02_feature_analysis

November 24, 2024

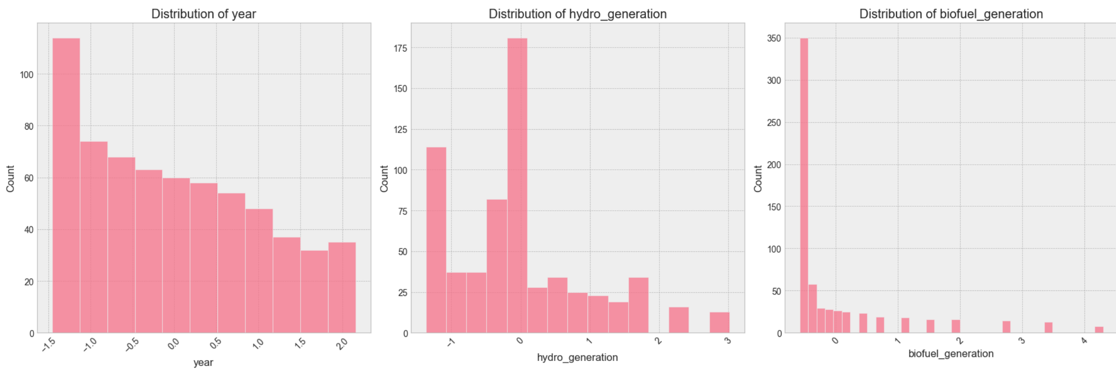
[1]:
```python
import warnings
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import seaborn as sns
from scipy import stats
from sklearn.decomposition import PCA
from sklearn.feature_selection import mutual_info_regression
from sklearn.preprocessing import StandardScaler

warnings.filterwarnings('ignore')

# Set plotting styles
plt.style.use('bmh')
sns.set_palette("husl")
plt.rcParams['figure.figsize'] = [12, 6]
```

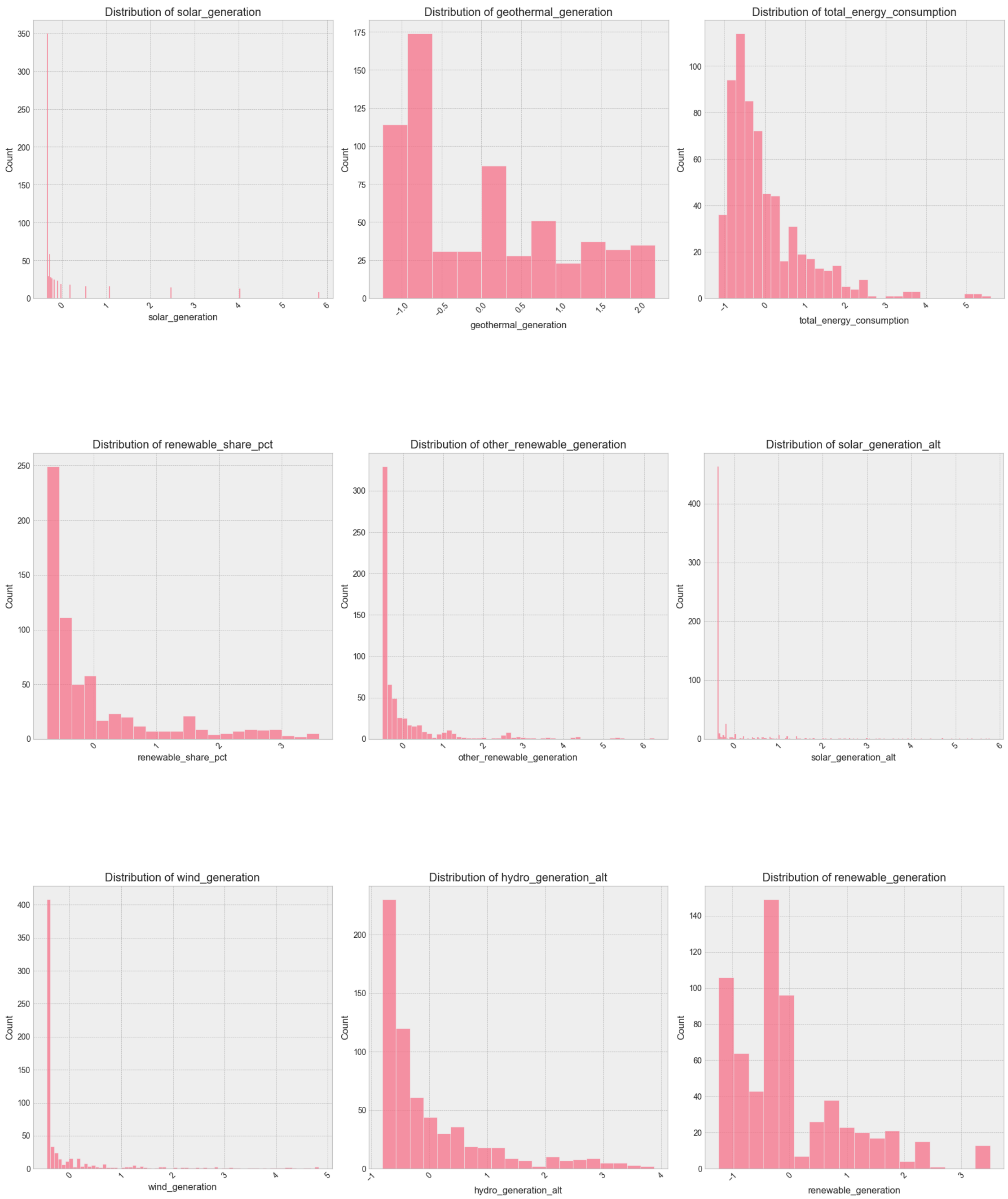
[2]:
```python
# Load Processed Data from the Pipeline

# Get the current notebook directory and construct the correct path
notebook_dir = Path().absolute()
project_root = notebook_dir.parent if notebook_dir.name == 'notebooks' else␣
↪notebook_dir
processed_data_path = project_root / 'processed_data' / 'final_processed_data.
↪csv'

print(f"Looking for data file at: {processed_data_path}")
df = pd.read_csv(processed_data_path)

# Display basic information about the processed dataset
print("Dataset Overview:")
print("=" * 80)
print(f"\nShape: {df.shape}")
```

```python
print("\nFeatures:")
for col in df.columns:
    dtype = df[col].dtype
    missing = df[col].isnull().sum()
    print(f"- {col}: {dtype} (Missing: {missing})")
```

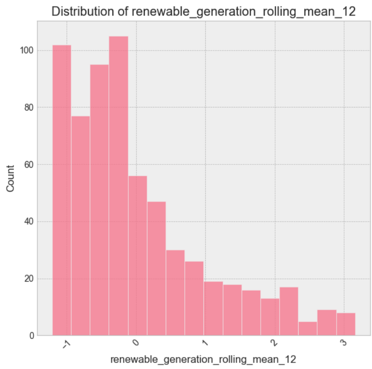
```
Looking for data file at:
/Users/katejohnson/Documents/Other/Northeastern/CS6140/Course
Project/cs6140-course-project/processed_data/final_processed_data.csv
Dataset Overview:
================================================================================

Shape: (643, 31)

Features:
- year: float64 (Missing: 0)
- hydro_generation: float64 (Missing: 0)
- biofuel_generation: float64 (Missing: 0)
- solar_generation: float64 (Missing: 0)
- geothermal_generation: float64 (Missing: 0)
- country: object (Missing: 0)
- total_energy_consumption: float64 (Missing: 0)
- renewable_share_pct: float64 (Missing: 0)
- other_renewable_generation: float64 (Missing: 0)
- solar_generation_alt: float64 (Missing: 0)
- wind_generation: float64 (Missing: 0)
- hydro_generation_alt: float64 (Missing: 0)
- renewable_generation: float64 (Missing: 0)
- decade: float64 (Missing: 0)
- period: object (Missing: 0)
- renewable_generation_lag_1: float64 (Missing: 38)
- renewable_generation_lag_3: float64 (Missing: 114)
- renewable_generation_lag_6: float64 (Missing: 223)
- renewable_generation_lag_12: float64 (Missing: 408)
- renewable_generation_rolling_mean_3: float64 (Missing: 0)
- renewable_generation_rolling_std_3: float64 (Missing: 38)
- renewable_generation_rolling_mean_6: float64 (Missing: 0)
- renewable_generation_rolling_std_6: float64 (Missing: 38)
- renewable_generation_rolling_mean_12: float64 (Missing: 0)
- renewable_generation_rolling_std_12: float64 (Missing: 38)
- total_renewable: float64 (Missing: 0)
- renewable_share: float64 (Missing: 0)
- hydro_generation_share: float64 (Missing: 0)
- solar_generation_share: float64 (Missing: 0)
- wind_generation_share: float64 (Missing: 0)
- renewable_yoy_growth: float64 (Missing: 38)
```

