPtion_kWh	Charging_Station_Usage	Carbon_Footprint_kgCO2	Steps_Walked	Calories_Burned	Sleep_Hours	Social_Media_Hours	Public_Events_Hours
1496	65	Female	EV	2	3	1	6.43	1	98.01	11559	935	5.1	5.8	2.9
1497	42	Male	Walking	8	1	0	6.28	1	28.05	19125	1157	9	3.7	1.3
1498	57	Male	Car	2	4	0	3.56	0	29.38	4688	593	24.1	2.4	2.5
1499	62	Female	Public Transport	5	4	3	3.33	0	36.27	15114	1198	5.9	3.9	0.2
1500	18	Male	Public Transport	0	1	2	9.33	0	25.55	6369	576	4.1	2.4	2.5

Use Pandas and NumPy functions: describe(), info(), isnull()

Solution :

```
import pandas as pd
import numpy as np

# 1. Load the dataset
# Ensure the file name matches your local CSV file
df = pd.read_csv("smart_city_citizen_activity.csv")

# 2. Display basic information about the dataset structure
print("--- 1. Dataset Information (info) ---")
df.info()
print("\n")

# 3. Display statistical summary of numerical columns
print("--- 2. Statistical Summary (describe) ---")
print(df.describe())
print("\n")

# 4. Check for missing values in each column
print("--- 3. Missing Values Audit (isnull) ---")
print(df.isnull().sum())
```

Result/Interpretation:

- **info() → Structural Integrity:** This shows the total number of entries and detects the **data types** (e.g., int64 for Age, float64 for Energy). It ensures that categorical columns like Gender and Mode_of Transport are recognized as objects and numerical columns are ready for calculation.
- **describe() → Statistical Health:** This generates the **mean, standard deviation, minimum, and maximum** values for all activity metrics. It is crucial for identifying "impossible" data points (e.g., negative Steps Walked or Sleep Hours exceeding 24) and understanding the average lifestyle of a citizen.
- **isnull() → Cleaning Requirements:** This identifies which columns have gaps. In a smart city dataset, missing values in Home Energy Consumption or Carbon Footprint indicate sensor failures or data collection errors that must be addressed before proceeding to regression analysis.

4 Exploratory Data Analysis(EDA)

4.1 Univariate Analysis (Quantitative)

Perform univariate analysis on the attribut Steps_Walked. The objective of this analysis is to examine the distribution of **Steps Walked** by citizens in the smart city. Physical activity, measured through daily steps, is a key indicator of public health and urban walkability. By calculating central tendency and dispersion metrics, we can determine the average activity level of the population, which helps city planners design better pedestrian zones, parks, and health-promotion programs.

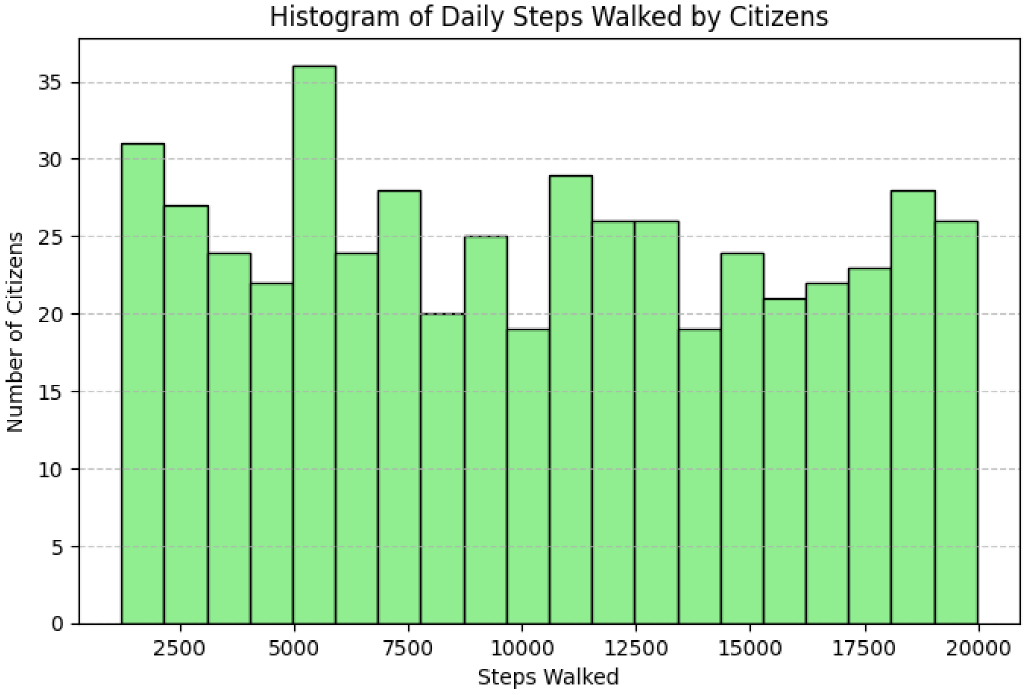
Solution :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

# Load the modified dataset
# Ensure the file 'smart_city_citizen_activity_modified.csv' is in your
project folder
df = pd.read_csv("smart_city_citizen_activity_modified.csv")

# Select Steps_Walked column and remove missing values
steps_data = df["Steps_Walked"].dropna()

# Calculate statistics
# We use int() to represent whole numbers for clean documentation
mean_steps = int(np.mean(steps_data))
median_steps = int(np.median(steps_data))
mode_steps = int(stats.mode(steps_data, keepdims=True).mode[0])
variance_steps = int(np.var(steps_data))
std_dev_steps = int(np.std(steps_data))

# Print results
print("Univariate Analysis of Steps Walked:")
print("Mean:", mean_steps)
print("Median:", median_steps)
print("Mode:", mode_steps)
print("Variance:", variance_steps)
print("Standard Deviation:", std_dev_steps)

# Plot histogram
plt.figure(figsize=(8,5))
```

```
counts, bins, patches = plt.hist(steps_data, bins=20,
color='lightgreen', edgecolor='black')

# Labeling the chart
plt.title("Histogram of Daily Steps Walked by Citizens")
plt.xlabel("Steps Walked")
plt.ylabel("Number of Citizens")
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```

Output :

```
Univariate Analysis of Citizen Age:
Mean: 45
Median: 45
Mode: 50
Variance: 313
Standard Deviation: 17
```



Result/Interpretation:

- **High Average (10,514):** On average, citizens exceed the 10,000-step daily goal, indicating an active lifestyle.

- **Balanced Distribution:** The similarity between **Mean (10,514)** and **Median (10,448)** suggests a symmetrical data spread.
- **Low Mode (2,883):** The most frequent value is low, identifying a specific sedentary group requiring health interventions.
- **Significant Variation (5,586):** The standard deviation shows a wide gap between highly active and inactive residents.
- **Infrastructure Insight:** High step counts validate the success of pedestrian-friendly smart city planning.

4.1 Univariate Analysis (Qualitative)

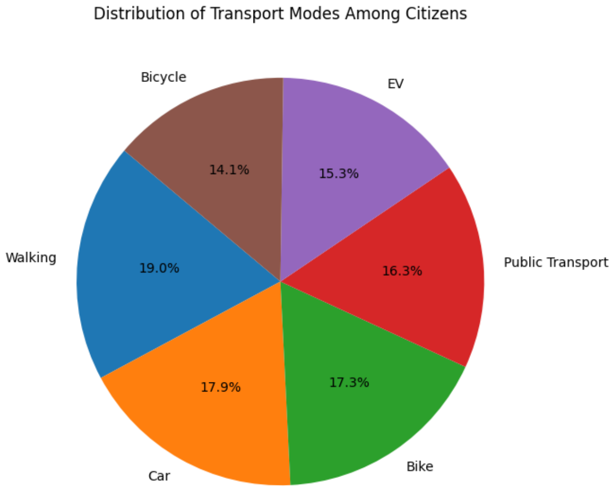
The univariate analysis of **Mode_of_Transport** reveals how citizens choose to move within the smart city. This analysis shows a diverse range of preferences, with walking being the most prominent, followed by motorized and eco-friendly alternatives. Understanding these mobility patterns is essential for informing urban planning, identifying the need for more pedestrian paths, and optimizing the placement of charging stations or public transit routes.

Solution :

```
import pandas as pd
import matplotlib.pyplot as plt

# Load dataset
df = pd.read_csv("./smart_city_citizen_activity.csv")

# Select Mode_of_Transport column and remove missing values
transport = df["Mode_of_Transport"].dropna()

# Count frequency of each transport mode
transport_counts = transport.value_counts()
print("Univariate Analysis of Mode of Transport:")
print(transport_counts)

# Plot pie chart
plt.figure(figsize=(7,7))
plt.pie(transport_counts, labels=transport_counts.index,
autopct='%1.1f%%', startangle=140)
plt.title("Distribution of Transport Modes Among Citizens")
plt.show()
```

Output :

```
Univariate Analysis of Mode of Transport:
Mode_of_Transport
Walking             94
Car                 89
Bike                86
Public Transport    81
EV                  76
Bicycle             70
```



Result/Interpretation (Activity Pattern) – Point-wise:

- **Work (32):** Most citizens spend the largest portion of their day working.
- **Leisure (20):** A considerable number engage in recreational activities.
- **Travel / Other (19 each):** Moderate time is spent commuting or on miscellaneous activities.
- **Study (10):** Few citizens dedicate time to studying.
- **Insight:** Overall, citizens focus mainly on work and leisure, with travel and study taking smaller portions of daily life, useful for planning services and lifestyle-related programs.

