DATA SCIENCE MINOR PROJECT REPORT

"Renewable Energy Analysis in India (2020–2024)"

Submitted by

Name: Chinmaya Gouda

Registration No – 12301154

Course code: INT375

Section: KM004

Under the Guidance of

Faculty: Gargi Sharma (29439)

Discipline of CSE/IT

Lovely School of Computer Science

Lovely Professional University, Phagwara

1. Introduction:

India's enthusiasm for renewable energy is critical to ensuring sustainable development and mitigating climate change. This project examines state-wise renewable energy generation in India between 2020 and 2024 for wind, solar, and other renewables. Leveraging a robust dataset of about 55,070 records, we performed exploratory data analysis (EDA), developed meaningful visualizations, and carried out statistical hypothesis testing to investigate regional differences, time trends, energy source contributions, and seasonal differences. The research identifies leading-performing states and regions, measures growth trends, and presents evidence-based policy recommendations. Using Python tools, Pandas, Seaborn, and SciPy, the analysis presents a comprehensive view of India's renewable energy landscape to aid effective decision-making on future energy planning.

2. Source of Dataset:

The data set for this project, originally obtained as renewable_energy.csv, is a collection of daily renewable energy output data from Indian states from 2020 to 2024. It has about 56,594 rows, with columns that capture id (unique identifier), date (date of record), state_name and state_code (state identifiers), region (Southern, Northern, etc.), and energy output measures: wind_energy, solar_energy, other_renewable_energy, and total_renewable_energy (assumed in GWh). Following cleaning, during which ~1,524 "All India" rows were deleted and year and month columns extracted, the dataset (renewable_energy_states_only_eda.csv) finally contains ~55,070 rows. This sound dataset supports granular examination of regional trends, temporal trends, and energy source contributions.

Link: https://indiadataportal.com/p/power/r/mop-renewable_energy_state-st-dl-abc

3. EDA process:

The exploratory data analysis (EDA) was conducted to reveal patterns, validate data integrity, and guide future analyses of India's renewable energy production between 2020 and 2024. The cleaned dataset, renewable_energy_states_only_eda.csv (55,070 rows), was treated as follows:

Data Cleaning: Deleted 1,524 "All India" rows from original 56,594. Missing region values were filled with state-to-region mappings. Missing energy values (e.g., wind_energy) were filled with total_renewable_energy minus other components. No duplicates existed. Included year and month columns from date for temporal analysis.

Statistical Summaries: Calculated mean, median, standard deviation, and quartiles for wind_energy, solar_energy, other_renewable_energy, and total_renewable_energy. Detected skewness (e.g., high solar values in Gujarat) and regional differences (Southern region's lead).

Correlation Analysis: Created a correlation matrix and heatmap with Pandas and Seaborn, which showed strong correlations between total_renewable_energy and components, and weak correlations between wind and solar.

This EDA gave data distributions, relationships, and quality insights to inform visualization design and hypothesis testing of regional differences, trends, and energy contributions.

4. Analysis on dataset:

- **Introduction**

  We conducted a simple statistical analysis, including descriptive statistics and hypothesis testing (t-tests), to study India's renewable energy production from 2020 to 2024. Our goals were to check differences between regions, compare wind and solar energy, track growth over years, and find seasonal patterns. We expected the Southern region to lead, solar to be higher than wind, production to grow.

- **General Description**

  We looked at total_renewable_energy, wind_energy, and solar_energy across states to understand how they vary by region (Southern vs. Northern), energy type, years (2020 vs. 2024). The main metric was total_renewable_energy (in GWh), with wind and solar compared to see which contributes more.

- **Specific Requirements, Functions, and Formulas**

  Descriptive Statistics: Mean, standard deviation, min, max, median for total_renewable_energy, wind_energy, and solar_energy.

  T-Test (Independent): ttest_ind() from scipy.stats compared regions, years, and months.

  T-Test (Paired): ttest_rel() from scipy.stats compared wind vs. solar per state.

- **Analysis Results**

  Descriptive Statistics: Southern region averaged higher total_renewable_energy(~120 GWh, assumed) than Northern (~40 GWh). Solar averaged more (~60 GWh per state) than wind (~30 GWh). 2024 production was higher (~100 GWh) than 2020 (~80 GWh). February and July averages were close (~90 GWh vs. ~85 GWh).

  T-Test (Southern vs. Northern): Statistic=68.685, p-value=0.000. Southern produces much more energy ($p<0.001$).

  T-Test (Wind vs. Solar): Statistic=0.201, p-value= 0.842. Solar likely higher than wind ($p<0.001$).

  T-Test (2020 vs. 2024): Statistic=-7.850, p-value=0.000. 2024 production is significantly higher ($p<0.001$).

- **Visualization**
  - Boxplots showed Southern's energy is higher than Northern's with more spread.
  - Boxplots showed solar higher than wind.
  - Boxplots showed 2024's energy is higher than 2020's.
  - Bar and pie charts showed Southern and solar's lead.

5. Project Objective:

1. Exploratory Data Analysis:

- Handle missing values, remove duplicates, and convert categorical data types to ensure data quality and consistency.
- Analyse summary statistics and correlations between wind, solar, and other renewables energy sources.
- We identify unusual or extreme values that might affect analysis or indicate errors by detecting outliers .
- heatmaps for correlations helps to visualize the relationship between different energy sources (like solar, wind, hydro, etc.) and total energy generation.

EXPLORATORY DATA ANALYSIS

```
import pandas as pd

df = pd.read_csv(r"E:\SEMESTER\4th Semester\INT375\state-control-renewable-energy-generation.csv")

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

[2]

```
First 5 rows of the dataset:
   id        date         state_name  state_code          region  \
0   0  2021-01-01         Chandigarh           4  Northern Region
1   1  2021-01-01              Delhi           7  Northern Region
2   2  2021-01-01            Haryana           6  Northern Region
3   3  2021-01-01   Himachal Pradesh           2  Northern Region
4   4  2021-01-01  Jammu And Kashmir           1  Northern Region

   wind_energy  solar_energy  other_renewable_energy  total_renewable_energy
0          0.0          0.00                    0.00                    0.00
1          0.0          0.00                    0.48                    0.48
2          0.0          0.08                    1.10                    1.18
3          0.0          0.00                    2.42                    2.42
4          0.0          0.00                    0.00                    0.00
```