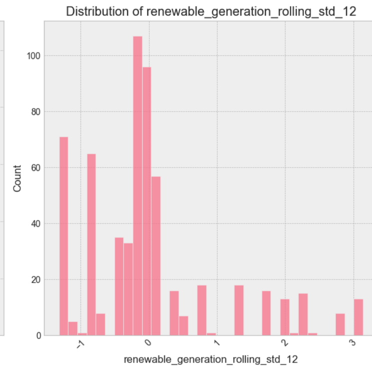
[3]:
```
# Feature Distribution Analysis
def analyze_feature_distributions():
    """Analyze the distribution of engineered features"""

    # Select numerical columns
    numeric_cols = df.select_dtypes(include=[np.number]).columns

    # Create distribution plots
    for i in range(0, len(numeric_cols), 3):
        cols = numeric_cols[i:i + 3]
        fig, axes = plt.subplots(1, len(cols), figsize=(18, 6))
        if len(cols) == 1:
            axes = [axes]

        for ax, col in zip(axes, cols):
            sns.histplot(data=df, x=col, ax=ax)
            ax.set_title(f'Distribution of {col}')
            ax.tick_params(axis='x', rotation=45)

        plt.tight_layout()
        plt.show()

    # Test for normality
    normality_tests = {}
    for col in numeric_cols:
        stat, p_value = stats.normaltest(df[col].dropna())
        normality_tests[col] = {'statistic': stat, 'p_value': p_value}

    return pd.DataFrame(normality_tests).T


# Run distribution analysis
distribution_results = analyze_feature_distributions()
print("\nNormality Test Results:")
display(distribution_results)
```

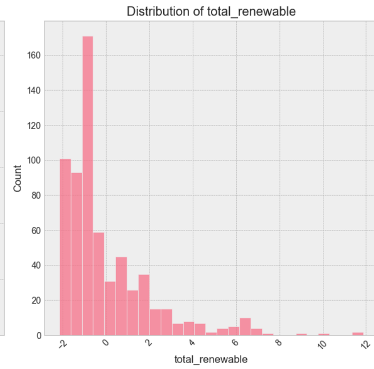


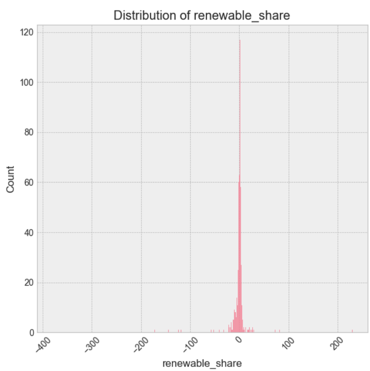

Distribution of solar_generation
Count
solar_generation
Distribution of geothermal_generation
Count
geothermal_generation
Distribution of total_energy_consumption
Count
total_energy_consumption
Distribution of renewable_share_pct
Count
renewable_share_pct
Distribution of other_renewable_generation
Count
other_renewable_generation
Distribution of solar_generation_alt
Count
solar_generation_alt
Distribution of wind_generation
Count
wind_generation
Distribution of hydro_generation_alt
Count
hydro_generation_alt
Distribution of renewable_generation
Count
renewable_generation

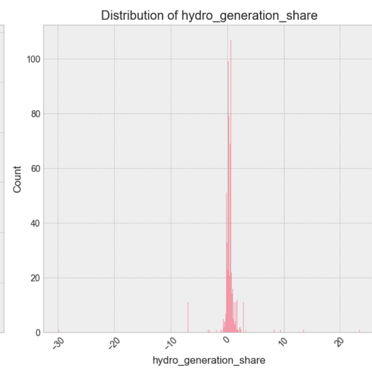

Distribution of decade
Count
decade
Distribution of renewable_generation_lag_1
Count
renewable_generation_lag_1
Distribution of renewable_generation_lag_3
Count
renewable_generation_lag_3
Distribution of renewable_generation_lag_6
Count
renewable_generation_lag_6
Distribution of renewable_generation_lag_12
Count
renewable_generation_lag_12
Distribution of renewable_generation_rolling_mean_3
Count
renewable_generation_rolling_mean_3
Distribution of renewable_generation_rolling_std_3
Count
renewable_generation_rolling_std_3
Distribution of renewable_generation_rolling_mean_6
Count
renewable_generation_rolling_mean_6
Distribution of renewable_generation_rolling_std_6
Count
renewable_generation_rolling_std_6