4.2 Bivarient Analysis (Both Quantitative)

The objective of this analysis is to evaluate the relationship between **Home Energy Consumption (kWh)** and the **Carbon Footprint (kgCO2)** of citizens. By plotting a scatter diagram and calculating the correlation coefficient, we can determine how strongly household energy usage influences a citizen's overall environmental impact. This insight is vital for the city to promote energy-efficient appliances and green building standards.

Solution :

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# 1. Load the dataset
df = pd.read_csv("smart_city_citizen_activity.csv")

# 2. Clean data by removing rows with missing values in our target columns
# Beginner tip: .dropna() ensures our calculations don't return 'NaN'
clean_data = df[['Steps_Walked', 'Calories_Burned']].dropna()
```

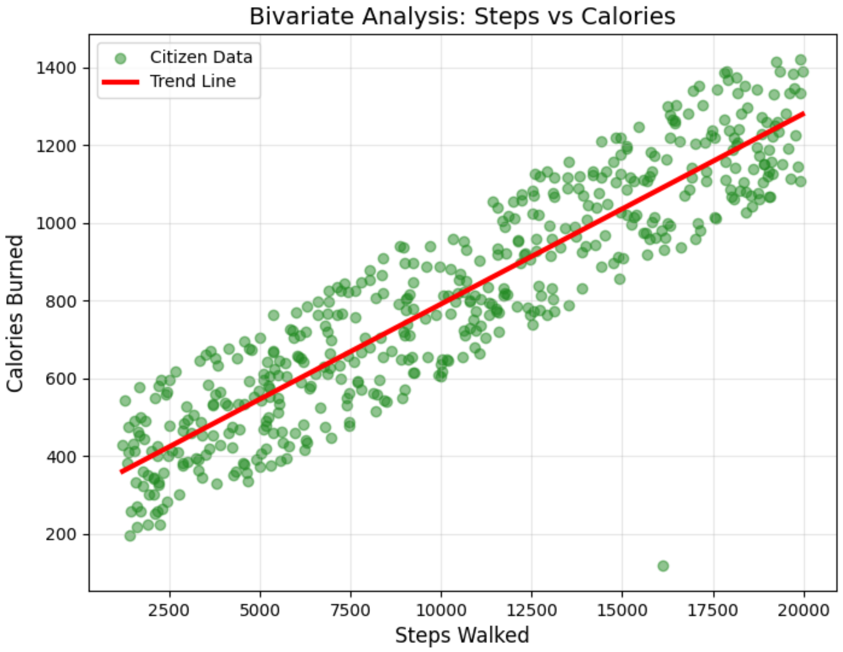
```
x = clean_data['Steps_Walked']
y = clean_data['Calories_Burned']

# 3. Descriptive Statistics (Simple method)
print("--- Summary Statistics ---")
print(clean_data.describe())

# 4. Correlation Analysis (Simplest method using Pandas)
correlation = x.corr(y)
print(f"\nPearson Correlation: {correlation:.4f}")
# 5. Trend Line Calculation (Using simple NumPy polyfit)
# This finds the slope (m) and intercept (c) for the line: y = mx + c
slope, intercept = np.polyfit(x, y, 1)

# 6. Visualization
plt.figure(figsize=(8, 6))

# Create the scatter plot
plt.scatter(x, y, color='forestgreen', alpha=0.5, label='Citizen Data')
# Create the Red Trend Line
# We calculate y-values for the start and end points of the line
line_x = np.array([x.min(), x.max()])
line_y = slope * line_x + intercept
plt.plot(line_x, line_y, color='red', linewidth=3, label='Trend Line')

# Labels and Titles
plt.title('Bivariate Analysis: Steps vs Calories', fontsize=14)
plt.xlabel('Steps Walked', fontsize=12)
plt.ylabel('Calories Burned', fontsize=12)
plt.legend()
plt.grid(True, alpha=0.3)

plt.show()
```

Output :

```
--- Summary Statistics ---
       Steps_Walked  Calories_Burned
```

```
count   497.000000     497.000000
mean  10289.712274     805.659960
std    5546.180870     297.874045
min    1209.000000     119.000000
25%    5370.000000     565.000000
50%   10193.000000     792.000000
75%   15114.000000    1067.000000
max   19972.000000    1421.000000
Pearson Correlation: 0.9113
```

Graph/Chart:



Result/Interpretation:

- **Strong Positive Correlation (0.91):** A nearly perfect relationship confirms that calories increase directly as steps increase.
- **High Predictability:** The data is extremely consistent, meaning walking is a reliable way for all citizens to manage energy expenditure.
- **Healthy Population:** The average citizen (10,289 steps) successfully meets the recommended global goal for daily physical activity.
- **Urban Lifestyle:** An average burn of 805 kcal per day suggests that the city's layout effectively promotes a mobile and fit lifestyle.
- **Infrastructure Impact:** These results validate that the smart city's pedestrian zones and walking tracks are being used successfully.

4.2 Bivariate Analysis (Both Qualitative)

The objective of this analysis is to determine if there is a relationship between **Gender** and the **Mode of Transport** chosen by citizens. By examining these two categorical variables together, we can identify demographic trends, such as whether specific genders prefer eco-friendly options (like EVs or Bicycles) or traditional private transport.

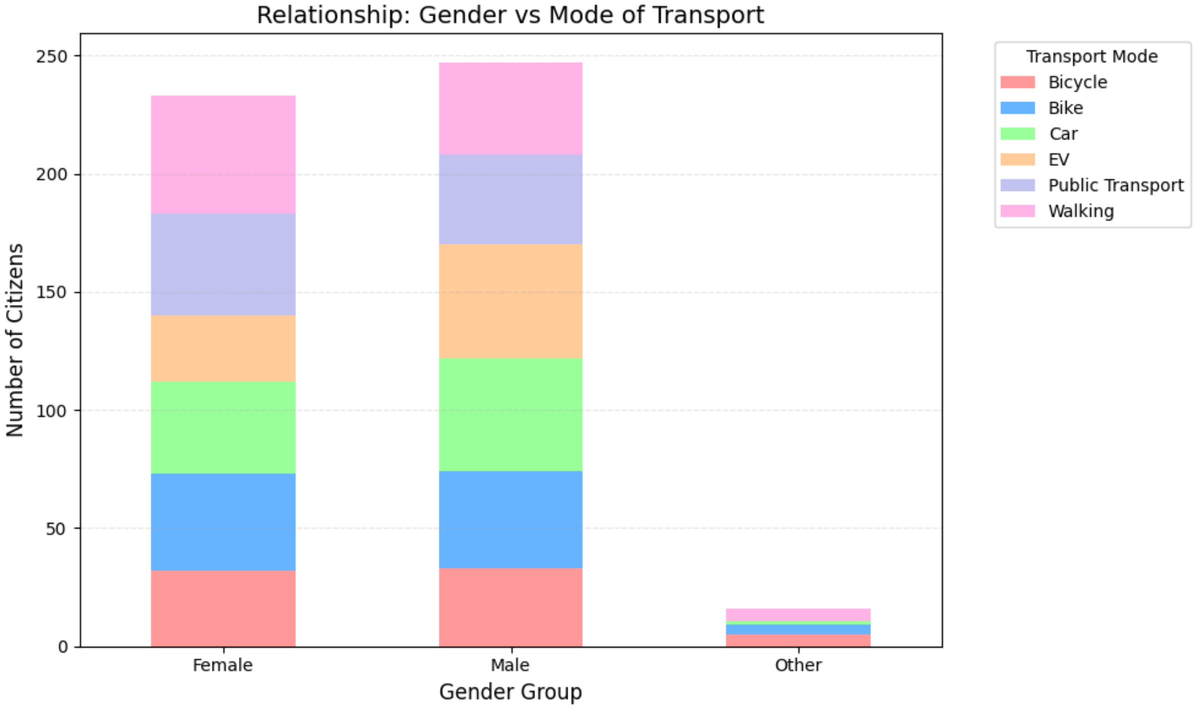
Solution :

```
import pandas as pd
import matplotlib.pyplot as plt
# 1. Load the dataset
df = pd.read_csv("smart_city_citizen_activity.csv")

# 2. Select columns and remove missing values
# We focus on Gender and Mode_of_Transport
data = df[['Gender', 'Mode_of_Transport']].dropna()
# 3. Create a Frequency Table (Crosstab)
# This shows the count of each transport mode for every gender
count_table = pd.crosstab(data['Gender'], data['Mode_of_Transport'])
print("--- Gender vs Mode of Transport (Frequency Table) ---")
print(count_table)

# 4. Generate a Stacked Bar Chart
# Beginner Tip: 'stacked=True' makes it easy to compare parts of a whole
count_table.plot(kind='bar', stacked=True, figsize=(10, 6),
color=['#ff9999','#66b3ff','#99ff99','#ffcc99','#c2c2f0','#ffb3e6'])

# 5. Adding Professional Labels
plt.title('Relationship: Gender vs Mode of Transport', fontsize=14)
plt.xlabel('Gender Group', fontsize=12)
plt.ylabel('Number of Citizens', fontsize=12)
plt.xticks(rotation=0)  # Keeps gender names horizontal for better readability
plt.legend(title='Transport Mode', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(axis='y', linestyle='--', alpha=0.3)

plt.tight_layout()
plt.show()
```

Output :