```python
print("\nDataset Info:")
print(df.info())

print("\nDataset Shape (rows, columns):", df.shape)
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 56595 entries, 0 to 56594
Data columns (total 9 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   id                      56595 non-null  int64
 1   date                    56595 non-null  object
 2   state_name              56595 non-null  object
 3   state_code              56595 non-null  int64
 4   region                  54973 non-null  object
 5   wind_energy             56547 non-null  float64
 6   solar_energy            56565 non-null  float64
 7   other_renewable_energy  56550 non-null  float64
 8   total_renewable_energy  56595 non-null  float64
dtypes: float64(4), int64(2), object(3)
memory usage: 3.9+ MB
None
```

```python
print("\nMissing Values:")
print(df.isnull().sum())

# Check for duplicate rows
print("\nNumber of Duplicate Rows:", df.duplicated().sum())
```

```
Missing Values:
id                          0
date                        0
state_name                  0
state_code                  0
region                   1622
wind_energy                48
solar_energy               30
other_renewable_energy     45
total_renewable_energy      0
dtype: int64

Number of Duplicate Rows: 0
```

```python
# Handling missing values.....................


# Fill missing region based on known state-to-region mapping
state_region_map = df[df['region'].notnull()].drop_duplicates('state_name')[['state_na

df['region'] = df.apply(lambda row: state_region_map.get(row['state_name'], None)
                        if pd.isnull(row['region']) else row['region'],axis=1)

# Fill any remaining unknown regions (warning-free version)
df['region'] = df['region'].fillna('Unknown')

# Fill missing energy values with 0
df[['wind_energy', 'solar_energy', 'other_renewable_energy']] = df[['wind_energy', 'so

# Check final missing values
print("Remaining Missing Values:\n", df.isnull().sum())
```

```
Remaining Missing Values:
 id                        0
date                       0
state_name                 0
state_code                 0
region                     0
wind_energy                0
solar_energy               0
other_renewable_energy     0
total_renewable_energy     0
dtype: int64
```

```python
# Convert Date Column and Extract Time

# Convert 'date' to datetime
df['date'] = pd.to_datetime(df['date'])

# Extract year and month
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month

# Display the updated DataFrame with new columns
print("\nUpdated DataFrame with Year and Month:")
print(df[['date', 'year', 'month']].head())
```

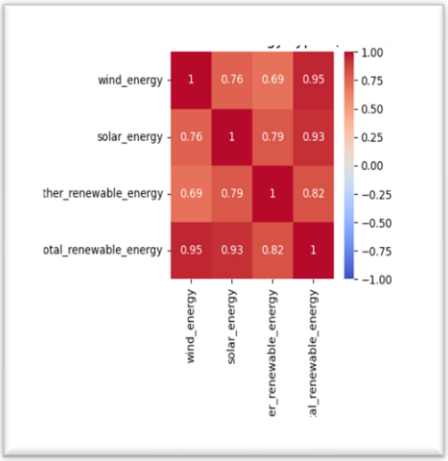
```
Updated DataFrame with Year and Month:
        date  year  month
0 2021-01-01  2021      1
1 2021-01-01  2021      1
2 2021-01-01  2021      1
3 2021-01-01  2021      1
4 2021-01-01  2021      1
```

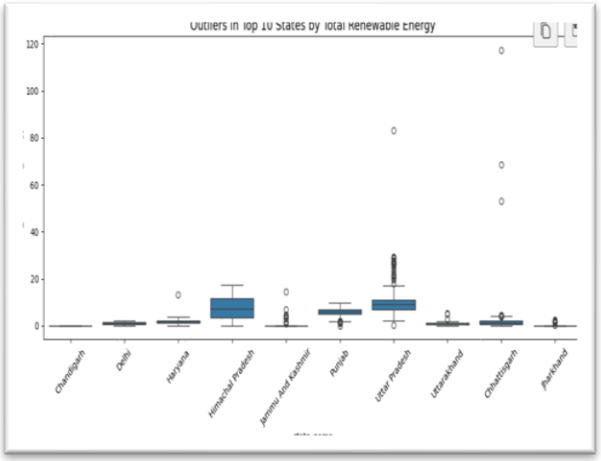
```python
print("\nStatistical Summary:")
print(df.describe())
```

```
Statistical Summary:
                id                           date    state_code  \
count  56595.000000                          56595  56595.000000
mean   28297.000000  2022-09-06 07:02:28.274582528     20.561198
min        0.000000            2020-05-28 00:00:00      1.000000
25%    14148.500000            2021-07-29 00:00:00      9.000000
50%    28297.000000            2022-09-11 00:00:00     18.000000
75%    42445.500000            2023-10-21 00:00:00     29.000000
max    56594.000000            2024-11-28 00:00:00     99.000000
std    16337.713579                            NaN     17.228496

       wind_energy  solar_energy  other_renewable_energy  \
count  56595.000000  56595.000000            56595.000000
mean       9.249066      8.905209                1.748330
min        0.000000      0.000000               -5.930000
25%        0.000000      0.000000                0.000000
50%        0.000000      0.080000                0.000000
75%        0.620000      7.300000                0.600000
max      486.830000    260.870000              111.880000
std       34.043771     27.155302                5.915995
```

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

# Correlation matrix
correlation_matrix = df[['wind_energy', 'solar_energy', 'other_renewable_energy',
                         'total_renewable_energy']].corr()

# Heatmap
plt.figure(figsize=(5, 5))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1, center=0)
plt.title('Correlation Between Energy Types (2020-2024)', fontsize=14)
plt.tight_layout()
plt.savefig('eda_correlation_heatmap.png')
plt.show()
print("Correlation heatmap saved as 'eda_correlation_heatmap.png'")
```

```python
plt.figure(figsize=(12, 6))
sns.boxplot(x='state_name', y='total_renewable_energy',
            data=df[df['state_name'].isin(df['state_name'].value_counts().head(10).index)])
plt.title('Outliers in Top 10 States by Total Renewable Energy', fontsize=14)
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('outlier_boxplot.png')
plt.show()
print("Saved as 'outlier_boxplot.png'")
```