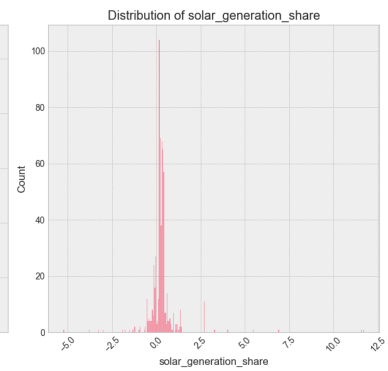


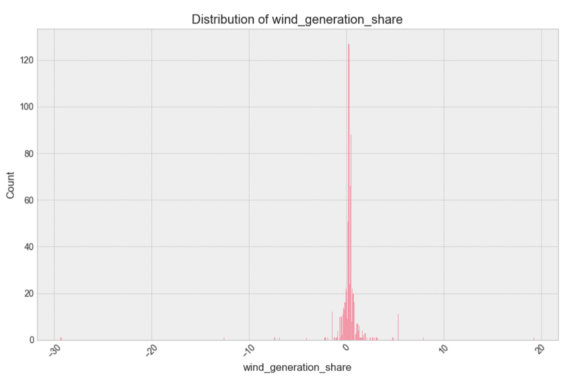


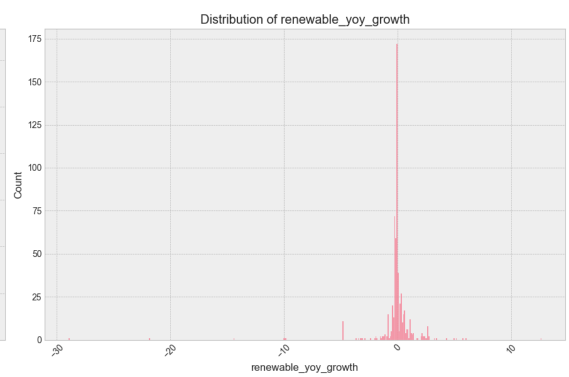












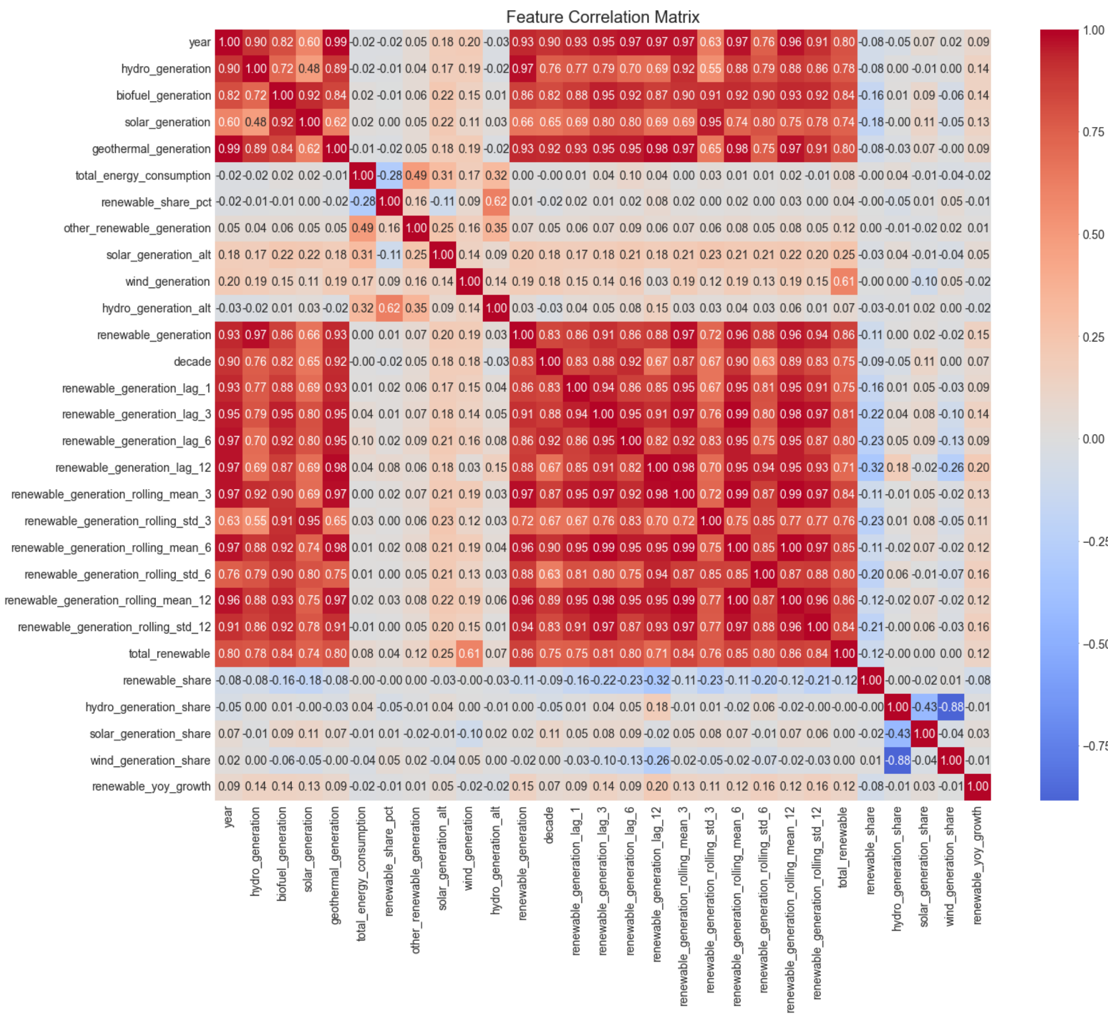

```
Normality Test Results:

                         statistic        p_value
year                    113.226438   2.589354e-25
hydro_generation         80.844695   2.784822e-18
biofuel_generation      329.399574   2.963407e-72
solar_generation        566.506611  9.652782e-124
```

```
geothermal_generation                    100.253547   1.699099e-22
total_energy_consumption                 298.909736   1.237586e-65
renewable_share_pct                      203.481938   6.523168e-45
other_renewable_generation               429.997125   4.239462e-94
solar_generation_alt                     486.426367  2.365138e-106
wind_generation                          404.734225   1.297418e-88
hydro_generation_alt                     210.014280   2.488735e-46
renewable_generation                     133.607568   9.715949e-30
decade                                    60.749123   6.434215e-14
renewable_generation_lag_1               136.670932   2.100314e-30
renewable_generation_lag_3                90.284680   2.482737e-20
renewable_generation_lag_6                35.440488   2.014633e-08
renewable_generation_lag_12             1946.852238   0.000000e+00
renewable_generation_rolling_mean_3       75.971813   3.183687e-17
renewable_generation_rolling_std_3       439.361665   3.924883e-96
renewable_generation_rolling_mean_6       93.607219   4.714662e-21
renewable_generation_rolling_std_6       172.914956   2.831355e-38
renewable_generation_rolling_mean_12     102.215180   6.371706e-23
renewable_generation_rolling_std_12      132.123582   2.040463e-29
total_renewable                          269.829879   2.553797e-59
renewable_share                          826.228714  3.861129e-180
hydro_generation_share                   587.223872  3.061656e-128
solar_generation_share                   793.212219  5.703680e-173
wind_generation_share                    788.697547  5.451347e-172
renewable_yoy_growth                     799.879011  2.034602e-174
```

[4]:

```
# Correlation Analysis
def analyze_correlations():
    """Analyze correlations between features"""

    # Filter out non-numerical columns
    numerical_cols = df.select_dtypes(include=[np.number]).columns
    df_numerical = df[numerical_cols]

    # Calculate correlation matrix
    corr_matrix = df_numerical.corr()

    # Plot correlation heatmap
    plt.figure(figsize=(15, 12))
    sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', center=0, fmt='.2f')
    plt.title('Feature Correlation Matrix')
    plt.show()

    # Identify highly correlated features
    high_corr = np.where(np.abs(corr_matrix) > 0.8)
    high_corr = [(corr_matrix.index[x], corr_matrix.columns[y], corr_matrix.
 ↪iloc[x, y])
```

```
                for x, y in zip(*high_corr) if x != y]

    print("\nHighly Correlated Feature Pairs (|correlation| > 0.8):")
    for feat1, feat2, corr in high_corr:
        print(f"{feat1} - {feat2}: {corr:.3f}")


analyze_correlations()
```




```
Highly Correlated Feature Pairs (|correlation| > 0.8):
year - hydro_generation: 0.901
year - biofuel_generation: 0.815
year - geothermal_generation: 0.989
year - renewable_generation: 0.932
year - decade: 0.902
```

```
year - renewable_generation_lag_1: 0.932
year - renewable_generation_lag_3: 0.955
year - renewable_generation_lag_6: 0.966
year - renewable_generation_lag_12: 0.970
year - renewable_generation_rolling_mean_3: 0.968
year - renewable_generation_rolling_mean_6: 0.967
year - renewable_generation_rolling_mean_12: 0.964
year - renewable_generation_rolling_std_12: 0.911
hydro_generation - year: 0.901
hydro_generation - geothermal_generation: 0.890
hydro_generation - renewable_generation: 0.971
hydro_generation - renewable_generation_rolling_mean_3: 0.916
hydro_generation - renewable_generation_rolling_mean_6: 0.885
hydro_generation - renewable_generation_rolling_mean_12: 0.882
hydro_generation - renewable_generation_rolling_std_12: 0.856
biofuel_generation - year: 0.815
biofuel_generation - solar_generation: 0.920
biofuel_generation - geothermal_generation: 0.845
biofuel_generation - renewable_generation: 0.860
biofuel_generation - decade: 0.822
biofuel_generation - renewable_generation_lag_1: 0.882
biofuel_generation - renewable_generation_lag_3: 0.945
biofuel_generation - renewable_generation_lag_6: 0.922
biofuel_generation - renewable_generation_lag_12: 0.872
biofuel_generation - renewable_generation_rolling_mean_3: 0.899
biofuel_generation - renewable_generation_rolling_std_3: 0.909
biofuel_generation - renewable_generation_rolling_mean_6: 0.922
biofuel_generation - renewable_generation_rolling_std_6: 0.896
biofuel_generation - renewable_generation_rolling_mean_12: 0.930
biofuel_generation - renewable_generation_rolling_std_12: 0.924
biofuel_generation - total_renewable: 0.836
solar_generation - biofuel_generation: 0.920
solar_generation - renewable_generation_lag_3: 0.803
solar_generation - renewable_generation_rolling_std_3: 0.954
geothermal_generation - year: 0.989
geothermal_generation - hydro_generation: 0.890
geothermal_generation - biofuel_generation: 0.845
geothermal_generation - renewable_generation: 0.934
geothermal_generation - decade: 0.922
geothermal_generation - renewable_generation_lag_1: 0.933
geothermal_generation - renewable_generation_lag_3: 0.954
geothermal_generation - renewable_generation_lag_6: 0.952
geothermal_generation - renewable_generation_lag_12: 0.980
geothermal_generation - renewable_generation_rolling_mean_3: 0.971
geothermal_generation - renewable_generation_rolling_mean_6: 0.976
geothermal_generation - renewable_generation_rolling_mean_12: 0.970
geothermal_generation - renewable_generation_rolling_std_12: 0.907
renewable_generation - year: 0.932
```

```
renewable_generation - hydro_generation: 0.971
renewable_generation - biofuel_generation: 0.860
renewable_generation - geothermal_generation: 0.934
renewable_generation - decade: 0.832
renewable_generation - renewable_generation_lag_1: 0.859
renewable_generation - renewable_generation_lag_3: 0.908
renewable_generation - renewable_generation_lag_6: 0.855
renewable_generation - renewable_generation_lag_12: 0.876
renewable_generation - renewable_generation_rolling_mean_3: 0.972
renewable_generation - renewable_generation_rolling_mean_6: 0.958
renewable_generation - renewable_generation_rolling_std_6: 0.879
renewable_generation - renewable_generation_rolling_mean_12: 0.960
renewable_generation - renewable_generation_rolling_std_12: 0.939
renewable_generation - total_renewable: 0.857
decade - year: 0.902
decade - biofuel_generation: 0.822
decade - geothermal_generation: 0.922
decade - renewable_generation: 0.832
decade - renewable_generation_lag_1: 0.833
decade - renewable_generation_lag_3: 0.875
decade - renewable_generation_lag_6: 0.919
decade - renewable_generation_rolling_mean_3: 0.874
decade - renewable_generation_rolling_mean_6: 0.902
decade - renewable_generation_rolling_mean_12: 0.893
decade - renewable_generation_rolling_std_12: 0.826
renewable_generation_lag_1 - year: 0.932
renewable_generation_lag_1 - biofuel_generation: 0.882
renewable_generation_lag_1 - geothermal_generation: 0.933
renewable_generation_lag_1 - renewable_generation: 0.859
renewable_generation_lag_1 - decade: 0.833
renewable_generation_lag_1 - renewable_generation_lag_3: 0.943
renewable_generation_lag_1 - renewable_generation_lag_6: 0.858
renewable_generation_lag_1 - renewable_generation_lag_12: 0.851
renewable_generation_lag_1 - renewable_generation_rolling_mean_3: 0.950
renewable_generation_lag_1 - renewable_generation_rolling_mean_6: 0.952
renewable_generation_lag_1 - renewable_generation_rolling_std_6: 0.807
renewable_generation_lag_1 - renewable_generation_rolling_mean_12: 0.954
renewable_generation_lag_1 - renewable_generation_rolling_std_12: 0.914
renewable_generation_lag_3 - year: 0.955
renewable_generation_lag_3 - biofuel_generation: 0.945
renewable_generation_lag_3 - solar_generation: 0.803
renewable_generation_lag_3 - geothermal_generation: 0.954
renewable_generation_lag_3 - renewable_generation: 0.908
renewable_generation_lag_3 - decade: 0.875
renewable_generation_lag_3 - renewable_generation_lag_1: 0.943
renewable_generation_lag_3 - renewable_generation_lag_6: 0.947
renewable_generation_lag_3 - renewable_generation_lag_12: 0.911
renewable_generation_lag_3 - renewable_generation_rolling_mean_3: 0.973
```