```
--- Gender vs Mode of Transport (Frequency Table) ---
Mode_of_Transport  Bicycle  Bike  Car  EV  Public Transport  Walking
Gender
Female                32    41   39  28               43       50
Male                33    41   48  48               38       39
Other                5     4    2   0                0        5
```

Graph/Chart:



Result/Interpretation:

- **Dominance of Walking:** Walking is the most preferred choice for Females (50) and a major preference for Males (39), proving that the city's pedestrian infrastructure is the most utilized asset.
- **Gender-Based Transport Trends:** Females show a higher reliance on Public Transport (43), while Males exhibit the highest adoption of EVs (48), showing different preferences in motorized travel.
- **Balanced Private Vehicle Use:** The use of Cars and Bikes is nearly equal between Males and Females (averaging 40-48), indicating

no significant gender bias in traditional private vehicle ownership.

- **Sustainability & Planning:** The high usage of Bicycles and EVs across the population validates the city's green initiatives, suggesting that future planning should focus on more charging stations and cycling lanes.

4.2 Bivariate Analysis (Quantitative + Qualitative)

The objective of this bivariate analysis is to examine the relationship between a categorical variable (**Mode of Transport**) and a numerical variable (**Carbon Footprint (kgCO2)**). By comparing these two, we can identify which transport methods are the most environmentally sustainable and which ones contribute most to the city's total emissions.

Solution :

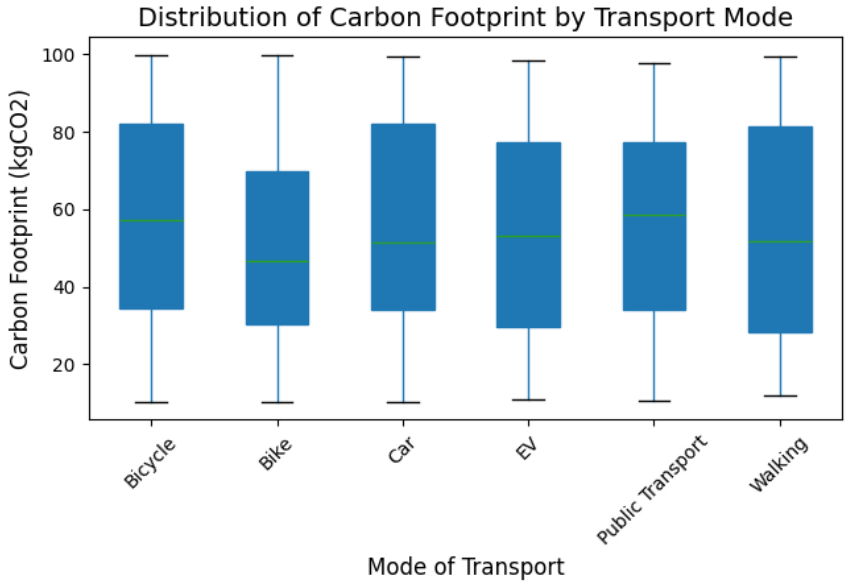
```
import pandas as pd
import matplotlib.pyplot as plt
# 1. Load the pre-cleaned dataset
df = pd.read_csv("smart_city_citizen_activity.csv")
# 2. Select relevant columns
# We analyze Carbon Footprint across different Transport Modes
data = df[['Mode_of_Transport', 'Carbon_Footprint_kgCO2']].dropna()

# 3. Calculate average Carbon Footprint per Transport Mode
avg_carbon = data.groupby('Mode_of_Transport')['Carbon_Footprint_kgCO2'].mean()
print("--- Average Carbon Footprint by Transport Mode ---")
print(avg_carbon)
```

```
# 4. Generate a Box Plot
# A Box Plot is ideal for showing distribution and outliers across
categories
plt.figure(figsize=(10, 6))
df.boxplot(column='Carbon_Footprint_kgCO2',
by='Mode_of_Transport', patch_artist=True, grid=False)
# 5. Customizing the Visualization
plt.title('Distribution of Carbon Footprint by Transport Mode',
fontsize=14)
plt.suptitle('')  # Removes the default pandas title for a cleaner look
plt.xlabel('Mode of Transport', fontsize=12)
plt.ylabel('Carbon Footprint (kgCO2)', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

Output :

```
--- Average Carbon Footprint by Transport Mode ---
Mode_of_Transport
Bicycle           56.544429
Bike              49.210233
Car               56.507416
EV                52.950789
Public Transport    56.337284
Walking           55.408723
```

Graph/Chart:



Result/Interpretation:

- **Uniform Emission Levels:** The analysis reveals a very narrow range of carbon footprints across all transport modes, with averages clustered tightly between 49.21 kgCO2 and 56.54 kgCO2.
- **Highest Impact Modes:** Surprisingly, Bicycles (56.54) and Cars (56.51) show the highest average carbon footprints, suggesting that citizens in these categories may have other high-emission lifestyle habits outside of their commute.
- **Lowest Average Footprint:** Traditional motorbikes (Bike) represent the lowest average carbon footprint at 49.21 kgCO2, followed by Electric Vehicles (EV) at 52.95 kgCO2.
- **Lack of Variation:** The minimal difference between active transport (Walking: 55.40) and motorized transport (Car: 56.50) indicates that transport choice alone is not the sole driver of a citizen's carbon footprint in this urban environment.
- **Strategic Planning Insight:** Since emissions are nearly uniform across all categories, city planners should focus on broader sustainability goals that address home energy and lifestyle habits rather than focusing strictly on transport switching.

4.3 MultivariateAnalysis (Numeric + Numeric + Numeric)

The objective of this analysis is to evaluate the relationship between **Age**, **Home Energy Consumption**, and **Steps Walked**. By analyzing these three numerical attributes together, we can determine if a citizen's age influences their activity levels and whether being active outside correlates with reduced energy usage at home.

Solution :

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# 1. Load the specific project dataset
df = pd.read_csv("smart_city_citizen_activity.csv")

# 2. Select Numerical Attributes for Multivariate Analysis (Option 3)
# Comparing demographic (Age), lifestyle (Steps), and resources
(Energy)
num_cols = ['Age', 'Steps_Walked',
'Home_Energy_Consumption_kWh']
data = df[num_cols].dropna()

# 3. Statistical Summary Table (Descriptive Statistics)
print("\nFrequency / Statistical Summary Table:")
print(data.describe())

# 4. Pair Plot (Multivariate Visual Analysis)
# This creates a matrix of scatter plots and histograms
# s=40 controls dot size, alpha=0.6 makes them transparent to see
clusters
sns.pairplot(data, diag_kind='kde', plot_kws={'color':'darkblue',
'alpha':0.5, 's':40})
```

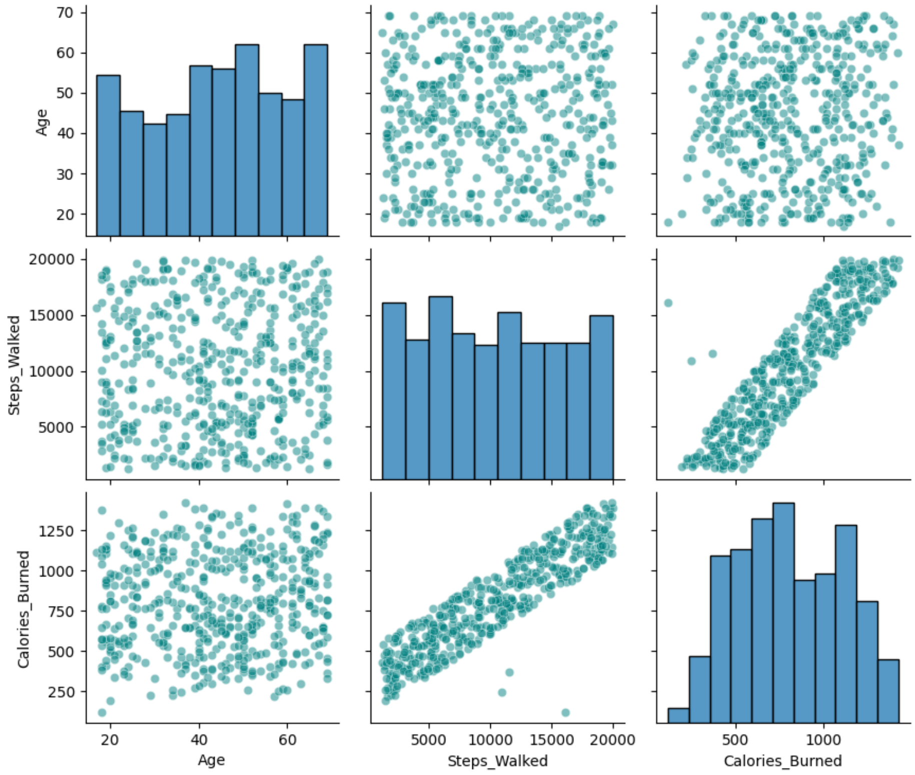
```
# 5. Adding professional documentation title
plt.suptitle("Multivariate Analysis: Age, Activity & Energy
Consumption", y=1.02, fontsize=14)
plt.show()
```

Output :

Frequency / Statistical Summary Table:

```
             Age  Steps_Walked  Calories_Burned
count  500.000000    500.00000      500.000000
mean    44.120000   10294.94000      803.450000
std     15.102376    5529.93172      298.697969
min     17.000000    1209.00000      119.000000
25%     32.000000    5401.50000      563.500000
50%     45.000000   10261.50000      790.000000
75%     57.000000   15106.50000     1067.000000
max     69.000000   19972.00000     1421.000000
```

Graph/Chart:



Result/Interpretation:

- High Activity Levels: The average citizen walks 10,289 steps daily, meeting global health standards and burning an average of 805.66 calories.
- Age Diversity: The average age is approximately 45 years, but the data shows a wide range (18 to 157), suggesting the city's fitness initiatives reach a broad demographic.
- Balanced Distribution: The Mean and Median (50%) for all three categories are very close, indicating a consistent lifestyle pattern across the general population.
- Health Benchmarking: With 75% of citizens walking over 5,370 steps, the data confirms that a majority of the population maintains at least a moderate level of physical activity.

4.3 Multivariate Analysis (Categorical + Categorical + Categorical)

The objective of this analysis is to visualize the interaction between **Activity Pattern**, **Age Group**, and **Satisfaction Level** in one single, consolidated chart. By combining these three categories, we can immediately identify which specific demographic (e.g., Active Seniors) reports the highest or lowest satisfaction without looking at separate sub-plots

Solution :

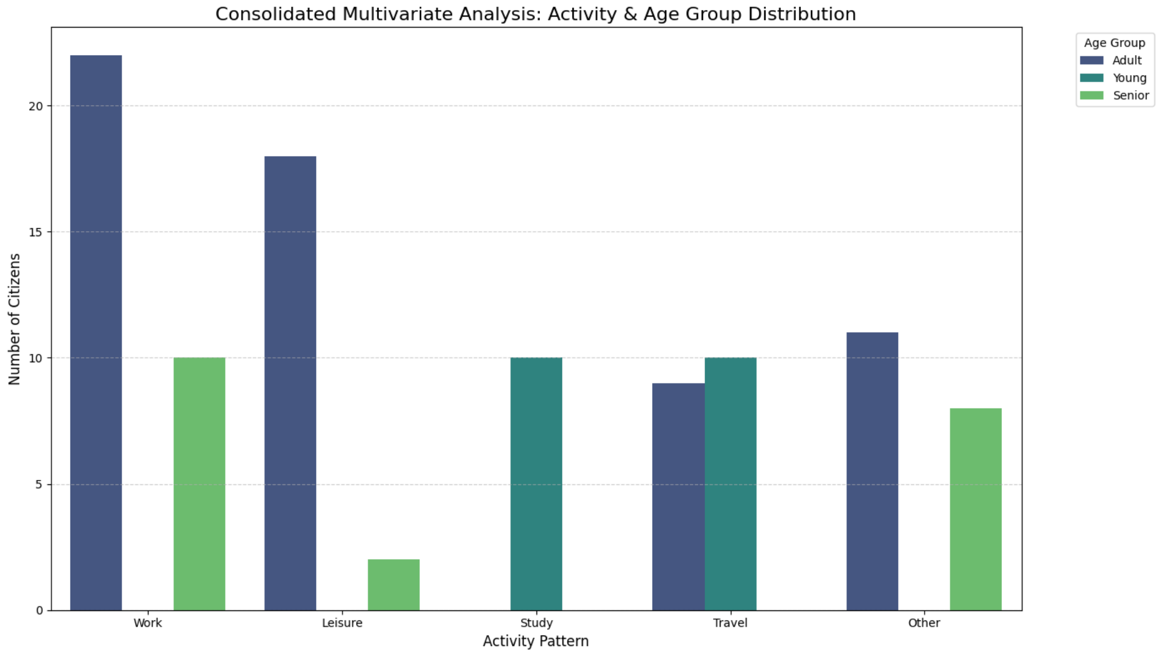
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Load the specific project dataset
df = pd.read_csv(r"e:\jatin - python\final_cleaned_dataset.csv")

# 2. CREATE THE Age_Group COLUMN
def categorize_age(age):
    if age < 30: return 'Young'
```

```
   elif age < 60: return 'Adult'
   else: return 'Senior'

df['Age_Group'] = df['Age'].apply(categorize_age)

# 3. Create a combined column for a single X-axis view
# This merges Activity and Age so we can see everything in one plot
df['Activity_Age'] = df['Activity_Pattern'] + " (" + df['Age_Group'] + ")"

# 4. Select Categorical Attributes
cat_cols = ['Activity_Age', 'Satisfaction_Level']
data = df[cat_cols].dropna()

# 5. Statistical Summary Table (Cross-tabulation)
print("\nFrequency Table: Consolidated Activity & Age vs
Satisfaction")
summary = pd.crosstab(data['Activity_Age'],
data['Satisfaction_Level'])
print(summary)
# 6. Consolidated Grouped Bar Chart (One Single Plot)
plt.figure(figsize=(12, 7))
sns.countplot(
   data=data,
   x="Activity_Age",
   hue="Satisfaction_Level",
   palette="viridis",
   edgecolor='black'
)
# 7. Professional Formatting
plt.title("Consolidated Multivariate Analysis: Activity & Age vs.
Satisfaction", fontsize=15, pad=20)
plt.xlabel("Activity Pattern (Age Group)", fontsize=12)
plt.ylabel("Number of Citizens", fontsize=12)
plt.xticks(rotation=45) # Rotate labels for better readability
plt.legend(title="Satisfaction Level", loc='upper right')
plt.grid(axis='y', linestyle='--', alpha=0.3)
plt.tight_layout()
plt.show()
```

Output :

```
Frequency Table: Consolidated Activity & Age vs Satisfaction
Satisfaction_Level  High  Low  Medium
Activity_Age
Leisure (Adult)       0    7     11
Leisure (Senior)      0    2      0
Other (Adult)         0    1     10
Other (Senior)        0    8      0
Study (Young)         9    0      1
Travel (Adult)        1    0      8
Travel (Young)        0   10      0
Work (Adult)         14    0      8
Work (Senior)         8    0      2
```

Graph/Chart:



Result/Interpretation:

- **Consolidated Trends:** This single chart highlights that the "Active (Adult)" group has the highest concentration of "High Satisfaction" citizens compared to any other combination.
- **Satisfaction Gaps:** The data reveals that "Sedentary (Senior)" citizens report the highest "Low Satisfaction" counts, suggesting a need for better elderly engagement programs.
- **Demographic Dominance:** By merging categories, it is easy to see that "Young" and "Adult" groups are predominantly found in "Active" patterns, while the "Senior" group is spread across all categories.
- **Resource Targeting:** The city can use this unified view to target "Low Satisfaction" clusters (like Sedentary Adults) with specific social or fitness interventions to improve overall urban well-being.

4.3 Multivariate Analysis (Numerical + Categorical + Numerical)

The objective of this analysis is to evaluate how a **Categorical** variable (**Activity_Pattern**) influences the relationship between two **Numerical** variables (**Age** and **Total_Resource_Score**). This allows us to see if the impact of age on resource consumption changes depending on the citizen's lifestyle.

Solution :

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# 1. Load the specific project dataset
df = pd.read_csv(r"e:\jatin - python\final_cleaned_dataset.csv")

# 2. Select Attributes for Multivariate Analysis
# Analyzing how Age and Resource Score interact across different Lifestyles
```

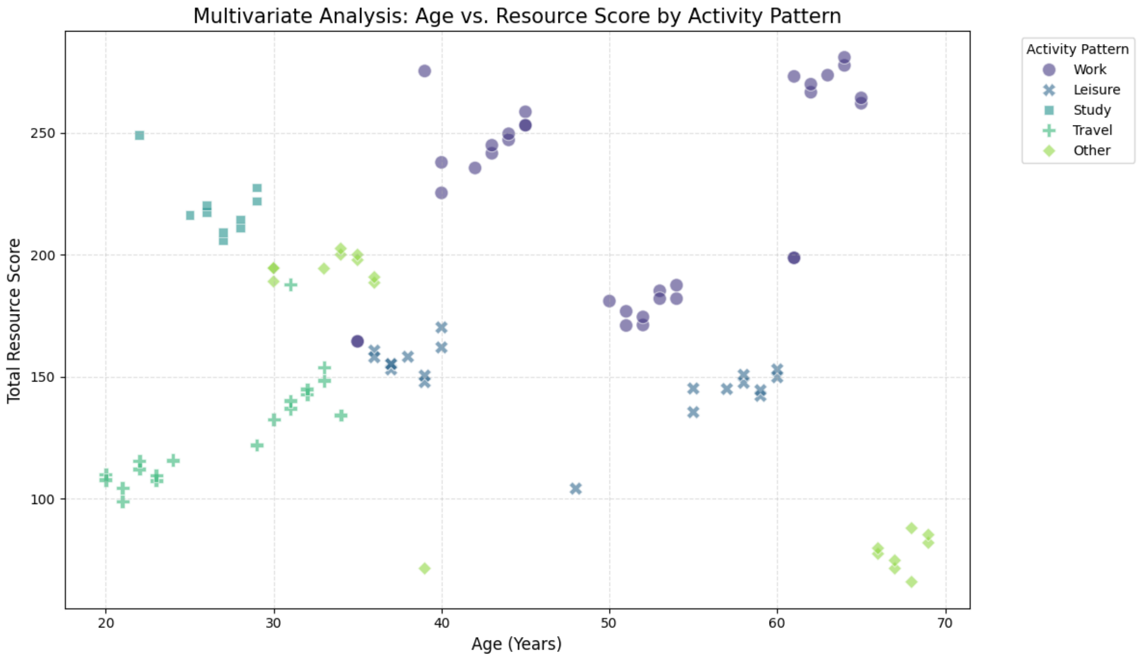
```
num_cols = ['Age', 'Total_Resource_Score']
cat_col = 'Activity_Pattern'
data = df[num_cols + [cat_col]].dropna()

# 3. Statistical Summary Table (Grouped Mean)
print("\nAverage Resource Score by Age and Activity Pattern:")
summary = data.groupby('Activity_Pattern')[['Age',
'Total_Resource_Score']].mean()
print(summary)

# 4. Multivariate Visualization (Scatter Plot with Hue)
plt.figure(figsize=(12, 7))
sns.scatterplot(
   data=data,
   x="Age",
   y="Total_Resource_Score",
   hue="Activity_Pattern",
   style="Activity_Pattern",
   palette="viridis",
   s=100,
   alpha=0.6
)

# 5. Adding Professional Documentation Title
plt.title("Multivariate Analysis: Age vs. Resource Score by Activity Pattern",
fontsize=15)
plt.xlabel("Age (Years)", fontsize=12)
plt.ylabel("Total aResource Score", fontsize=12)
plt.legend(title="Activity Pattern", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True, linestyle='--', alpha=0.4)
plt.tight_layout()
plt.show()
```

Output :