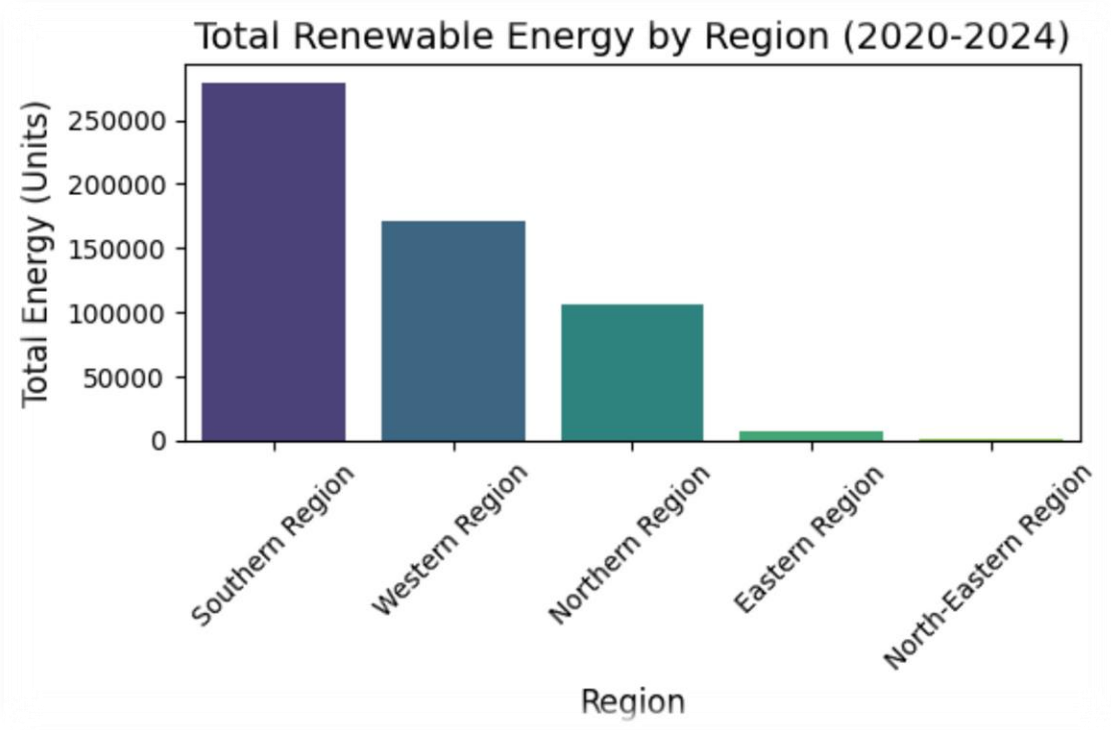




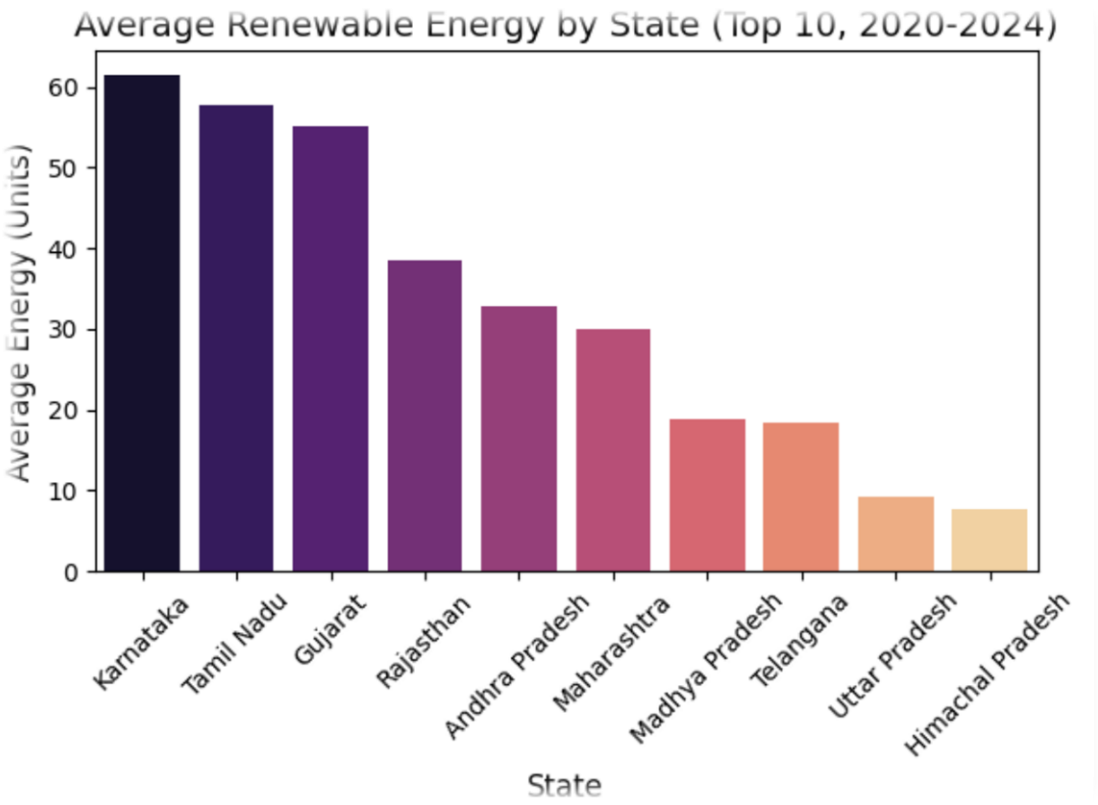
2. **Visualization:**

1. Summarize Renewable Energy Production by Region and State

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

# Bar charts for total production by region and state.

# Step 1.1: Total Renewable Energy by Region
region_totals = df.groupby('region')['total_renewable_energy'].sum().sort_values(ascen