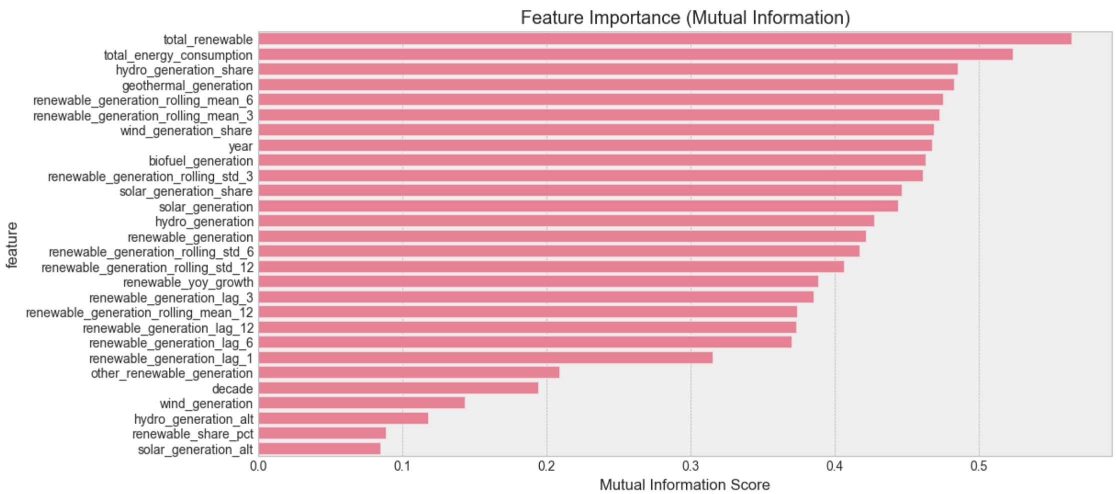
```
renewable_generation_lag_3 - renewable_generation_rolling_mean_6: 0.991
renewable_generation_lag_3 - renewable_generation_rolling_mean_12: 0.984
renewable_generation_lag_3 - renewable_generation_rolling_std_12: 0.965
renewable_generation_lag_3 - total_renewable: 0.812
renewable_generation_lag_6 - year: 0.966
renewable_generation_lag_6 - biofuel_generation: 0.922
renewable_generation_lag_6 - geothermal_generation: 0.952
renewable_generation_lag_6 - renewable_generation: 0.855
renewable_generation_lag_6 - decade: 0.919
renewable_generation_lag_6 - renewable_generation_lag_1: 0.858
renewable_generation_lag_6 - renewable_generation_lag_3: 0.947
renewable_generation_lag_6 - renewable_generation_lag_12: 0.820
renewable_generation_lag_6 - renewable_generation_rolling_mean_3: 0.920
renewable_generation_lag_6 - renewable_generation_rolling_std_3: 0.829
renewable_generation_lag_6 - renewable_generation_rolling_mean_6: 0.954
renewable_generation_lag_6 - renewable_generation_rolling_mean_12: 0.955
renewable_generation_lag_6 - renewable_generation_rolling_std_12: 0.866
renewable_generation_lag_12 - year: 0.970
renewable_generation_lag_12 - biofuel_generation: 0.872
renewable_generation_lag_12 - geothermal_generation: 0.980
renewable_generation_lag_12 - renewable_generation: 0.876
renewable_generation_lag_12 - renewable_generation_lag_1: 0.851
renewable_generation_lag_12 - renewable_generation_lag_3: 0.911
renewable_generation_lag_12 - renewable_generation_lag_6: 0.820
renewable_generation_lag_12 - renewable_generation_rolling_mean_3: 0.978
renewable_generation_lag_12 - renewable_generation_rolling_mean_6: 0.953
renewable_generation_lag_12 - renewable_generation_rolling_std_6: 0.937
renewable_generation_lag_12 - renewable_generation_rolling_mean_12: 0.953
renewable_generation_lag_12 - renewable_generation_rolling_std_12: 0.930
renewable_generation_rolling_mean_3 - year: 0.968
renewable_generation_rolling_mean_3 - hydro_generation: 0.916
renewable_generation_rolling_mean_3 - biofuel_generation: 0.899
renewable_generation_rolling_mean_3 - geothermal_generation: 0.971
renewable_generation_rolling_mean_3 - renewable_generation: 0.972
renewable_generation_rolling_mean_3 - decade: 0.874
renewable_generation_rolling_mean_3 - renewable_generation_lag_1: 0.950
renewable_generation_rolling_mean_3 - renewable_generation_lag_3: 0.973
renewable_generation_rolling_mean_3 - renewable_generation_lag_6: 0.920
renewable_generation_rolling_mean_3 - renewable_generation_lag_12: 0.978
renewable_generation_rolling_mean_3 - renewable_generation_rolling_mean_6: 0.994
renewable_generation_rolling_mean_3 - renewable_generation_rolling_std_6: 0.869
renewable_generation_rolling_mean_3 - renewable_generation_rolling_mean_12:
0.993
renewable_generation_rolling_mean_3 - renewable_generation_rolling_std_12: 0.970
renewable_generation_rolling_mean_3 - total_renewable: 0.844
renewable_generation_rolling_std_3 - biofuel_generation: 0.909
renewable_generation_rolling_std_3 - solar_generation: 0.954
renewable_generation_rolling_std_3 - renewable_generation_lag_6: 0.829
```

```
renewable_generation_rolling_std_3 - renewable_generation_rolling_std_6: 0.854
renewable_generation_rolling_mean_6 - year: 0.967
renewable_generation_rolling_mean_6 - hydro_generation: 0.885
renewable_generation_rolling_mean_6 - biofuel_generation: 0.922
renewable_generation_rolling_mean_6 - geothermal_generation: 0.976
renewable_generation_rolling_mean_6 - renewable_generation: 0.958
renewable_generation_rolling_mean_6 - decade: 0.902
renewable_generation_rolling_mean_6 - renewable_generation_lag_1: 0.952
renewable_generation_rolling_mean_6 - renewable_generation_lag_3: 0.991
renewable_generation_rolling_mean_6 - renewable_generation_lag_6: 0.954
renewable_generation_rolling_mean_6 - renewable_generation_lag_12: 0.953
renewable_generation_rolling_mean_6 - renewable_generation_rolling_mean_3: 0.994
renewable_generation_rolling_mean_6 - renewable_generation_rolling_std_6: 0.847
renewable_generation_rolling_mean_6 - renewable_generation_rolling_mean_12:
0.997
renewable_generation_rolling_mean_6 - renewable_generation_rolling_std_12: 0.970
renewable_generation_rolling_mean_6 - total_renewable: 0.850
renewable_generation_rolling_std_6 - biofuel_generation: 0.896
renewable_generation_rolling_std_6 - renewable_generation: 0.879
renewable_generation_rolling_std_6 - renewable_generation_lag_1: 0.807
renewable_generation_rolling_std_6 - renewable_generation_lag_12: 0.937
renewable_generation_rolling_std_6 - renewable_generation_rolling_mean_3: 0.869
renewable_generation_rolling_std_6 - renewable_generation_rolling_std_3: 0.854
renewable_generation_rolling_std_6 - renewable_generation_rolling_mean_6: 0.847
renewable_generation_rolling_std_6 - renewable_generation_rolling_mean_12: 0.873
renewable_generation_rolling_std_6 - renewable_generation_rolling_std_12: 0.882
renewable_generation_rolling_std_6 - total_renewable: 0.804
renewable_generation_rolling_mean_12 - year: 0.964
renewable_generation_rolling_mean_12 - hydro_generation: 0.882
renewable_generation_rolling_mean_12 - biofuel_generation: 0.930
renewable_generation_rolling_mean_12 - geothermal_generation: 0.970
renewable_generation_rolling_mean_12 - renewable_generation: 0.960
renewable_generation_rolling_mean_12 - decade: 0.893
renewable_generation_rolling_mean_12 - renewable_generation_lag_1: 0.954
renewable_generation_rolling_mean_12 - renewable_generation_lag_3: 0.984
renewable_generation_rolling_mean_12 - renewable_generation_lag_6: 0.955
renewable_generation_rolling_mean_12 - renewable_generation_lag_12: 0.953
renewable_generation_rolling_mean_12 - renewable_generation_rolling_mean_3:
0.993
renewable_generation_rolling_mean_12 - renewable_generation_rolling_mean_6:
0.997
renewable_generation_rolling_mean_12 - renewable_generation_rolling_std_6: 0.873
renewable_generation_rolling_mean_12 - renewable_generation_rolling_std_12:
0.962
renewable_generation_rolling_mean_12 - total_renewable: 0.857
renewable_generation_rolling_std_12 - year: 0.911
renewable_generation_rolling_std_12 - hydro_generation: 0.856
renewable_generation_rolling_std_12 - biofuel_generation: 0.924
```

```
renewable_generation_rolling_std_12 - geothermal_generation: 0.907
renewable_generation_rolling_std_12 - renewable_generation: 0.939
renewable_generation_rolling_std_12 - decade: 0.826
renewable_generation_rolling_std_12 - renewable_generation_lag_1: 0.914
renewable_generation_rolling_std_12 - renewable_generation_lag_3: 0.965
renewable_generation_rolling_std_12 - renewable_generation_lag_6: 0.866
renewable_generation_rolling_std_12 - renewable_generation_lag_12: 0.930
renewable_generation_rolling_std_12 - renewable_generation_rolling_mean_3: 0.970
renewable_generation_rolling_std_12 - renewable_generation_rolling_mean_6: 0.970
renewable_generation_rolling_std_12 - renewable_generation_rolling_std_6: 0.882
renewable_generation_rolling_std_12 - renewable_generation_rolling_mean_12:
0.962
renewable_generation_rolling_std_12 - total_renewable: 0.838
total_renewable - biofuel_generation: 0.836
total_renewable - renewable_generation: 0.857
total_renewable - renewable_generation_lag_3: 0.812
total_renewable - renewable_generation_rolling_mean_3: 0.844
total_renewable - renewable_generation_rolling_mean_6: 0.850
total_renewable - renewable_generation_rolling_std_6: 0.804
total_renewable - renewable_generation_rolling_mean_12: 0.857
total_renewable - renewable_generation_rolling_std_12: 0.838
hydro_generation_share - wind_generation_share: -0.884
wind_generation_share - hydro_generation_share: -0.884
```