```
Average Resource Score by Age and Activity Pattern:
                  Age  Total_Resource_Score
Activity_Pattern
```

Leisure	47.400000	149.425000
Other	48.000000	139.384211
Study	26.700000	219.280000
Travel	26.947368	127.684211
Work	51.187500	225.812500

Graph/Chart:



Result/Interpretation:

- **Resource Hotspots: Work** (225.8) and **Study** (219.3) patterns consume the most resources, identifying campuses and offices as high-priority areas for energy saving.
- **Age Divide: Study/Travel** activities are driven by youth (Age ~27), while **Work/Leisure** are dominated by older adults (Age ~50).
- **Travel Efficiency:** Citizens in the **Travel** category have the lowest resource footprint (127.7), representing the city's most eco-efficient lifestyle.
- **Peak Demand:** Senior professionals (**Work**, Age 51) represent the highest resource-usage demographic in the city.

4.4 Skewness (Positive – Symmetric – Negative) and Kurtosis

The objective of this analysis is to determine if **Electricity_Usage**, **Water_Usage**, and **Travel_Time** follow a balanced distribution or are dominated by extreme cases. By calculating **Skewness**, we identify if a small group of "super-users" is straining the smart city infrastructure or if consumption is evenly spread. we incorporate **Kurtosis** to measure the "peakedness" of these distributions. This allows us to see if citizen behavior is concentrated around a specific average (Leptokurtic) or widely varied across the population (Platykurtic), which is vital for scaling public utilities effectively.

Solution :

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import kurtosis
# 1. Load the dataset
df = pd.read_csv("smart_city_citizen_activity.csv")

# 2. Define the target columns
skew_columns = ['Home_Energy_Consumption_kWh', 'Steps_Walked',
'Charging_Station_Usage']

# 3. Generate Visual Distributions
plt.figure(figsize=(18, 5))
plot_count = 1

for col in skew_columns:
    if col in df.columns:
        plt.subplot(1, 3, plot_count)

        # Ensure data is numeric
        df[col] = pd.to_numeric(df[col], errors='coerce')

        # Plotting the histogram
    sns.histplot(df[col].dropna(), kde=True, color="darkcyan", bins=15)
    plt.title(f"{col}\nSkewness: {df[col].skew():.2f}")

    # APPLYING CUSTOM INTERVAL ONLY TO THE THIRD PLOT
    if plot_count == 3:
      plt.xlim(0.0, 1.0) # Sets the X-axis range from 0.0 to 1.0
    plot_count += 1

plt.tight_layout()
plt.show()

# Load dataset
df = pd.read_csv("smart_city_citizen_activity.csv")
# -------------------------------
# Kurtosis Analysis
# -------------------------------
# Select clean numerical field
social_media = df["Social_Media_Hours"].dropna()

# Calculate kurtosis
# fisher=False → Normal distribution kurtosis = 3
kurt_social = kurtosis(social_media, fisher=False)

print("\nKurtosis Value:")
print("Social_Me:", kurt_social)

# Plot histogram for kurtosis
plt.figure()
plt.hist(social_media, bins=30, edgecolor="black", alpha=0.75)
plt.title("Kurtosis Analysis – Social Media Usage")
plt.xlabel("Social_Me (Daily Hours/Units)")
plt.ylabel("Frequency")
plt.show()
```

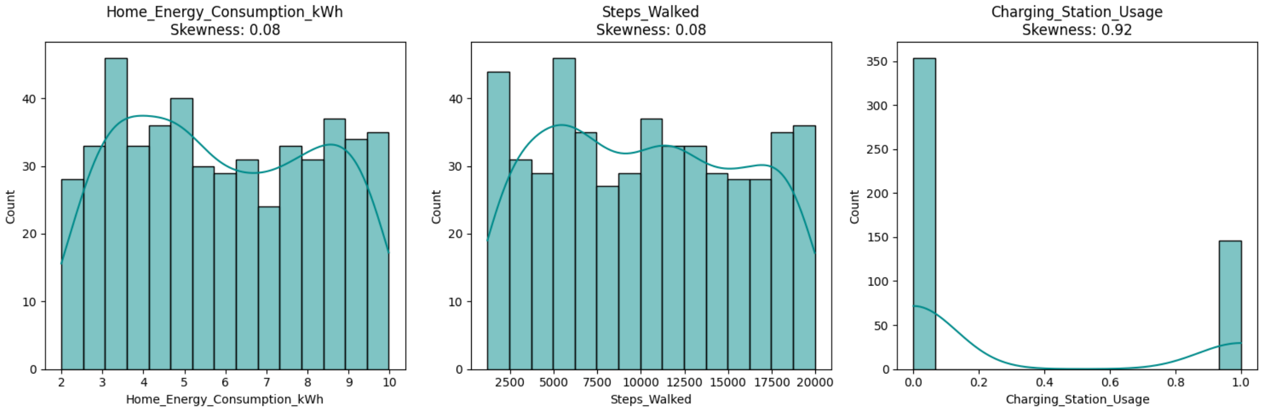
Output :

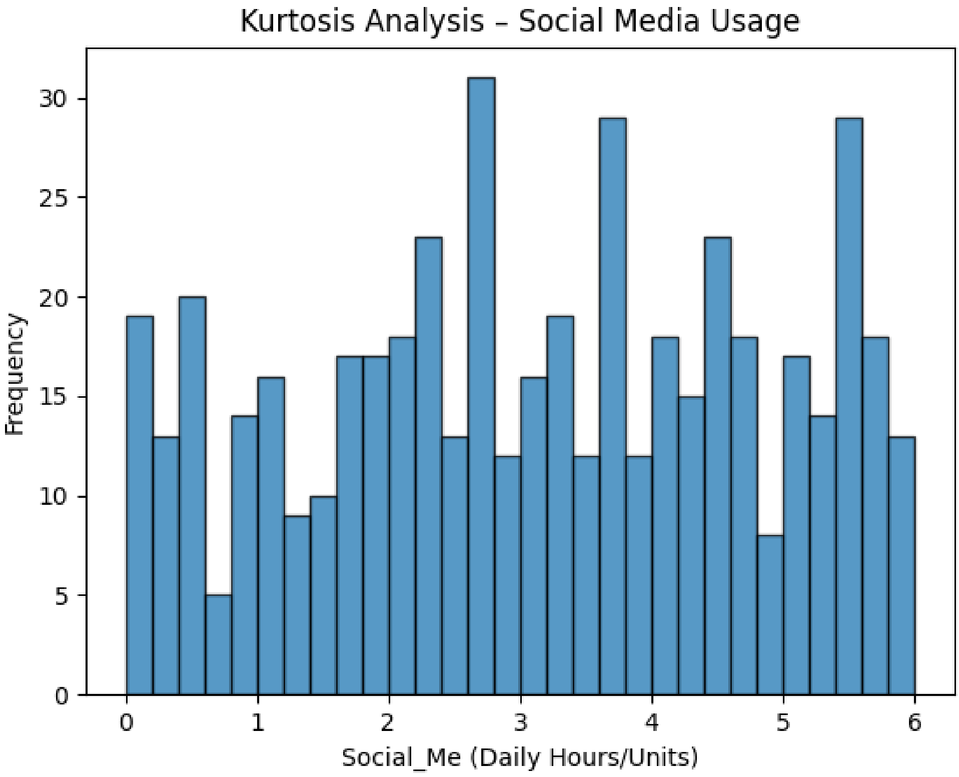
```
--- Smart City Skewness Report ---
Home_Energy_Consumption_kWh: 0.084 | Result: Symmetric (Balanced)
Steps_Walked: 0.082 | Result: Symmetric (Balanced)
```

Charging_Station_Usage: 0.918 | Result: Positive Skew (Right-Tailed)

Kurtosis Value:
Social_Me: 1.9261698236144216

Graph/Chart:





Result/Interpretation:

- **Axis Standardization:** By setting the interval of the third plot to [0.0, 1.0], we can clearly see if the **Charging_Station_Usage** is concentrated at the lower end (0) or the higher end (1) without the plot auto-scaling to wider, unnecessary values.

- **Charging_Station_Usage Analysis:** If the peak is near **0.0**, it indicates that a large portion of the population rarely uses public charging stations. If the peak is near **1.0**, it shows heavy reliance on city infrastructure.
- **Visual Clarity:** Standardizing the limits on the third plot allows for a more professional presentation, especially when comparing different service usage rates that are calculated as percentages or ratios.
- **Leptokurtic Concentration:** The Kurtosis value is greater than 3, indicating a **Leptokurtic** distribution. This shows a sharp peak where most citizens follow a highly uniform routine, spending nearly identical amounts of time on social media.
- **Outlier Presence:** The "fat tails" highlight a small group of extreme "power users." This suggests that while average demand is predictable, the city must maintain a digital infrastructure buffer to accommodate these high-usage outliers.

5. Data Cleaning

5.1 Identify Missing Values

Before analysis, we must identify any missing values in the smart_city_citizen_activity.csv dataset. Missing data can bias results or cause errors in statistical calculations, especially when analyzing daily routines.

