Energy_Dataset_18_22

March 9, 2025

1 Renewable Energy Consumption Trends and CO Emissions Analysis (2018-2022)

1.1 Project Introduction

This analysis explores the evolution of renewable energy consumption and its impact on CO emissions across 175 countries during 2018-2022. Motivated by the urgency of the global energy transition in the face of climate change, this project aims to identify patterns, leaders, and challenges in the shift toward sustainable energy systems.

The analysis reveals significant regional disparities in renewable energy adoption, ranging from 49.75% in South America to only 11.46% in the Middle East. It examines the complex relationship between renewable energy development and CO emissions reduction, which varies considerably across regions. While some areas like Europe (-0.69) and Oceania (-1.00) show promising negative correlations (more renewables = less emissions), other regions like Asia (+0.99) demonstrate that increasing renewable capacity doesn't automatically reduce emissions when economic growth remains tied to fossil fuel consumption.

Using statistical analysis, data visualization, and predictive modeling techniques, this project:

1. Maps global renewable energy trends at national and regional levels
2. Identifies leading countries in both absolute production and proportion of energy mix
3. Analyzes correlations between renewable adoption and emissions reduction
4. Develops forecasts for renewable energy growth and emissions for 2023-2027
5. Formulates strategic recommendations for accelerating the energy transition

The findings highlight that while renewable energy has grown by 11.51% globally between 2018-2022, reaching 25.05% of the global energy mix, this pace remains insufficient to meet Paris Agreement climate objectives. The analysis offers evidence-based insights for policymakers, businesses, and international organizations to develop more effective decarbonization strategies.

Through interactive visualizations and comprehensive data analysis, this notebook provides both retrospective understanding and prospective guidance on one of the most critical challenges of our time: transforming global energy systems to mitigate climate change while supporting economic development.