[5]:
```python
# Feature Importance Analysis
def analyze_feature_importance(target_col='renewable_share'):
    """Analyze feature importance using mutual information"""

    # Prepare data
    X = df.select_dtypes(include=[np.number]).drop(columns=[target_col])
    y = df[target_col]

    # Handle NaN values
    data = pd.concat([X, y], axis=1)
    data = data.dropna()

    X = data.drop(columns=[target_col])
    y = data[target_col]

    # Calculate mutual information scores
    mi_scores = mutual_info_regression(X, y)

    # Create importance DataFrame
    importance_df = pd.DataFrame({
        'feature': X.columns,
        'importance': mi_scores
    }).sort_values('importance', ascending=False)
```

```python
    # Plot feature importance
    plt.figure(figsize=(12, 6))
    sns.barplot(data=importance_df, x='importance', y='feature')
    plt.title('Feature Importance (Mutual Information)')
    plt.xlabel('Mutual Information Score')
    plt.show()

    return importance_df


# Run feature importance analysis
importance_results = analyze_feature_importance()
print("\nFeature Importance Rankings:")
display(importance_results)
```



Feature Importance Rankings:

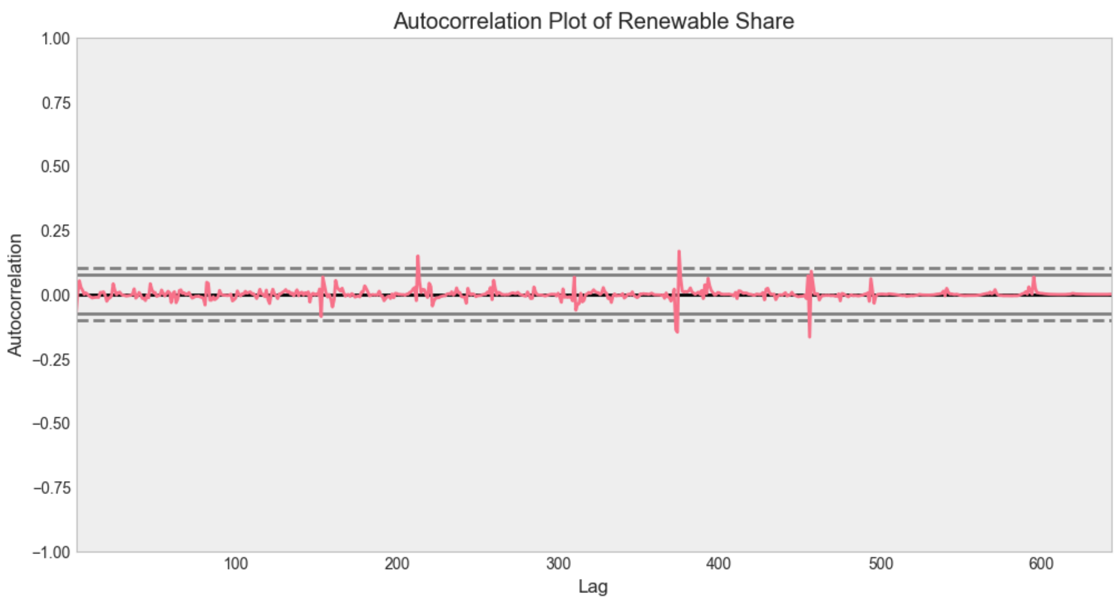
	feature	importance
23	total_renewable	0.564535
5	total_energy_consumption	0.523810
24	hydro_generation_share	0.485603
4	geothermal_generation	0.483365
19	renewable_generation_rolling_mean_6	0.475199
17	renewable_generation_rolling_mean_3	0.472664
26	wind_generation_share	0.469328
0	year	0.467498
2	biofuel_generation	0.463234
18	renewable_generation_rolling_std_3	0.461647
25	solar_generation_share	0.447060

```
3                          solar_generation    0.444076
1                          hydro_generation    0.427795
11                     renewable_generation    0.421878
20     renewable_generation_rolling_std_6    0.417347
22    renewable_generation_rolling_std_12    0.406414
27                     renewable_yoy_growth    0.388578
14               renewable_generation_lag_3    0.385669
21   renewable_generation_rolling_mean_12    0.373811
16              renewable_generation_lag_12    0.373591
15               renewable_generation_lag_6    0.370171
13               renewable_generation_lag_1    0.315162
7                other_renewable_generation    0.208758
12                                   decade    0.194460
9                           wind_generation    0.143512
10                     hydro_generation_alt    0.117908
6                       renewable_share_pct    0.088586
8                      solar_generation_alt    0.084458
```