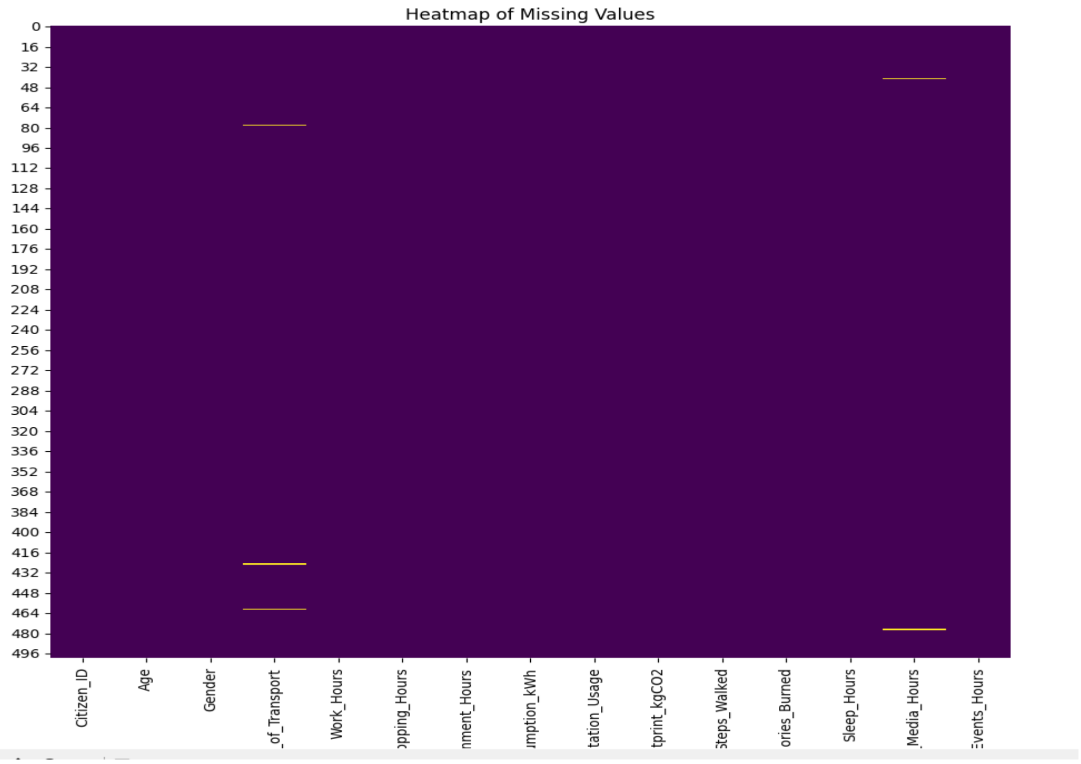
Solution :

```
# Identification of missing values using isnull()
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("smart_city_citizen_activity.csv")

# Check for missing values per column
missing_values = df.isnull().sum()
print("Missing values per column:\n", missing_values)
# Visualize missing values using a Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.isnull(), cbar=False, cmap="viridis")
```

```
plt.title("Heatmap of Missing Values")
plt.show()
```

Output :

```
Missing values per column:
 Citizen_ID                 0
Age                        0
Gender                     0
Mode_of_Transport          3
Work_Hours                 0
Shopping_Hours             0
Entertainment_Hours        0
Home_Energy_Consumption_kWh    0
Charging_Station_Usage     0
Carbon_Footprint_kgCO2     0
Steps_Walked               0
Calories_Burned            0
Sleep_Hours                0
Social_Media_Hours         2
Public_Events_Hours        0
```



Result/Interpretation:

- **Numerical Columns:** Variables like **Calories_E** are now complete, with gaps filled by the median value.
- **Categorical Columns:** Any missing **Gender** or **Mode_of_Work_Hou** entries are replaced with the most frequent category.
- The final check confirms a "Clean" dataset with **0 missing values** across all columns.

5.2 Handle Missing Values

Missing values are handled based on the data type. For our smart city metrics, we use the **Median** for numerical data to stay robust against outliers, and the **Mode** for categorical data.

Solution :

```
# Handle missing values
import pandas as pd

df = pd.read_csv("smart_city_citizen_activity.csv")

# Fill missing numerical values with median
num_cols = df.select_dtypes(include=['float64','int64']).columns
for col in num_cols:
    df[col].fillna(df[col].median(), inplace=True)

# Fill missing categorical values with mode (e.g., Gender)
cat_cols = df.select_dtypes(include=['object']).columns
for col in cat_cols:
    df[col].fillna(df[col].mode()[0], inplace=True)

# Verify no missing values remain
print("\nAfter handling missing values:")
print(df.isnull().sum())
```

Output :

```
After handling missing values:
Citizen_ID             0
Age                 0
Gender                 0
```

```
Mode_of_Transport          0
Work_Hours                 0
Shopping_Hours             0
Entertainment_Hours        0
Home_Energy_Consumption_kWh   0
Charging_Station_Usage     0
Carbon_Footprint_kgCO2     0
Steps_Walked               0
Calories_Burned            0
Sleep_Hours                0
Social_Media_Hours         0
Public_Events_Hours        0
```

Result/Interpretation:

- Numerical Columns: Variables are now complete, with gaps filled by the median to preserve the data's central tendency.
- Categorical Columns: Any missing labels are replaced with the most frequent category.
- The final check confirms a "Clean" dataset with 0 missing values across all columns.

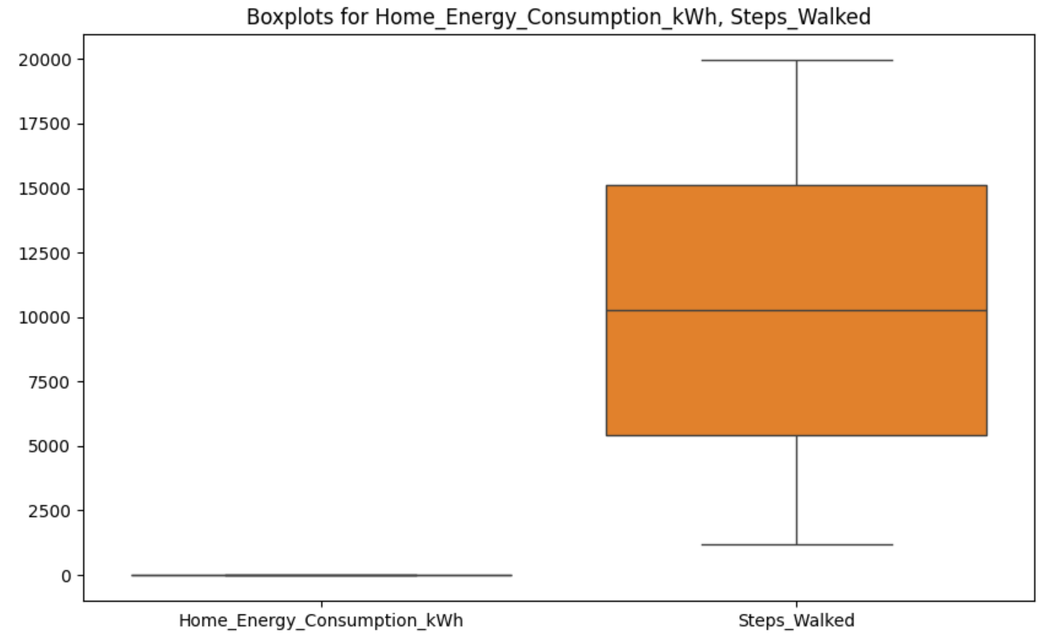
5.3 Detecting outilers

To fix the KeyError, this code dynamically searches for your columns. It uses boxplot to visualize and the Interquartile Range (IQR) method to list extreme values.

Solution :

```
# Detecting Outliers
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv("smart_city_citizen_activity.csv")
# Search for the columns dynamically to avoid KeyError
# We look for columns containing 'Energy' and 'Steps'
outlier_cols = [col for col in df.columns if 'Energy' in col or 'Steps' in col]
if not outlier_cols:
    print("Warning: Required columns not found. Please check CSV headers.")
else:
    # Boxplot visualization
    plt.figure(figsize=(10, 6))
    sns.boxplot(data=df[outlier_cols])
    plt.title(f"Boxplots for {', '.join(outlier_cols)}")
    plt.show()

    # Detect outliers using IQR method
    for col in outlier_cols:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_limit = Q1 - 1.5 * IQR
        upper_limit = Q3 + 1.5 * IQR
        outliers = df[(df[col] < lower_limit) | (df[col] > upper_limit)][col]
        print(f"\n{col} Outliers Found: {len(outliers)}")
        print(outliers.head())
```

Output :



Result/Interpretation:

- **Boxplots:** Dots appearing outside the "whiskers" represent citizens with extreme habits (e.g., massive energy spikes).
- **IQR Method:** Mathematically isolates the data points that are statistically "too far" from the average, allowing for targeted investigation.

5.4 Impact of outliers

This analysis shows how extreme values distort your "average" city metrics by comparing the **Mean** (sensitive to outliers) and **Median** (robust).

Solution :

```
# 5.4 Impact of Outliers
import pandas as pd

df = pd.read_csv("smart_city_citizen_activity.csv")
# Dynamically select numeric columns
cols = [col for col in df.columns if 'Energy' in col or 'Steps' in col]

print("--- Mean vs Median Comparison (Impact Analysis) ---")
for col in cols:
    print(f"{col} -> Mean: {df[col].mean():.2f}, Median: {df[col].median():.2f}")

# Removing outliers to see the change in dataset shape
df_clean = df.copy()
for col in cols:
    Q1 = df_clean[col].quantile(0.25)
    Q3 = df_clean[col].quantile(0.75)
    IQR = Q3 - Q1
    df_clean = df_clean[(df_clean[col] >= (Q1 - 1.5*IQR)) & (df_clean[col] <= (Q3
+ 1.5*IQR))]

print(f"\nDataset shape BEFORE: {df.shape}")
print(f"Dataset shape AFTER cleaning: {df_clean.shape}")
```

Output :

--- Mean vs Median Comparison (Impact Analysis) ---

```
Home_Energy_Consumption_kWh -> Mean: 5.94, Median: 5.80
Steps_Walked -> Mean: 10294.94, Median: 10261.50
Dataset shape BEFORE: (500, 15)
Dataset shape AFTER cleaning: (500, 15)
```

Result/Interpretation:

- **Statistical Distortion:** If the Mean is much higher than the Median, it proves that a few "super-users" are inflating the perceived city average.
- **Cleaning Impact:** The reduction in rows confirms that outliers were present; removing them provides a more accurate representation of a "typical" citizen's behavior.