[1]:
```
# Étape 1 : Importer et explorer les données

# Importer les bibliothèques nécessaires
import pandas as pd
```

```python
import matplotlib.pyplot as plt

# Charger le fichier
df = pd.read_csv('C:/Users/user/Desktop/energy_dataset_18_22.csv', sep=';')


# Afficher les premières lignes pour voir la structure
print("Aperçu des premières lignes du dataset :")
display(df.head())

# Information sur le dataset
print("\nInformations sur le dataset :")
print(df.info())

#  Vérifier les types de données
print("\nTypes de colonnes :")
print(df.dtypes)

# Vérifier les valeurs manquantes
print("\nNombre de valeurs manquantes par colonne :")
print(df.isnull().sum())

# Résumé statistique des données numériques
print("\nStatistiques descriptives :")
display(df.describe())

# Nombre de lignes et de colonnes
print("\nDimensions du dataset : ", df.shape)
```

```
Aperçu des premières lignes du dataset :

       country  year    coal    gas    oil  nuclear  hydro  solar   wind  \
0        China  2018  2800.0  240.0  600.0     70.0  270.0  170.0  360.0
1        India  2018   990.0   75.0  245.0     40.0  130.0   30.0   60.0
2        Japan  2018   310.0  145.0  190.0     65.0   85.0   55.0   15.0
3  South Corea  2018   230.0   80.0  120.0    130.0    8.0   15.0    7.0
4    Indonesia  2018   185.0   85.0   95.0      0.0   35.0    5.0    2.0

   biofuel      gdp  population total_energy  renewable_energy  \
0     90.0  13895.0      1393.0       4600.0             890.0
1     45.0   2701.0      1353.0       1615.0             265.0
2     25.0   4971.0       126.0        890.0             180.0
3     12.0   1724.0        51.0        602.0              42.0
4     25.0   1042.0       268.0        432.0              67.0

   renewables_share_energy co2_emissions  co2_per_capita  co2_per_gdp  \
0                   19.348       13544.7           9.723        0.975
1                   16.409        4871.4           3.600        1.804
```

```
2                   20.225        2166.9          17.198        0.436
3                    6.977       1475.96          28.940        0.856
4                   15.509        1234.5           4.606        1.185

  energy_with_unit emissions_with_unit
0           4600.0             13544.7
1           1615.0              4871.4
2            890.0              2166.9
3            602.0             1475.96
4            432.0              1234.5

Informations sur le dataset :
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 175 entries, 0 to 174
Data columns (total 20 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   country                 175 non-null    object
 1   year                    175 non-null    int64
 2   coal                    172 non-null    object
 3   gas                     172 non-null    object
 4   oil                     173 non-null    object
 5   nuclear                 175 non-null    float64
 6   hydro                   173 non-null    float64
 7   solar                   175 non-null    float64
 8   wind                    173 non-null    float64
 9   biofuel                 169 non-null    float64
 10  gdp                     175 non-null    float64
 11  population              175 non-null    float64
 12  total_energy            167 non-null    object
 13  renewable_energy        175 non-null    float64
 14  renewables_share_energy  175 non-null    float64
 15  co2_emissions           169 non-null    object
 16  co2_per_capita          175 non-null    float64
 17  co2_per_gdp             175 non-null    float64
 18  energy_with_unit        175 non-null    object
 19  emissions_with_unit     175 non-null    object
dtypes: float64(11), int64(1), object(8)
memory usage: 27.5+ KB
None

Types de colonnes :
country                     object
year                         int64
coal                        object
gas                         object
oil                         object
```

```
nuclear                    float64
hydro                      float64
solar                      float64
wind                       float64
biofuel                    float64
gdp                        float64
population                 float64
total_energy                object
renewable_energy           float64
renewables_share_energy    float64
co2_emissions               object
co2_per_capita             float64
co2_per_gdp                float64
energy_with_unit            object
emissions_with_unit         object
dtype: object

Nombre de valeurs manquantes par colonne :
country                    0
year                       0
coal                       3
gas                        3
oil                        2
nuclear                    0
hydro                      2
solar                      0
wind                       2
biofuel                    6
gdp                        0
population                 0
total_energy               8
renewable_energy           0
renewables_share_energy    0
co2_emissions              6
co2_per_capita             0
co2_per_gdp                0
energy_with_unit           0
emissions_with_unit        0
dtype: int64

Statistiques descriptives :
              year     nuclear       hydro       solar        wind  \
count   175.000000  175.000000  173.000000  175.000000  173.000000
mean   2020.000000   38.362286   68.290231   24.200114   41.056879
std       1.418272   79.135662  104.158597   38.916486   85.905430
min    2018.000000    0.000000    0.000000    0.100000    0.000000
25%    2019.000000    0.000000   11.220000    5.100000    2.240000
```

```
50%    2020.000000    0.000000   25.500000   10.000000    9.200000
75%    2021.000000   39.400000   58.800000   26.875000   33.600000
max    2022.000000  387.600000  387.600000  221.000000  450.000000

          biofuel          gdp   population  renewable_energy  \
count  169.000000   175.000000   175.000000        175.000000
mean    20.829586  2060.257143   155.705714        153.918114
std     29.737794  3997.633061   309.030181        222.283448
min      0.000000    42.300000     4.900000          6.000000
25%      5.000000   350.745000    34.000000         33.550000
50%     10.500000   606.670000    67.000000         65.150000
75%     25.000000  1842.900000   126.000000        195.850000
max    152.250000 21553.350000  1393.000000       1038.200000

       renewables_share_energy  co2_per_capita  co2_per_gdp
count               175.000000      175.000000   175.000000
mean                 25.130897       16.867011     1.146000
std                  15.457590       15.069676     0.681001
min                   1.980000        1.429000     0.238000
25%                  14.608000        5.911500     0.618000
50%                  23.077000       11.998000     1.010000
75%                  33.000000       22.104500     1.509500
max                  74.101000       69.382000     3.052000


Dimensions du dataset :  (175, 20)
```

[5]:

```python
# Nettoyage des données

# Import des bibliothèques nécessaires
import pandas as pd
import numpy as np

# Charger le fichier
df = pd.read_csv('C:/Users/user/Desktop/energy_dataset_18_22.csv', sep=';')

# Afficher les types AVANT conversion
print("Types de données AVANT conversion :")
print(df.dtypes)

# Convertir les colonnes en numérique - avec les bons noms de colonnes
columns_to_convert = [
    'coal', 'gas', 'oil',
    'nuclear', 'hydro', 'solar',
    'wind', 'biofuel', 'renewable_energy'
]

# Conversion en numérique
```

```python
df[columns_to_convert] = df[columns_to_convert].apply(pd.to_numeric, errors='coerce')

# Vérifier les types de données
print("\nTypes de données après conversion :")
print(df.dtypes)
```

```
Types de données AVANT conversion :
country                     object
year                         int64
coal                        object
gas                         object
oil                         object
nuclear                    float64
hydro                      float64
solar                      float64
wind                       float64
biofuel                    float64
gdp                        float64
population                 float64
total_energy                object
renewable_energy           float64
renewables_share_energy    float64
co2_emissions               object
co2_per_capita             float64
co2_per_gdp                float64
energy_with_unit            object
emissions_with_unit         object
dtype: object

Types de données après conversion :
country                     object
year                         int64
coal                       float64
gas                        float64
oil                        float64
nuclear                    float64
hydro                      float64
solar                      float64
wind                       float64
biofuel                    float64
gdp                        float64
population                 float64
total_energy                object
renewable_energy           float64
renewables_share_energy    float64
co2_emissions               object
co2_per_capita             float64
```

```
co2_per_gdp                float64
energy_with_unit            object
emissions_with_unit         object
dtype: object
```