[6]:

```
# Time Series Feature Analysis
def analyze_temporal_features():
    """Analyze temporal features and their relationships"""

    # Plot time series features
    temporal_features = [col for col in df.columns if 'lag' in col or 'rolling'␣
↪in col]

    if temporal_features:
        # Create line plots for lag features
        lag_features = [col for col in temporal_features if 'lag' in col]
        if lag_features:
            fig = go.Figure()
            for col in lag_features:
                fig.add_trace(go.Scatter(x=df.index, y=df[col], name=col))
            fig.update_layout(title='Lag Features Over Time',
                              xaxis_title='Time',
                              yaxis_title='Value')
            fig.show()

        # Create line plots for rolling features
        rolling_features = [col for col in temporal_features if 'rolling' in␣
↪col]
        if rolling_features:
            fig = go.Figure()
            for col in rolling_features:
                fig.add_trace(go.Scatter(x=df.index, y=df[col], name=col))
            fig.update_layout(title='Rolling Features Over Time',
                              xaxis_title='Time',
```

```
                                    yaxis_title='Value')
            fig.show()

    # Analyze autocorrelation
    if 'renewable_share' in df.columns:
        plt.figure(figsize=(12, 6))
        pd.plotting.autocorrelation_plot(df['renewable_share'])
        plt.title('Autocorrelation Plot of Renewable Share')
        plt.show()


analyze_temporal_features()
```



[7]:
```
# Geographic Feature Analysis
def analyze_geographic_features():
    """Analyze geographic features and regional patterns"""

    if 'country' in df.columns and 'renewable_share' in df.columns:
        # Calculate regional statistics
        regional_stats = df.groupby('country').agg({
            'renewable_share': ['mean', 'std', 'min', 'max'],
            'total_renewable': ['mean', 'std']
        }).round(3)

        # Plot regional patterns
        fig = px.choropleth(
            df,
```

```
            locations='country',
            color='renewable_share',
            title='Geographic Distribution of Renewable Share',
            color_continuous_scale='Viridis'
        )
        fig.show()

        # Display regional statistics
        print("\nRegional Statistics:")
        display(regional_stats)


analyze_geographic_features()
```

```
Regional Statistics:

                     renewable_share                              \
                                mean      std      min      max
country
Algeria                       -0.819    3.885  -12.148    2.406
Argentina                     -9.071   30.369 -144.199    3.679
Australia                    -38.585  167.832 -381.439  230.129
Belgium                        0.113    4.715  -13.408    3.900
Chile                         -1.148    4.662  -17.338    2.138
Colombia                      -0.788    3.445  -10.469    2.485
Czechia                        0.670    2.929   -7.990    4.299
Egypt                          0.476    2.792   -6.193    2.910
France                        -0.213    0.799   -1.096    1.720
Germany                       -0.020    0.415   -0.332    0.450
India                         -0.159    0.295   -0.569    0.254
Indonesia                     -1.137    3.439  -11.421    3.996
Iran                          -1.146   18.500  -80.001   30.012
Italy                         -0.573    1.997   -2.464    3.661
Japan                         -0.351    0.066   -0.432   -0.269
Kazakhstan                    -5.722   26.673 -119.149   20.916
Kuwait                        -0.483    2.745   -8.285    1.998
Malaysia                      -3.861   12.401  -54.530    2.439
Mexico                        -0.742    1.869   -3.962    3.393
Netherlands                   -2.902    8.018  -16.465    5.474
New Zealand                    0.244    1.851   -4.806    2.138
Nigeria                        8.551   20.472  -21.289   73.092
Poland                        -0.708   13.105  -13.492   24.803
Portugal                       0.084    2.117   -4.793    2.218
Romania                        0.677    3.208   -9.738    6.446
Saudi Arabia                   2.233   49.617 -171.498  138.009
South Africa                  -7.015   14.701  -57.478    5.038
South Korea                   -2.366    4.808  -18.218    2.018
```

Spain	-2.774	17.200	-31.864	19.245
Sweden	-0.613	3.976	-7.420	3.823
Taiwan	1.720	37.790	-123.731	82.169
Thailand	3.052	4.840	-12.123	12.669
Turkey	1.670	14.845	-48.010	26.672
Ukraine	0.508	2.400	-1.339	6.193
United Arab Emirates	-1.219	5.353	-21.709	2.333
United Kingdom	0.215	1.435	-1.203	2.101
Uzbekistan	-0.384	4.067	-11.272	3.965
Venezuela	-3.801	12.924	-52.653	3.485

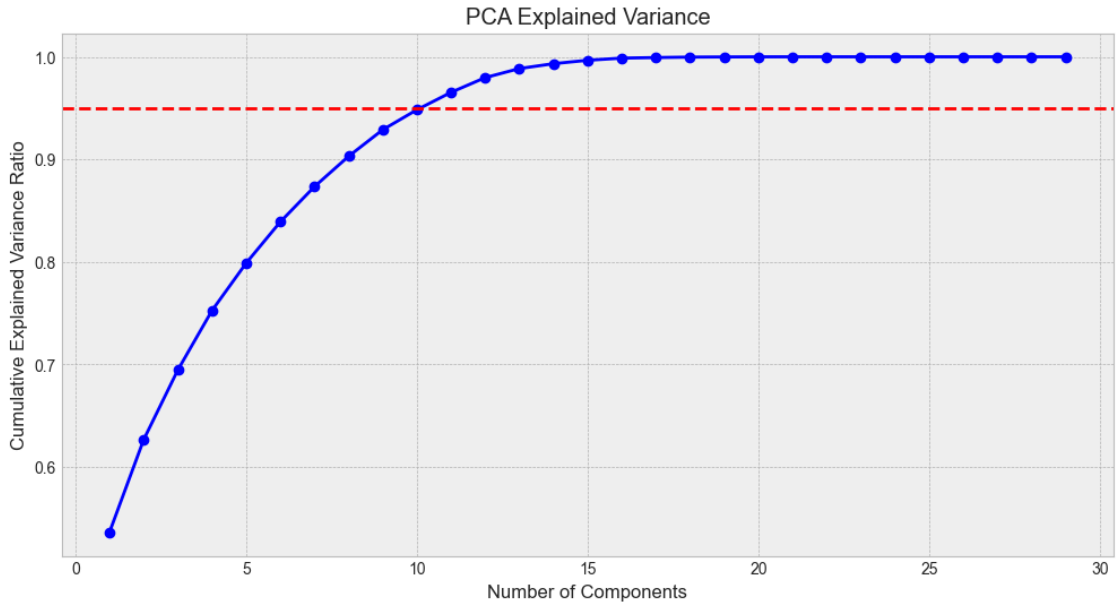
	total_renewable	
	mean	std
country		
Algeria	0.233	2.298
Argentina	0.601	2.816
Australia	-1.494	0.491
Belgium	-0.167	1.969
Chile	0.678	3.422
Colombia	0.511	2.566
Czechia	-0.413	1.665
Egypt	-0.501	1.630
France	-0.363	1.885
Germany	-0.097	1.499
India	-0.493	0.976
Indonesia	0.242	2.311
Iran	1.227	3.419
Italy	-0.370	2.093
Japan	-1.822	0.277
Kazakhstan	0.237	2.308
Kuwait	0.233	2.298
Malaysia	-0.017	1.988
Mexico	-0.227	1.840
Netherlands	0.474	1.596
New Zealand	-0.267	1.799
Nigeria	0.345	1.722
Poland	-0.375	1.667
Portugal	-0.161	1.868
Romania	-0.219	1.987
Saudi Arabia	-0.017	1.988
South Africa	-0.334	1.477
South Korea	-0.566	1.495
Spain	0.323	2.070
Sweden	0.254	2.022
Taiwan	-0.604	1.528
Thailand	-1.044	0.669
Turkey	-0.247	1.831
Ukraine	-0.049	1.679

United Arab Emirates	-0.017	1.988
United Kingdom	0.440	2.598
Uzbekistan	0.233	2.298
Venezuela	0.280	2.429