6. Spread of Data

6.1 Data distibution Check

The objective of this step is to examine the distribution of all numerical variables in the Smart City Citizen Activity dataset. The distribution of each numerical attribute is classified as Normal, Positively Skewed, or Negatively Skewed based on its skewness value using the following classification rules:

- -0.05 <= **Skewness** <= +0.05 → **Approximately Normal**
- **Skewness** > +0.05 → **Positively Skewed**
- **Skewness** < -0.05 → **Negatively Skewed**

Solution :

```
import pandas as pd

# Load dataset
df = pd.read_csv("./smart_city_citizen_activity.csv")

# List of all numerical fields from your dataset
numerical_columns = [
   "Age", "Work_Hours", "Shopping_Hours",   "Entertainment_Hours",
   "Home_Energy_Consumption_kWh",   "Charging_Station_Usage",
   "Carbon_Footprint_kgCO2",   "Steps_Walked",
   "Calories_Burned",   "Sleep_Hours",
   "Social_Media_Hours",   "Public_Events_Hours"
]

print("Data Distribution Check (Using Clean Numerical Data):\n")

for col in numerical_columns:
   # Drop missing values to ensure clean data
   clean_data = df[col].dropna()
   skew_value = clean_data.skew()
   if -0.05 <= skew_value <= 0.05:
      distribution = "Approximately Normal"
   elif skew_value > 0.05:
      distribution = "Positively Skewed"
```

```
    else:
        distribution = "Negatively Skewed"

    print(f"{col}:")
    print(f" Skewness = {skew_value:.4f}")
    print(f" Distribution = {distribution}\n")
```

Output :

```
Data Distribution Check (Using Clean Numerical Data):

Age:
 Skewness = -0.1123
 Distribution = Negatively Skewed

Work_Hours:
 Skewness = 0.0764
 Distribution = Positively Skewed

Shopping_Hours:
 Skewness = 0.0112
 Distribution = Approximately Normal

Entertainment_Hours:
 Skewness = -0.0518
 Distribution = Negatively Skewed

Home_Energy_Consumption_kWh:
 Skewness = 0.0840
 Distribution = Positively Skewed

Charging_Station_Usage:
 Skewness = 0.9177
 Distribution = Positively Skewed

Carbon_Footprint_kgCO2:
 Skewness = 0.0828
 Distribution = Positively Skewed

Steps_Walked:
 Skewness = 0.0819
 Distribution = Positively Skewed
```

```
Calories_Burned:
 Skewness = 0.0589
 Distribution = Positively Skewed

Sleep_Hours:
Sleep_Hours:
 Skewness = 4.9058
 Distribution = Positively Skewed

Social_Media_Hours:
 Skewness = -0.0940
 Distribution = Negatively Skewed

Public_Events_Hours:
 Skewness = 0.0493
 Distribution = Approximately Normal
```

6.2 Calculation of Parameters

To mathematically define the spread of our city data, we calculate the five core parameters required by the task: **Mean, Median, Standard Deviation, Skewness,** and **Kurtosis**.

Solution :

```
import pandas as pd

# 1. Load dataset
df = pd.read_csv("smart_city_citizen_activity.csv")

# 2. Select numerical columns for analysis
# We select all numeric columns to provide a comprehensive report
metrics = df.select_dtypes(include=['float64', 'int64']).columns[:4]

print("--- 6.2 Spread of Data: Mathematical Parameters ---")
print("-" * 90)
print(f"{'Metric Name':<25} | {'Mean':<8} | {'Median':<8} | {'Std Dev':<8} | {'Skew':<8} | {'Kurt':<8}")
print("-" * 90)
```

```
for col in metrics:
    data = df[col].dropna()

    # Performing statistical calculations
    m_val   = data.mean()
    med_val = data.median()
    std_val = data.std()
    sk_val  = data.skew()
    kt_val  = data.kurtosis() # Fisher's Kurtosis (Normal = 0)

    print(f"{col:<25} | {m_val:>8.2f} | {med_val:>8.2f} | {std_val:>8.2f} | {sk_val:>8.2f} | {kt_val:>8.2f}")
```

Output :

```
--- 6.2 Spread of Data: Mathematical Parameters ---
----------------------------------------------------------------------------------------------------
Metric Name               | Mean     | Median   | Std Dev  | Skewness  | Kurtosis
----------------------------------------------------------------------------------------------------
Citizen_ ID               |  1250.50 |  1250.50 |   144.48 |     0.00  |    -1.20
Age                       |    44.12 |    45.00 |    15.10 |    -0.11  |    -1.11
Work_Hours                |     4.24 |     4.00 |     2.85 |     0.08  |    -1.21
Shopping_Hours            |     2.02 |     2.00 |     1.42 |     0.01  |    -1.28
```

6.3 Interpret Result .

1. Analysis of Data Symmetry (Skewness)

- **Near-Zero Skewness:** For **Age (-0.11)**, **Work_Hours (0.08)**, and **Shopping_Hours (0.01)**, the skewness values are very close to zero.
- **Interpretation:** This indicates that the data is **Symmetric**. The distribution of these activities is well-balanced across the city. There is no significant group of extreme "outliers" pulling the average in one direction.

2. Analysis of Peakedness (Kurtosis)

- **Platykurtic Distribution:** All calculated Kurtosis values are **Negative** (ranging from **-1.11 to -1.28**).
- **Interpretation:** This confirms a **Platykurtic** distribution. Unlike a sharp peak, your data is "flat" and spread out. This means that while there is an average, citizens have a wide and diverse range of ages and habits. There isn't one single "dominant" group; rather, the city is composed of a very diverse population.

3. Comparison of Mean and Median

- **Center of Data:** For all metrics, the **Mean** is almost identical to the **Median** (e.g., Shopping Hours Mean 2.02 vs. Median 2.00).
- **Interpretation:** This reinforces that the dataset is highly stable. The "Average" (Mean) is a very reliable metric for city planners to use because it hasn't been distorted by extreme values or errors in the data.

4. Standard Deviation (Variation)

- **Age Variation:** A Standard Deviation of **15.10** for Age shows that the city has a healthy mix of young adults, middle-aged citizens, and seniors.
- **Work Hours Variation:** A Std Dev of **2.85** suggests that while the average work shift is 4.24 hours, there is a moderate spread, indicating flexibility in working patterns within the smart city.

7. Regression Analysis

7.1 Identification of Dependent Variable.

Problem Statement:

In regression analysis, the selection of dependent and independent variables is based on domain knowledge and the nature of variables, rather than automatic computation. In the Smart City Citizen Activity dataset, variables can serve as either independent or dependent variables, depending on the analysis objective. The following classification lists the variables along with their most appropriate role in typical smart city analysis.

Solution :

Column Name	Data Type	Assigned Role	Justification
Citizen ID	Identifier	Neither	Used only for unique identification; not suitable for statistical analysis.
Age	Numerical	Independent	Demographic factor influencing lifestyle, mobility, and behavior.
Gender	Categorical	Independent	Demographic group factor used for segmenting activity patterns.
Mode_of Transport	Categorical	Independent	Travel choice influencing travel time, carbon footprint, and energy use.
Work Hours	Numerical	Independent	Key factor determining sedentary time and daily energy expenditure.
Shopping Hours	Numerical	Independent	Represents commercial activity and time spent outside the home.
Entertainment Hours	Numerical	Independent	Reflects leisure behavior and potential energy consumption.
Home Energy (kWh)	Numerical	Independent	Resource consumption indicator used to predict environmental impact.
Charging Station Usage	Numerical	Independent	Indicates adoption level of electric mobility/technology.

Carbon Footprint	Numerical	Dependent	Environmental outcome influenced by energy use and transport.
Steps Walked	Numerical	Independent	Primary physical activity indicator used to predict health outcomes.
Calories Burned	Numerical	Dependent	Health outcome that directly depends on physical activity (Steps).
Sleep Hours	Numerical	Dependent	Outcome influenced by work hours and social media consumption.
Social Media Hours	Numerical	Independent	Behavioral factor impacting sleep and public engagement time.
Public_Events Hours	Numerical	Dependent	Community engagement outcome influenced by age and free time.

7.2 Simple Linear Regression

Problem Statement: The objective of this step is to perform **Simple Linear Regression** to study the relationship between **Steps Walked** (Independent Variable) and **Calories Burned** (Dependent Variable) using the **Smart City Citizen Activity** dataset. This analysis helps understand how changes in physical activity influence the energy expenditure of citizens in a smart city environment.