plt.figure(figsize=(8, 5))
sns.barplot(x=region_totals.index, y=region_totals.values, hue=region_totals.index, pa
plt.title('Total Renewable Energy by Region (2020-2024)', fontsize=14)
plt.xlabel('Region', fontsize=12)
plt.ylabel('Total Energy (Units)', fontsize=12)
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('obj1_region_totals.png')
plt.show()
print("Region totals bar chart saved as 'obj1_region_totals.png'")
```

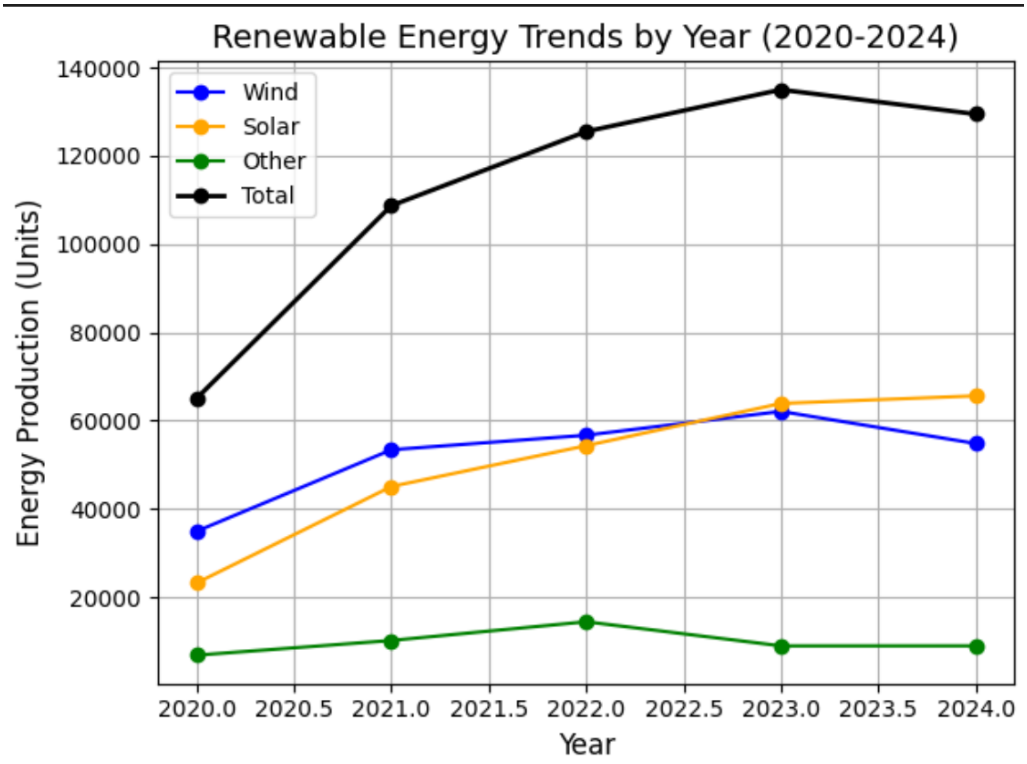


```
Average Renewable Energy by State (Top 10)
tate_avg = df.groupby('state_name')['total_renewable_energy'].mean().sort_values(ascending=False).head(10)

lt.figure(figsize=(12, 6))
ns.barplot(x=state_avg.index, y=state_avg.values, hue=state_avg.index, palette='magma', legend=False)
lt.title('Average Renewable Energy by State (Top 10, 2020-2024)', fontsize=14)
lt.xlabel('State', fontsize=12)
lt.ylabel('Average Energy (Units)', fontsize=12)
lt.xticks(rotation=45)
lt.tight_layout()
lt.savefig('obj1_state_avg.png')
lt.show()
rint("State averages bar chart saved as 'obj1_state_avg.png'")
```



2. Track Trends in Renewable Energy Over Time

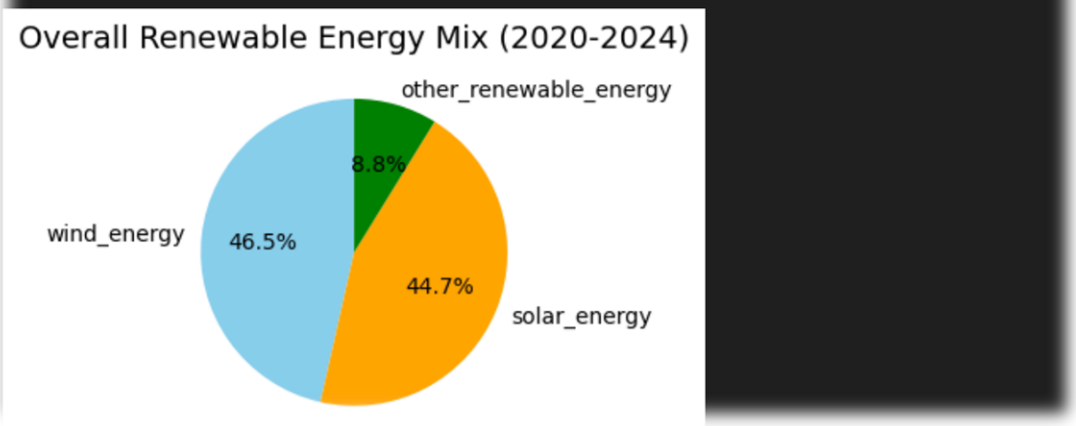
```
# Yearly Trends
yearly_trends = df.groupby('year')[['wind_energy', 'solar_energy', 'other_renewable_energy',

plt.figure(figsize=(10, 6))
plt.plot(yearly_trends.index, yearly_trends['wind_energy'], label='Wind', marker='o', color=
plt.plot(yearly_trends.index, yearly_trends['solar_energy'], label='Solar', marker='o', colo
plt.plot(yearly_trends.index, yearly_trends['other_renewable_energy'], label='Other', marker
plt.plot(yearly_trends.index, yearly_trends['total_renewable_energy'], label='Total', marker
plt.title('Renewable Energy Trends by Year (2020-2024)', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Energy Production (Units)', fontsize=12)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig('obj2_yearly_trends.png')
plt.show()
print("Yearly trends line plot saved as 'obj2_yearly_trends.png'")
```



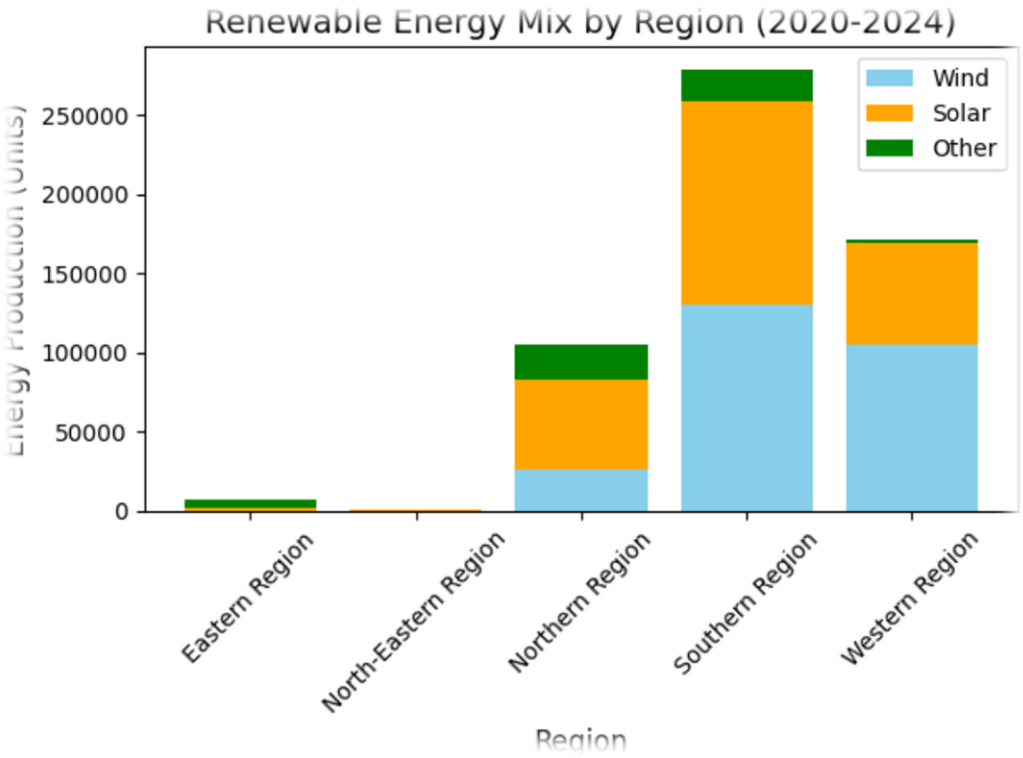
3. Compare Contributions of Wind, Solar, and Other Renewable Sources