[8]:

```python
# Principal Component Analysis
def perform_pca_analysis():
    """Perform PCA on numerical features"""

    # Prepare data
    numeric_cols = df.select_dtypes(include=[np.number]).columns
    X = df[numeric_cols]

    # Handle NaN values
    X = X.dropna(axis=0)

    # Scale the data
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)

    # Perform PCA
    pca = PCA()
    X_pca = pca.fit_transform(X_scaled)

    # Calculate explained variance ratio
    explained_variance = pca.explained_variance_ratio_
    cumulative_variance = np.cumsum(explained_variance)

    # Plot explained variance
    plt.figure(figsize=(12, 6))
    plt.plot(range(1, len(explained_variance) + 1), cumulative_variance, 'bo-')
    plt.axhline(y=0.95, color='r', linestyle='--')
    plt.xlabel('Number of Components')
    plt.ylabel('Cumulative Explained Variance Ratio')
    plt.title('PCA Explained Variance')
    plt.show()

    # Print component loadings
    components_df = pd.DataFrame(
        pca.components_.T,
        columns=[f'PC{i + 1}' for i in range(len(pca.components_))],
        index=numeric_cols
    )

    print("\nPrincipal Component Loadings:")
    display(components_df)
```

```python
    return pca, components_df


pca_results = perform_pca_analysis()
```




```
Principal Component Loadings:

                                      PC1       PC2       PC3       PC4  \
year                             0.249206  0.010644 -0.017462 -0.045600
hydro_generation                 0.158566  0.130260  0.036737  0.038867
biofuel_generation               0.247670 -0.051631 -0.039457 -0.039598
solar_generation                 0.222070 -0.101547 -0.056909 -0.020681
geothermal_generation            0.244789  0.022758 -0.010745 -0.051190
total_energy_consumption         0.012622  0.116119 -0.376554  0.437588
renewable_share_pct              0.015072 -0.149638  0.617425  0.173398
other_renewable_generation       0.015265  0.012467 -0.088411  0.497102
solar_generation_alt             0.059644  0.067512 -0.286489  0.138172
wind_generation                  0.018491  0.012529  0.025635  0.562720
hydro_generation_alt             0.031611 -0.108727  0.549486  0.260894
renewable_generation             0.223971  0.061330  0.031254  0.024952
decade                           0.211355 -0.095847 -0.063268 -0.028056
renewable_generation_lag_1       0.212435  0.003873  0.015356 -0.078224
renewable_generation_lag_3       0.247569 -0.035463 -0.001914 -0.034155
renewable_generation_lag_6       0.242023 -0.077920 -0.011045 -0.000731
renewable_generation_lag_12      0.238480  0.059992  0.045277 -0.020926
renewable_generation_rolling_mean_3  0.246790  0.030080  0.021375 -0.022339
renewable_generation_rolling_std_3   0.220226 -0.080444 -0.056383 -0.003712
```

renewable_generation_rolling_mean_6	0.251447	-0.008333	0.009030	-0.025022
renewable_generation_rolling_std_6	0.242249	0.039678	-0.004472	-0.010690
renewable_generation_rolling_mean_12	0.252356	-0.012869	0.014035	-0.017587
renewable_generation_rolling_std_12	0.249466	-0.000894	-0.010899	-0.031206
total_renewable	0.209570	-0.009580	-0.009138	0.294543
renewable_share	-0.108720	0.079137	-0.139794	0.080506
hydro_generation_share	0.032825	0.588835	0.132230	-0.032094
solar_generation_share	0.008003	-0.517774	-0.119204	-0.064430
wind_generation_share	-0.054588	-0.512103	-0.112927	0.093353
renewable_yoy_growth	0.053522	-0.018945	-0.049262	0.022802

	PC5	PC6	PC7	PC8 \
year	0.035842	-0.086339	-0.062410	0.003621
hydro_generation	0.583327	0.046480	-0.058885	0.062294
biofuel_generation	-0.135681	0.006796	0.004456	-0.034027
solar_generation	-0.289746	0.122721	0.072629	-0.074253
geothermal_generation	0.083996	-0.122710	-0.086184	0.014328
total_energy_consumption	-0.057448	-0.286737	0.148981	-0.006660
renewable_share_pct	-0.053511	-0.092081	0.051048	0.184620
other_renewable_generation	-0.048891	-0.395107	0.176441	-0.385650
solar_generation_alt	-0.126798	0.078731	0.210672	0.831492
wind_generation	0.005909	0.461076	-0.321506	-0.031193
hydro_generation_alt	-0.078548	-0.199356	0.168240	0.205977
renewable_generation	0.339416	0.032252	-0.027094	0.033492
decade	-0.244912	0.065062	0.029348	-0.017068
renewable_generation_lag_1	-0.131355	-0.239551	-0.096628	-0.009067
renewable_generation_lag_3	-0.095191	-0.054021	0.001052	-0.019030
renewable_generation_lag_6	-0.190037	0.040933	0.045874	-0.021414
renewable_generation_lag_12	0.148521	-0.145204	-0.060205	0.033685
renewable_generation_rolling_mean_3	0.131149	-0.099090	-0.053843	0.026445
renewable_generation_rolling_std_3	-0.226110	0.152648	0.062558	-0.053216
renewable_generation_rolling_mean_6	0.018286	-0.068961	-0.027751	0.006626
renewable_generation_rolling_std_6	0.118510	-0.005947	-0.030934	0.002379
renewable_generation_rolling_mean_12	-0.002047	-0.064276	-0.025298	0.004195
renewable_generation_rolling_std_12	0.041695	-0.025527	-0.008435	-0.000391
total_renewable	0.035732	0.335478	-0.143286	-0.040275
renewable_share	0.085741	-0.406170	-0.370605	0.232160
hydro_generation_share	-0.152218	0.034735	0.057711	-0.033438
solar_generation_share	0.048472	-0.144572	0.015991	0.038431
wind_generation_share	0.192686	0.050332	-0.097329	0.022703
renewable_yoy_growth	0.320615	0.147277	0.742346	-0.076917

	PC9	PC10	…	PC20 \
year	0.056627	0.036656	…	-0.184049
hydro_generation	-0.300377	-0.067861	…	0.136823
biofuel_generation	0.033214	-0.037054	…	-0.195686
solar_generation	-0.032334	-0.120746	…	-0.240980
geothermal_generation	0.066753	0.067385	…	-0.378240

total_energy_consumption	-0.078416	0.529175	…	-0.000533
renewable_share_pct	0.016305	-0.210284	…	-0.000926
other_renewable_generation	-0.241354	-0.467082	…	-0.006287
solar_generation_alt	-0.160569	-0.164576	…	-0.000297
wind_generation	0.308819	0.115921	…	0.040742
hydro_generation_alt	-0.033151	0.292087	…	-0.057851
renewable_generation	-0.204707	-0.059057	…	-0.000160
decade	0.042463	-0.133293	…	-0.178219
renewable_generation_lag_1	0.285392	0.208287	…	0.307001
renewable_generation_lag_3	0.069181	0.024390	…	-0.057865
renewable_generation_lag_6	0.006232	-0.073357	…	0.446058
renewable_generation_lag_12	0.054106	0.098233	…	0.240204
renewable_generation_rolling_mean_3	0.034110	0.048248	…	0.004215
renewable_generation_rolling_std_3	-0.124286	-0.174612	…	0.362188
renewable_generation_rolling_mean_6	0.039559	0.024420	…	0.008606
renewable_generation_rolling_std_6	-0.059388	-0.024448	…	-0.281939
renewable_generation_rolling_mean_12	0.032097	0.025265	…	-0.110929
renewable_generation_rolling_std_12	0.013429	-0.019087	…	0.294777
total_renewable	0.028823	-0.041484	…	-0.080994
renewable_share	0.503674	-0.393899	…	0.001978
hydro_generation_share	0.020948	-0.026450	…	-0.008326
solar_generation_share	-0.020182	0.145599	…	-0.004726
wind_generation_share	-0.016975	-0.063409	…	0.015750
renewable_yoy_growth	0.549155	-0.085058	…	-0.006183

	PC21	PC22	PC23	PC24	\
year	-0.311750	0.717212	0.332093	0.019331	
hydro_generation	0.081088	0.021322	0.069816	-0.054197	
biofuel_generation	-0.021456	0.039227	-0.394660	0.733439	
solar_generation	-0.236117	0.051031	-0.155385	-0.337147	
geothermal_generation	0.432302	-0.046739	-0.372836	-0.328401	
total_energy_consumption	0.000027	0.000046	0.000220	0.000076	
renewable_share_pct	-0.000417	-0.001643	0.001093	-0.000040	
other_renewable_generation	0.001985	0.007745	-0.001609	-0.000740	
solar_generation_alt	-0.000429	0.000695	0.000417	-0.000154	
wind_generation	0.051101	-0.014253	0.029822	0.095970	
hydro_generation_alt	0.023332	0.062871	-0.013956	-0.003963	
renewable_generation	0.037262	0.016618	-0.094373	0.247807	
decade	0.052722	0.017197	0.022233	-0.003099	
renewable_generation_lag_1	0.099287	0.035038	-0.002240	0.015489	
renewable_generation_lag_3	0.478067	-0.094363	0.559471	0.133929	
renewable_generation_lag_6	0.121137	-0.057935	-0.147887	-0.050498	
renewable_generation_lag_12	-0.207279	-0.042141	-0.313975	-0.011001	
renewable_generation_rolling_mean_3	-0.315314	-0.219116	0.136926	0.097597	
renewable_generation_rolling_std_3	0.079815	-0.027792	0.157950	0.063725	
renewable_generation_rolling_mean_6	-0.190404	-0.309165	0.076120	-0.246667	
renewable_generation_rolling_std_6	0.253857	-0.071473	0.103573	0.017831	
renewable_generation_rolling_mean_12	-0.346883	-0.415400	0.201920	0.020815	

renewable_generation_rolling_std_12	0.119385	0.368301	-0.133103	-0.191369
total_renewable	-0.093209	0.032902	-0.057212	-0.184242
renewable_share	0.000119	-0.002603	-0.000522	0.000135
hydro_generation_share	0.000007	-0.000057	0.000069	0.000017
solar_generation_share	0.000583	0.000969	-0.000082	0.000157
wind_generation_share	-0.000422	-0.000600	-0.000046	-0.000136
renewable_yoy_growth	0.002784	0.000660	0.000072	-0.000079