[7]:

```python
# Conversion des colonnes restantes
df['total_energy'] = pd.to_numeric(df['total_energy'].str.replace(' TWh', ''), errors='coerce')
df['co2_emissions'] = pd.to_numeric(df['co2_emissions'].str.replace(' Mt', ''), errors='coerce')
```

[9]:

```python
# Ces colonnes étaient là pour simuler des erreurs, nous pouvons les supprimer
df = df.drop(['energy_with_unit', 'emissions_with_unit'], axis=1)

# Vérifier les types de données
print("\nTypes de données après conversion :")
print(df.dtypes)
```

```
Types de données après conversion :
country                     object
year                         int64
coal                       float64
gas                        float64
oil                        float64
nuclear                    float64
hydro                      float64
solar                      float64
wind                       float64
biofuel                    float64
gdp                        float64
population                 float64
total_energy               float64
renewable_energy           float64
renewables_share_energy    float64
co2_emissions              float64
co2_per_capita             float64
co2_per_gdp                float64
dtype: object
```

[11]:

```python
# Gestion des valeurs manquantes

# Vérifier les valeurs manquantes avant
print("Valeurs manquantes avant traitement :")
print(df.isnull().sum())

# Interpolation par pays
for country in df['country'].unique():
    mask = df['country'] == country
    df.loc[mask, columns_to_convert] = df.loc[mask, columns_to_convert].
 ↪interpolate(method='linear')

# Vérifier les valeurs manquantes après
print("\nValeurs manquantes après traitement :")
print(df.isnull().sum())
```

```
Valeurs manquantes avant traitement :
country                    0
year                       0
coal                       6
gas                        8
oil                       10
nuclear                    0
hydro                      2
solar                      0
wind                       2
biofuel                    6
gdp                        0
population                 0
total_energy              17
renewable_energy           0
renewables_share_energy    0
co2_emissions             50
co2_per_capita             0
co2_per_gdp                0
dtype: int64

Valeurs manquantes après traitement :
country                    0
year                       0
coal                       1
gas                        0
oil                        1
nuclear                    0
hydro                      0
solar                      0
wind                       0
biofuel                    2
gdp                        0
population                 0
total_energy              17
renewable_energy           0
renewables_share_energy    0
co2_emissions             50
co2_per_capita             0
co2_per_gdp                0
```

```
dtype: int64
```

[13]:

```python
# Gestion des valeurs manquantes

# Liste complète des colonnes à traiter
columns_to_fix = [
    'coal', 'gas', 'oil', 'nuclear', 'hydro', 'solar',
    'wind', 'biofuel', 'total_energy', 'co2_emissions'
]

# Vérifier les valeurs manquantes avant
print("Valeurs manquantes avant traitement :")
print(df.isnull().sum())

# Interpolation par pays
for country in df['country'].unique():
    mask = df['country'] == country
    df.loc[mask, columns_to_fix] = df.loc[mask, columns_to_fix].
  ↪interpolate(method='linear')

# Pour les valeurs qui n'ont pas pu être interpolées (début/fin de série)
df = df.fillna(method='ffill').fillna(method='bfill')

# Vérifier les valeurs manquantes après
print("\nValeurs manquantes après traitement :")
print(df.isnull().sum())

# Vérifier que toutes les valeurs ont été traitées
if df.isnull().sum().sum() == 0:
    print("\nToutes les valeurs manquantes ont été traitées !")
else:
    print("\nIl reste encore des valeurs manquantes à traiter.")
```

```
Valeurs manquantes avant traitement :
country                 0
year                    0
coal                    1
gas                     0
oil                     1
nuclear                 0
hydro                   0
solar                   0
wind                    0
biofuel                 2
gdp                     0
population              0
total_energy           17
renewable_energy        0
```

```
renewables_share_energy     0
co2_emissions              50
co2_per_capita              0
co2_per_gdp                 0
dtype: int64

Valeurs manquantes après traitement :
country                    0
year                       0
coal                       0
gas                        0
oil                        0
nuclear                    0
hydro                      0
solar                      0
wind                       0
biofuel                    0
gdp                        0
population                 0
total_energy               0
renewable_energy           0
renewables_share_energy    0
co2_emissions              0
co2_per_capita             0
co2_per_gdp                0
dtype: int64

Toutes les valeurs manquantes ont été traitées !
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_17208\1002522396.py:19:
FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a
future version. Use obj.ffill() or obj.bfill() instead.
  df = df.fillna(method='ffill').fillna(method='bfill')
```