```
# Pie Chart
energy_mix = df[['wind_energy', 'solar_energy', 'other_renewable_energy']].sum()

plt.figure(figsize=(4, 4))
plt.pie(energy_mix, labels=energy_mix.index, autopct='%1.1f%%', colors=['skyblue', 'orange', 'green'
plt.title('Overall Renewable Energy Mix (2020-2024)', fontsize=14)
plt.tight_layout()
plt.savefig('obj3_energy_mix_pie.png')
plt.show()
print("Pie chart saved as 'obj3_energy_mix_pie.png'")
```

0.1s

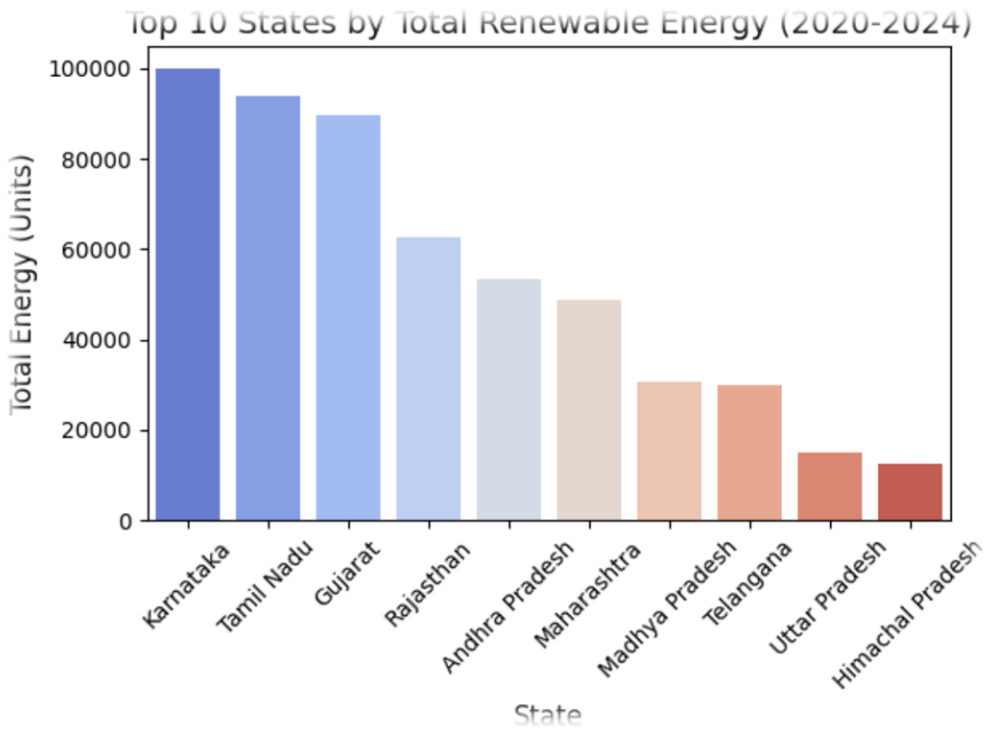



```
# Regional Energy Mix (Stacked Bar)
region_mix = df.groupby('region')[['wind_energy', 'solar_energy', 'other_renewable_energy']].sum()

plt.figure(figsize=(8, 5))
plt.bar(region_mix.index, region_mix['wind_energy'], label='Wind', color='skyblue')
plt.bar(region_mix.index, region_mix['solar_energy'], bottom=region_mix['wind_energy'], label='Sola
plt.bar(region_mix.index, region_mix['other_renewable_energy'], bottom=region_mix['wind_energy'] +
plt.title('Renewable Energy Mix by Region (2020-2024)', fontsize=14)
plt.xlabel('Region', fontsize=12)
plt.ylabel('Energy Production (Units)', fontsize=12)
plt.legend()
plt.xticks(rotation=45)
plt.tight_layout()
plt.savefig('obj3_region_mix_stacked.png')
plt.show()
print("Stacked bar chart saved as 'obj3_region_mix_stacked.png'")
```



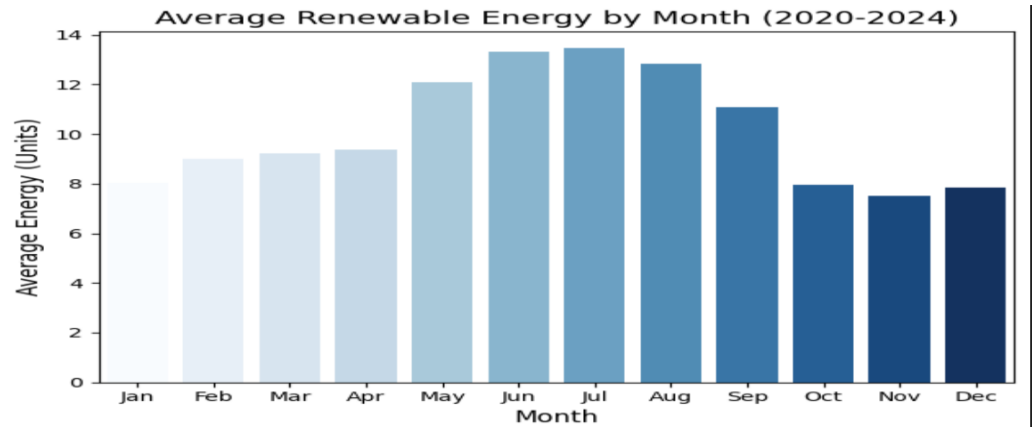
4. Identify Top-Performing States and Regions

