Introduction to the Dataset

The Energy_dataset.csv contains data related to energy consumption and various factors affecting it. The dataset includes the following columns:

1. **customer_id**: Unique identifier for each customer.
2. **energy_consumption_kwh**: Total energy consumption in kilowatt-hours.
3. **peak_hours_usage**: Energy usage during peak hours.
4. **off_peak_usage**: Energy usage during off-peak hours.
5. **renewable_energy_pct**: Percentage of energy from renewable sources.
6. **billing_amount**: Amount billed to the customer.
7. **household_size**: Number of people in the household.
8. **temperature_avg**: Average temperature in the region.
9. **smart_meter_installed**: Whether a smart meter is installed.
10. **time_of_day_pricing**: Whether time-of-day pricing is used.
11. **annual_energy_trend**: Annual trend in energy consumption.
12. **solar_panel**: Whether solar panels are installed.
13. **target_high_usage**: Whether the customer is targeted for high usage.

Manipulations Performed

Loading the Dataset:

df = pd.read_csv('Energy_dataset.csv')

Showing the Shape of the Dataset:

shape = df.shape

print(f"Shape of the dataset: {shape}")

This step displays the number of rows and columns in the dataset.

Describing Statistical Summaries:

```
statistical_summary = df.describe()

print("\nStatistical Summary:\n", statistical_summary)
```

This step provides statistical summaries such as count, mean, standard deviation, min, max, and quartiles for numeric columns.

Checking Skewness:

```
skewness = df.skew(numeric_only=True)

print("\nSkewness:\n", skewness)
```

This step calculates the skewness for numeric columns to understand the distribution of the data.

Detecting Outliers Using a Box Plot:

```
plt.figure(figsize=(15, 10))

sns.boxplot(data=df.select_dtypes(include=['float64', 'int64']))

plt.xticks(rotation=90)

plt.title('Box Plot for Outlier Detection')

plt.show()
```