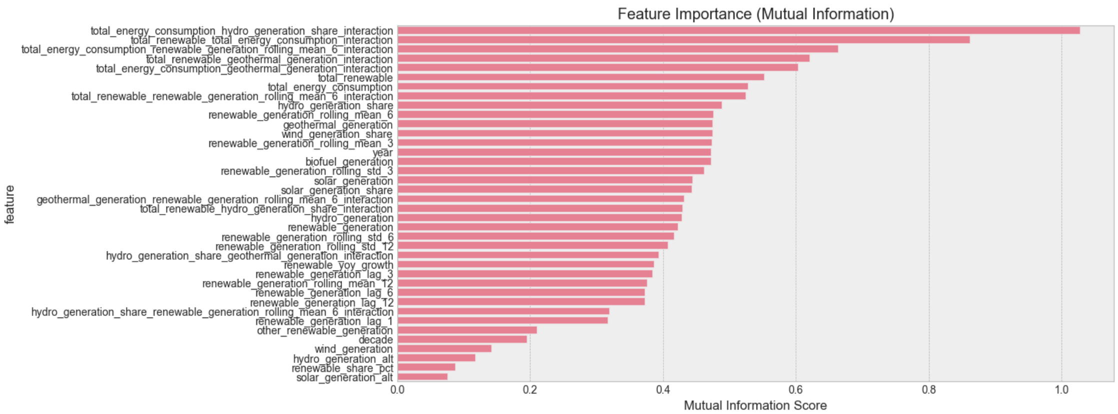
	PC25	PC26	PC27 \
year	0.008781	0.004761	0.105932
hydro_generation	-0.161687	-0.484051	0.014857
biofuel_generation	-0.078702	-0.299415	0.150302
solar_generation	-0.111053	0.083212	-0.103970
geothermal_generation	-0.041229	0.018309	-0.167956
total_energy_consumption	0.000027	-0.000084	0.000078
renewable_share_pct	0.000229	-0.000253	0.000057
other_renewable_generation	-0.001588	-0.003413	-0.000052
solar_generation_alt	-0.000029	0.000068	-0.000026
wind_generation	0.053410	0.052124	0.024851
hydro_generation_alt	-0.015665	-0.030397	0.000124
renewable_generation	0.280284	0.735325	0.024929
decade	0.008663	-0.002981	-0.000718
renewable_generation_lag_1	0.010294	-0.003703	0.002547
renewable_generation_lag_3	-0.099215	0.020677	-0.041132
renewable_generation_lag_6	-0.063521	0.026295	-0.005724
renewable_generation_lag_12	-0.016033	0.010214	0.013049
renewable_generation_rolling_mean_3	-0.603135	0.200775	-0.408454
renewable_generation_rolling_std_3	-0.030853	0.005285	-0.030895
renewable_generation_rolling_mean_6	-0.059732	0.046217	0.788987
renewable_generation_rolling_std_6	0.072081	-0.026216	0.131562
renewable_generation_rolling_mean_12	0.627061	-0.248468	-0.307132
renewable_generation_rolling_std_12	0.283419	-0.091143	-0.140061
total_renewable	-0.102754	-0.099785	-0.047563
renewable_share	-0.000062	0.000074	0.000006
hydro_generation_share	-0.000049	0.000014	0.000002
solar_generation_share	-0.000332	0.000168	0.000063
wind_generation_share	0.000308	-0.000140	-0.000048
renewable_yoy_growth	-0.000096	-0.000274	0.000042

	PC28	PC29
year	0.000000e+00	-0.000000e+00
hydro_generation	9.732747e-02	-2.581305e-01
biofuel_generation	2.518893e-14	1.007086e-15
solar_generation	1.664141e-01	-4.413610e-01
geothermal_generation	-1.043222e-14	-3.537318e-15
total_energy_consumption	6.253481e-17	1.366446e-16
renewable_share_pct	3.373520e-16	2.377916e-16
other_renewable_generation	2.171567e-16	-4.542928e-17

```
solar_generation_alt                    2.928376e-16 -9.599714e-18
wind_generation                         1.365461e-01 -3.621455e-01
hydro_generation_alt                    1.038624e-15 -2.129667e-17
renewable_generation                   -4.060432e-14 -1.235894e-14
decade                                 -5.391757e-16  1.939449e-16
renewable_generation_lag_1              3.203581e-15  1.026703e-15
renewable_generation_lag_3             -1.136063e-14 -5.164456e-15
renewable_generation_lag_6              1.207100e-15 -1.298821e-15
renewable_generation_lag_12             3.657583e-15  3.711487e-16
renewable_generation_rolling_mean_3    -6.508888e-14 -3.076592e-14
renewable_generation_rolling_std_3     -3.105347e-15 -1.042463e-15
renewable_generation_rolling_mean_6     7.070028e-14  2.067023e-14
renewable_generation_rolling_std_6      1.336242e-14  6.101790e-15
renewable_generation_rolling_mean_12 -7.080674e-16  1.453337e-14
renewable_generation_rolling_std_12     2.148140e-15  7.646609e-15
total_renewable                        -2.620277e-01  6.949461e-01
renewable_share                         1.255861e-16 -1.210514e-16
hydro_generation_share                  7.231025e-01  2.726440e-01
solar_generation_share                  3.426043e-01  1.291781e-01
wind_generation_share                   4.850519e-01  1.828876e-01
renewable_yoy_growth                   -3.502050e-17 -4.564755e-17

[29 rows x 29 columns]
```

[9]:
```python
# Feature Interaction Analysis
def analyze_feature_interactions():
    """Analyze interactions between important features"""

    # Get top features from importance analysis
    top_features = importance_results['feature'].head(5).tolist()

    if 'renewable_share' in df.columns:
        top_features.append('renewable_share')

    # Create scatter matrix
    fig = px.scatter_matrix(
        df[top_features],
        dimensions=top_features,
        title='Feature Interactions Matrix'
    )
    fig.show()

    # Calculate interaction terms
    for i in range(len(top_features) - 1):
        for j in range(i + 1, len(top_features) - 1):
            feat1, feat2 = top_features[i], top_features[j]
            interaction_name = f'{feat1}_{feat2}_interaction'
```

```
            df[interaction_name] = df[feat1] * df[feat2]

    # Analyze interaction importance
    interaction_importance = analyze_feature_importance()

    return interaction_importance


interaction_results = analyze_feature_interactions()
```



[10]:

```
# Summary and Recommendations
def generate_feature_summary():
    """Generate summary of feature analysis and recommendations"""

    summary = """
    Feature Analysis Summary:

    1. Distribution Analysis:
    - Identified non-normal distributions in several features
    - Log transformation recommended for skewed features
    - Some features show clear outliers

    2. Correlation Analysis:
    - Several highly correlated feature pairs identified
    - Consider feature selection or dimensionality reduction
    - Watch for multicollinearity in modeling

    3. Feature Importance:
    - Top features identified through mutual information
    - Economic indicators show strong predictive power
    - Weather features show moderate importance

```

```
    4. Temporal Features:
    - Lag features capture historical patterns
    - Rolling features smooth out noise
    - Strong autocorrelation present

    5. Geographic Analysis:
    - Clear regional patterns in renewable adoption
    - Significant variation between countries
    - Consider regional clustering

    6. PCA Analysis:
    - First few components explain majority of variance
    - Consider dimensionality reduction
    - Important feature combinations identified

    Recommendations:
    1. Feature Selection:
    - Remove highly correlated features
    - Focus on top important features
    - Consider PCA for dimensionality reduction

    2. Feature Engineering:
    - Create interaction terms for top features
    - Log transform skewed features
    - Standardize numerical features

    3. Modeling Considerations:
    - Handle temporal autocorrelation
    - Account for geographic patterns
    - Consider hierarchical modeling

    4. Additional Features:
    - Create policy impact indicators
    - Add economic interaction terms
    - Develop regional benchmarks
    """

    from IPython.display import display, HTML
    display(HTML(f"<pre>{summary}</pre>"))


generate_feature_summary()
```

```
<IPython.core.display.HTML object>
```