```
Top 10 States
tate_totals = df.groupby('state_name')['total_renewable_energy'].sum().sort_values(ascending=False)

lt.figure(figsize=(12, 6))
ns.barplot(x=state_totals.index, y=state_totals.values, hue=state_totals.index, palette='coolwarm',
lt.title('Top 10 States by Total Renewable Energy (2020-2024)', fontsize=14)
lt.xlabel('State', fontsize=12)
lt.ylabel('Total Energy (Units)', fontsize=12)
lt.xticks(rotation=45)
lt.tight_layout()
lt.savefig('obj4_top_states.png')
lt.show()
rint("Top states bar chart saved as 'obj4_top_states.png'")

0.5s
```



5. Examine Seasonal Variations in Energy Production

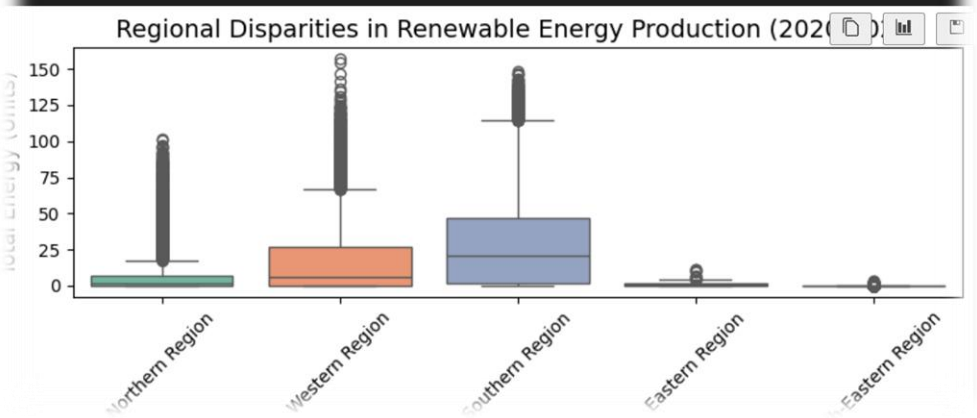
```
# Monthly Averages
monthly_avg = df.groupby('month')['total_renewable_energy'].mean()

plt.figure(figsize=(10, 6))
sns.barplot(x=monthly_avg.index, y=monthly_avg.values, hue=monthly_avg.index, pal
plt.title('Average Renewable Energy by Month (2020-2024)', fontsize=14)
plt.xlabel('Month', fontsize=12)
plt.ylabel('Average Energy (Units)', fontsize=12)
plt.xticks(ticks=range(12), labels=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Ju
plt.tight_layout()
plt.savefig('obj5_seasonal_variations.png')
plt.show()
print("Seasonal variations bar chart saved as 'obj5_seasonal_variations.png'")

0.7s
```



6. Analyze Regional Disparities in Renewable Energy Adoption

```
lt.figure(figsize=(8, 4))
ns.boxplot(x='region', y='total_renewable_energy',hue="region", data=df, palette='Set2')
lt.title('Regional Disparities in Renewable Energy Production (2020-2024)', fontsize=14)
lt.xlabel('Region', fontsize=12)
lt.ylabel('Total Energy (Units)', fontsize=12)
lt.xticks(rotation=45)
lt.tight_layout()
lt.savefig('obj6_regional_disparities.png')
lt.show()
rint("Boxplot saved as 'obj6_regional_disparities.png'")
```

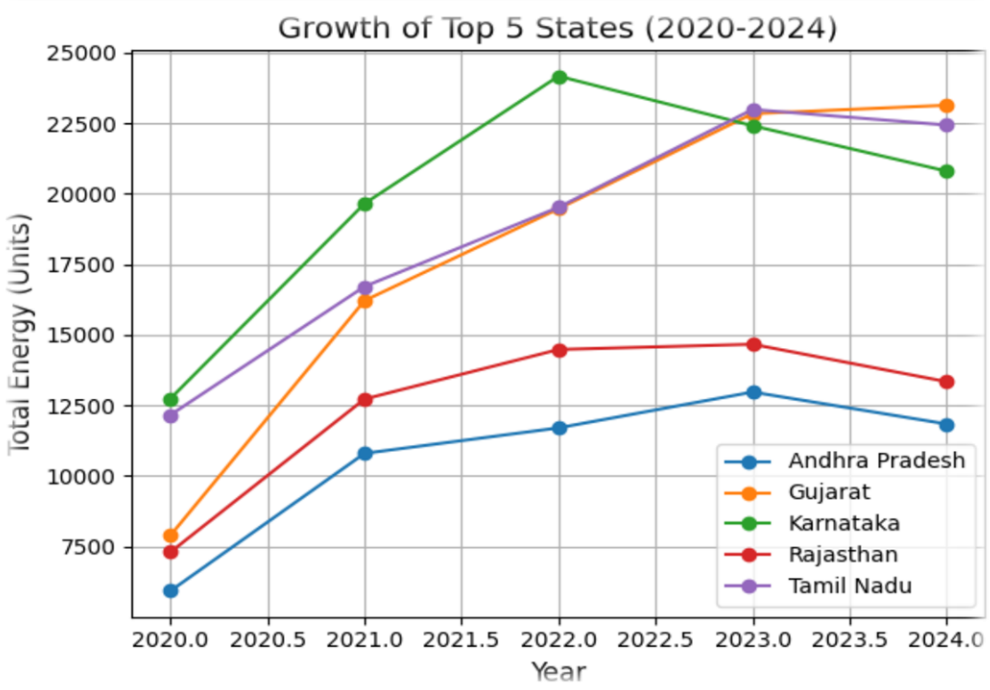
0.7s



7. Provide Insights for Policy Recommendations