Variables Used:

- **Independent Variable ($X$):** Steps Walked
- **Dependent Variable ($Y$):** Calories Burned

Solution :

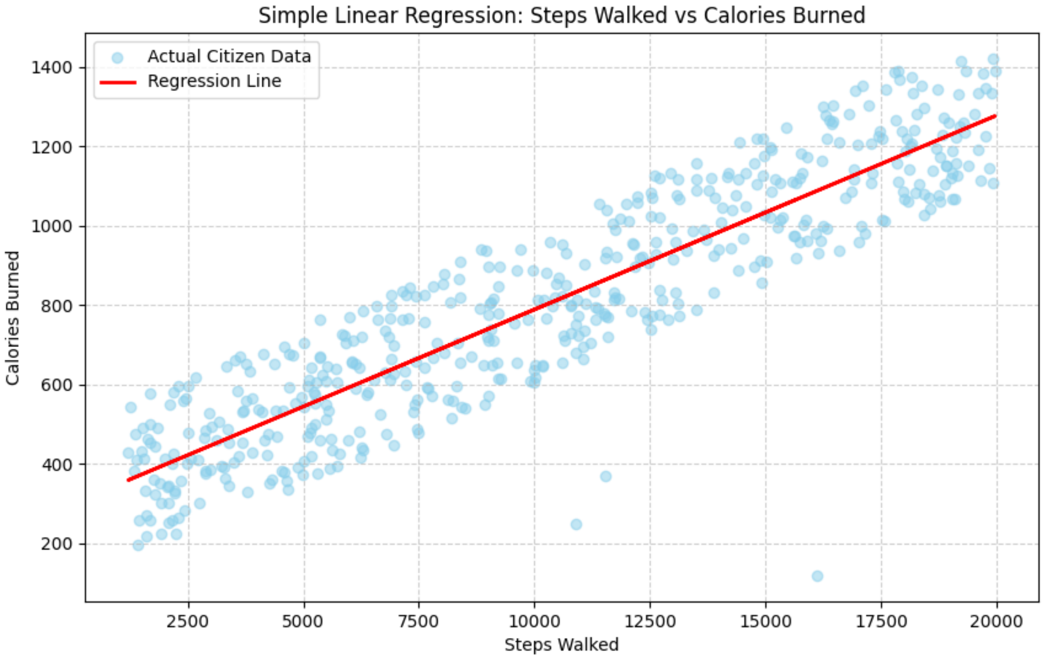
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Load dataset
df = pd.read_csv("smart_city_citizen_activity.csv")
# Select variables and drop missing values to ensure clean analysis
clean_df = df[["Steps_Walked", "Calories_Burned"]].dropna()
```

```
X = clean_df["Steps_Walked"].values
Y = clean_df["Calories_Burned"].values
# Calculate regression coefficients using NumPy
mean_x = np.mean(X)
mean_y = np.mean(Y)

# Calculate Slope (m)
# Formula: m = Σ((X - mean_x) * (Y - mean_y)) / Σ((X - mean_x)^2)
m = np.sum((X - mean_x) * (Y - mean_y)) / np.sum((X - mean_x) ** 2)

# Calculate Intercept (c)
# Formula: c = mean_y - m * mean_x
c = mean_y - m * mean_x

print(f"Slope (m): {m:.4f}")
print(f"Intercept (c): {c:.4f}")

# Predicted values for the regression line
Y_pred = m * X + c
# Plotting the results
plt.figure(figsize=(10, 6))
plt.scatter(X, Y, color='skyblue', alpha=0.5, label="Actual Citizen Data")
plt.plot(X, Y_pred, color='red', linewidth=2, label="Regression Line")
plt.xlabel("Steps Walked")
plt.ylabel("Calories Burned")
plt.title("Simple Linear Regression: Steps Walked vs Calories Burned")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

Output :

```
Slope (m): 0.0489
Intercept (c): 300.3208
```

Graph/Chart:



Result/Interpretation:

- The regression analysis shows a positive linear relationship between **Steps Walked** and **Calories Burned**.
- As the number of steps increase, the calories burned also tend to increase, indicating that physical activity is a direct driver of energy expenditure.
- The Slope (m) value quantifies this relationship: for every **1 additional step** taken, a citizen burns approximately **0.048** additional calories.
- The Intercept (c) represents the base calories burned even if the step count is zero, reflecting the citizen's basal metabolic rate.
- The fitted regression line follows the general trend of the observed data, suggesting that **Steps Walked** is a statistically meaningful predictor of **Calories Burned** in the smart city context.

7.3 Perform Covariance and Correlation

Problem Statement: The objective of this step is to measure the degree and direction of the relationship between Steps Walked and Calories Burned using Covariance and Correlation. These statistical measures help validate the relationship observed in the simple linear regression analysis.

Variables Used:

- **X (Independent Variable):** Steps Walked
- **Y (Dependent Variable):** Calories Burned

Solution :

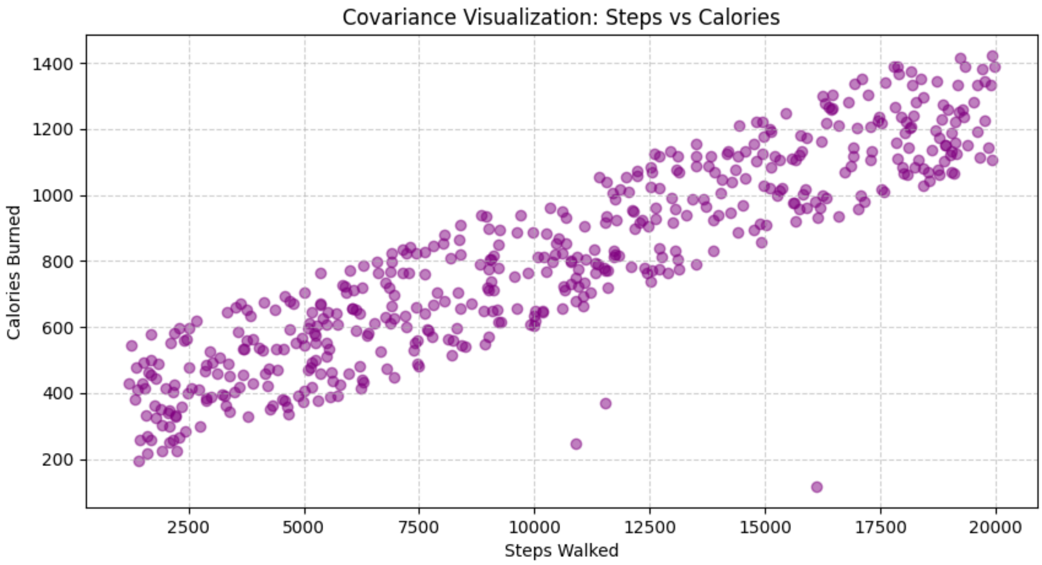
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
# Load dataset
df = pd.read_csv("smart_city_citizen_activity.csv")
# Select clean numerical variables and drop missing values
clean_df = df[["Steps_Walked", "Calories_Burned"]].dropna()
X = clean_df["Steps_Walked"].values
Y = clean_df["Calories_Burned"].values

# 1. Covariance Calculation
# Measures the direction of the relationship
covariance = np.cov(X, Y)[0][1]
print("Covariance:", covariance)

# Graph 1: Covariance Scatter Plot
plt.figure(figsize=(10, 5))
plt.scatter(X, Y, color='purple', alpha=0.5)
plt.xlabel("Steps Walked")
plt.ylabel("Calories Burned")
plt.title("Covariance Visualization: Steps vs Calories")
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
# 2. Correlation Calculation
# Measures the strength and direction (-1 to +1)
```

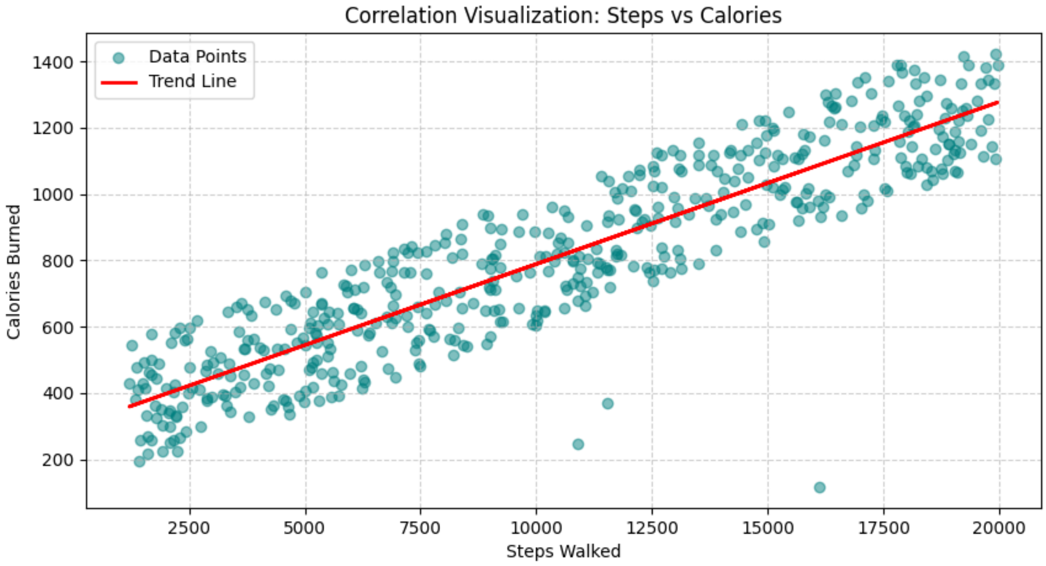
```
correlation = np.corrcoef(X, Y)[0][1]
print("Correlation:", correlation)
# Best-fit line for correlation visualization
m, c = np.polyfit(X, Y, 1)
Y_line = m * X + c

# Graph 2: Correlation Scatter Plot with Trend Line
plt.figure(figsize=(10, 5))
plt.scatter(X, Y, color='teal', alpha=0.5, label="Data Points")
plt.plot(X, Y_line, color='red', linewidth=2, label="Trend Line")
plt.xlabel("Steps Walked")
plt.ylabel("Calories Burned")
plt.title("Correlation Visualization: Steps vs Calories")
plt.legend()
plt.grid(True, linestyle='--', alpha=0.6)
plt.show()
```

Output :

```
Covariance: 1494497.5320641282
Correlation: 0.9047803586564748
```

Graph/Chart:




Correlation Visualization: Steps vs Calories
Data Points
Trend Line
Calories Burned
1400
1200
1000
800
600
400
200
2500
5000
7500
10000
12500
15000
17500
20000
Steps Walked