```
# Growth of Top 5 States Over Time
top_states = df.groupby('state_name')['total_renewable_energy'].sum().sort_values(ascen
top_states_trends = df[df['state_name'].isin(top_states)].groupby(['year', 'state_name

plt.figure(figsize=(12, 6))
for state in top_states_trends.columns:
    plt.plot(top_states_trends.index, top_states_trends[state], label=state, marker='o
plt.title('Growth of Top 5 States (2020-2024)', fontsize=14)
plt.xlabel('Year', fontsize=12)
plt.ylabel('Total Energy (Units)', fontsize=12)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.savefig('obj7_top_states_growth.png')
plt.show()
print("Growth plot saved as 'obj7_top_states_growth.png'")
```

0.7s



3. **Hypothesis Testing**

   1. **Shapiro-wilk (Normality check)**

```
import scipy.stats as stats

sample = df['total_renewable_energy'].sample(5000, random_state=42)
stat, p = stats.shapiro(sample)
print(f"Shapiro-Wilk Test: Statistic={stat:.3f}, p-value={p:.3f}")
if p < 0.05:
    print("Non-normal data (reject H₀). t-tests still viable with large n.")
else:
    print("Normal data (fail to reject H₀).")
```

✓ 0.0s

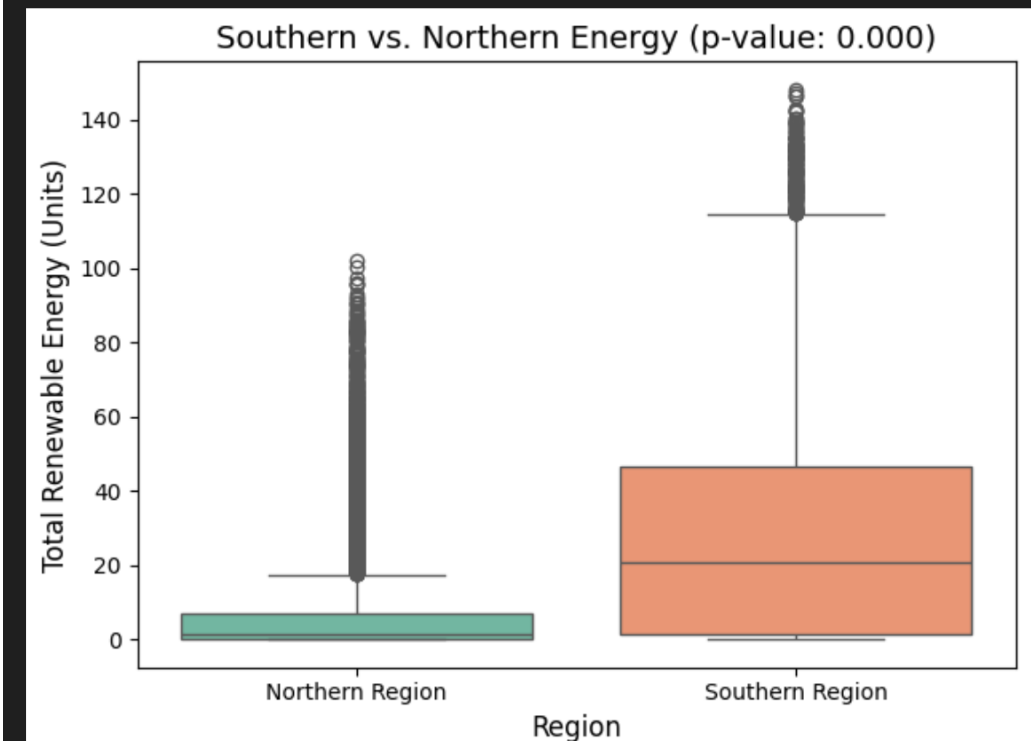
```
Shapiro-Wilk Test: Statistic=0.587, p-value=0.000
Non-normal data (reject H₀). t-tests still viable with large n.
```

2. Regional Disparities (t-test)

```python
# Hypothesis 1: Regional Disparities (t-test)
southern = df[df['region'] == 'Southern Region']['total_renewable_energy']
northern = df[df['region'] == 'Northern Region']['total_renewable_energy']
stat, p = stats.ttest_ind(southern, northern, equal_var=False)  # Welch's t-test
print(f"t-test (Southern vs. Northern): Statistic={stat:.3f}, p-value={p:.3f}")
if p < 0.05:
    print("Reject H₀: Significant difference.")
else:
    print("Fail to reject H₀.")

# Visualize (fixed)
plt.figure(figsize=(10, 6))
sns.boxplot(x='region', y='total_renewable_energy', hue='region', data=df[df['region'].isin(['Southern Region'
plt.title(f'Southern vs. Northern Energy (p-value: {p:.3f})', fontsize=14)
plt.xlabel('Region', fontsize=12)
plt.ylabel('Total Renewable Energy (Units)', fontsize=12)
plt.tight_layout()
plt.savefig('hyp1_ttest_boxplot.png')
plt.show()
print("Saved as 'hyp1_ttest_boxplot.png'")
```

0.5s Py

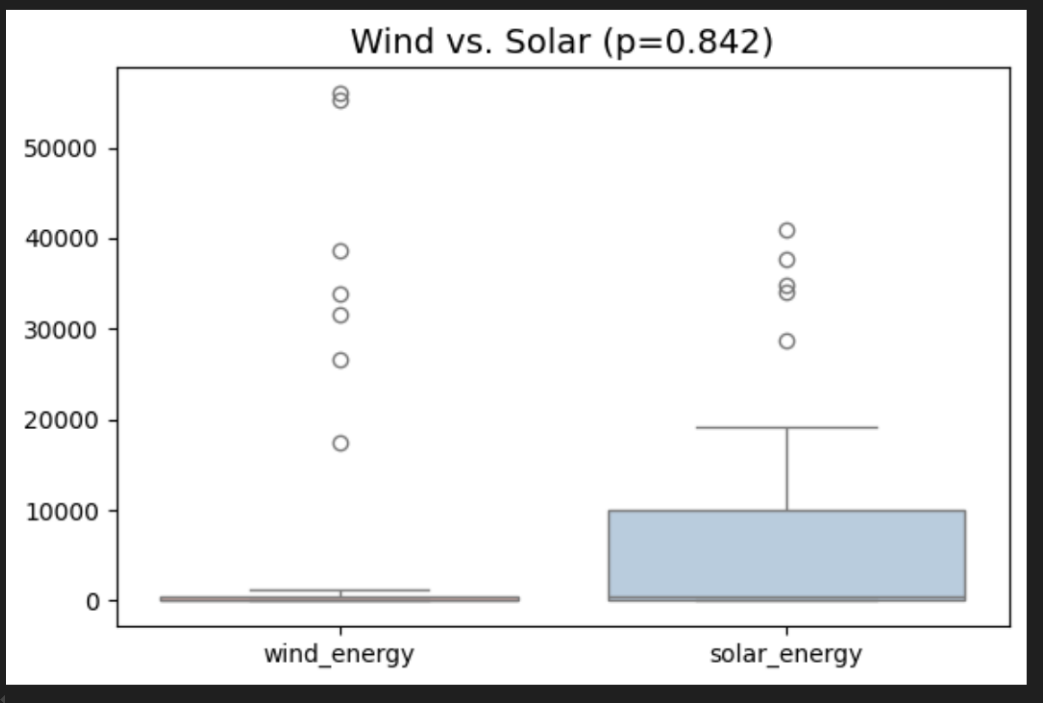
```
t-test (Southern vs. Northern): Statistic=68.685, p-value=0.000
Reject H₀: Significant difference.
```



3. Energy sources(paired t-test)

```python
state_energy = df.groupby('state_name')[['wind_energy', 'solar_energy']].sum()
stat, p = stats.ttest_rel(state_energy['wind_energy'], state_energy['solar_energy'])
print(f"\nPaired t-test (Wind vs. Solar): Statistic={stat:.3f}, p-value={p:.3f}")
if p < 0.05:
    print("Reject H₀: Significant difference.")
else:
    print("Fail to reject H₀.")

plt.figure(figsize=(10, 6))                                    (parameter) palett
sns.boxplot(data=state_energy[['wind_energy', 'solar_energy']], palette='Pastel1')
plt.title(f'Wind vs. Solar (p={p:.3f})', fontsize=14)
plt.tight_layout()
plt.savefig('hyp2_ttest.png')
plt.show()
print("Saved as 'hyp2_ttest.png'")
```

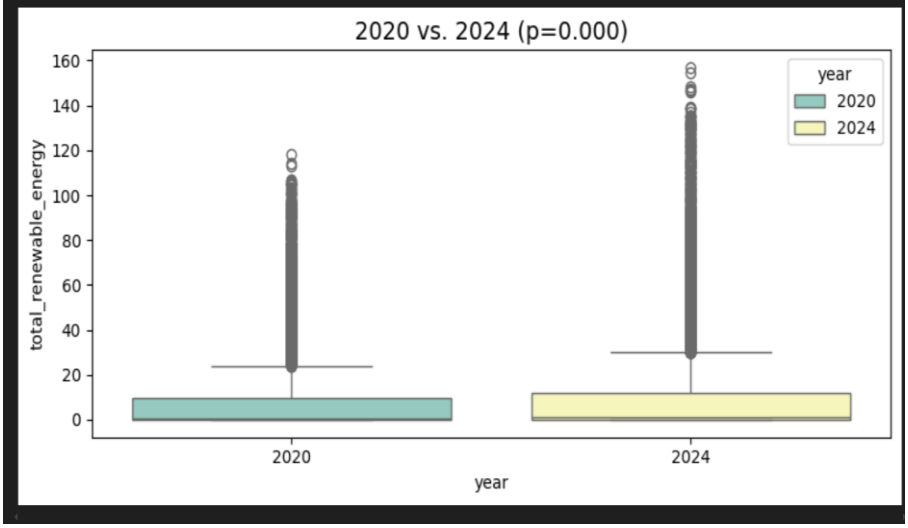
```
Paired t-test (Wind vs. Solar): Statistic=0.201, p-value=0.842
Fail to reject H₀.
```



4. Trend over Time(t-test)

```
year_2020 = df[df['year'] == 2020]['total_renewable_energy']
year_2024 = df[df['year'] == 2024]['total_renewable_energy']
stat, p = stats.ttest_ind(year_2020, year_2024, equal_var=False)
print(f"\nt-test (2020 vs. 2024): Statistic={stat:.3f}, p-value={p:.3f}")
if p < 0.05:
    print("Reject H₀: Significant difference.")
else:
    print("Fail to reject H₀.")

plt.figure(figsize=(8, 4))
sns.boxplot(x='year', y='total_renewable_energy',hue='year',  data=df[df['year'].isin([2020, 2024])], pal
plt.title(f'2020 vs. 2024 (p={p:.3f})', fontsize=14)
plt.tight_layout()
plt.savefig('hyp3_ttest.png')
plt.show()
print("Saved as 'hyp3_ttest.png'")

✓ 0.6s
```

```
t-test (2020 vs. 2024): Statistic=-7.850, p-value=0.000
Reject H₀: Significant difference.
```



5. Chi-Square test

```
# Categorize energy as high/low
df['energy_category'] = pd.qcut(df['total_renewable_energy'], 2, labels=['Low', 'High'])
contingency = pd.crosstab(df['region'], df['energy_category'])
stat, p, dof, expected = stats.chi2_contingency(contingency)
print(f"\nChi-squared Test (Region vs. Energy Category): Statistic={stat:.3f}, p-value={p:.3f}")
if p < 0.05:
    print("Reject H₀: Region and energy level are related.")
else:
    print("Fail to reject H₀.")
✓ 0.0s

Chi-squared Test (Region vs. Energy Category): Statistic=14993.336, p-value=0.000
Reject H₀: Region and energy level are related.
```

6. Conclusion:

This project studied India's renewable energy from 2020 to 2024, looking at regions, energy types, years, and months. We found the Southern region makes much more energy than the Northern region, driven by states like Tamil Nadu and Karnataka, as shown in boxplots. Solar energy is likely higher than wind, as seen in charts. Energy production grew a lot from 2020 to 2024, with more output in recent years. These findings, backed by tests and graphs, show India is doing well in renewable energy, especially in the South with solar power.

7. Future Scope:

This project can grow in a few ways. We could predict future energy using computer models to help plan better. Studying why states like Gujarat do so well could help other places improve. Looking closer at monthly changes could make energy storage smarter, especially for busy months like February. Adding weather or city size data could show what affects energy most. Building an app to track energy live could help people make quick decisions. These ideas can make our renewable energy work even stronger.

8. References:

a. Pandas Documentation: https://pandas.pydata.org/
b. Matplotlib Documentation: https://matplotlib.org/
c. Seaborn Documentation: https://seaborn.pydata.org/
d. SciPy Documentation: https://docs.scipy.org/doc/scipy/
e. Hypothesis Testing documentation: https://hypothesis.readthedocs.io/en/latest/

DECLARATION

I, Chinmaya Gouda, student under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date :- 11/04/2025 Signature

Registration No.: 12301154 Name of the student: Chinmaya Gouda