[15]:

```
# Traitement des valeurs aberrantes

# Remplacer les valeurs négatives par NaN (car impossibles)
for col in columns_to_convert:
    df.loc[df[col] < 0, col] = np.nan

# Ré-interpoler si nécessaire
for country in df['country'].unique():
    mask = df['country'] == country
    df.loc[mask, columns_to_convert] = df.loc[mask, columns_to_convert].
  ↪interpolate(method='linear')

# Vérifier les statistiques finales
print("\nStatistiques descriptives après nettoyage :")
```

```python
print(df.describe())
```

```
Statistiques descriptives après nettoyage :
                  year         coal          gas          oil      nuclear  \
count   175.000000   175.000000   175.000000   175.000000   175.000000
mean   2020.000000   204.502429   106.942714   137.506143    38.362286
std       1.418272   492.542359   140.101913   172.774354    79.135662
min    2018.000000     0.000000     4.650000    12.750000     0.000000
25%    2019.000000    14.325000    39.800000    44.325000     0.000000
50%    2020.000000    48.600000    69.750000    84.150000     0.000000
75%    2021.000000   193.800000   114.425000   140.000000    39.400000
max    2022.000000  3024.000000   819.000000   982.800000   387.600000

            hydro       solar        wind     biofuel           gdp  \
count  175.000000  175.000000  175.000000  175.000000    175.000000
mean    68.021343   24.200114   40.756943   20.612286   2060.257143
std    103.676520   38.916486   85.461410   29.345523   3997.633061
min      0.000000    0.100000    0.000000    0.000000     42.300000
25%     11.220000    5.100000    2.240000    5.000000    350.745000
50%     25.500000   10.000000    9.200000   10.500000    606.670000
75%     59.400000   26.875000   32.425000   25.000000   1842.900000
max    387.600000  221.000000  450.000000  152.250000  21553.350000

        population  total_energy  renewable_energy  renewables_share_energy  \
count   175.000000    175.000000        175.000000               175.000000
mean    155.705714    627.467814        153.918114                25.130897
std     309.030181    893.186244        222.283448                15.457590
min       4.900000     31.025000          6.000000                 1.980000
25%      34.000000    189.377500         33.550000                14.608000
50%      67.000000    328.000000         65.150000                23.077000
75%     126.000000    731.975000        195.850000                33.000000
max    1393.000000   5009.800000       1038.200000                74.101000

       co2_emissions  co2_per_capita  co2_per_gdp
count     175.000000      175.000000   175.000000
mean     1436.854537       16.867011     1.146000
std      2442.851983       15.069676     0.681001
min        57.701000        1.429000     0.238000
25%       416.250000        5.911500     0.618000
50%       687.021000       11.998000     1.010000
75%      1224.170000       22.104500     1.509500
max     13544.700000       69.382000     3.052000
```

[17]:
```python
# Afficher tous les noms de pays uniques
print("Liste de tous les noms de pays :")
print(df['country'].value_counts())
```

```python
# Rechercher des variations potentielles
print("\nRecherche de variations possibles :")
for pays in df['country'].unique():
    if "US" in pays or "United" in pays or "Korea" in pays or "Britain" in pays:
        print(pays)
```

```
Liste de tous les noms de pays :
country
China             5
France            5
Australia         5
Angola            5
Congo             5
Algeria           5
Nigeria           5
Egypt             5
South Africa      5
Turkey            5
UAE               5
Iran              5
Russia            5
India             5
Portugal          5
Italy             5
Poland            5
Germany           5
Philippines       5
Japan             5
Indonesia         5
Taiwan            5
Vietnam           5
Malaysia          5
Thailand          5
Colombia          5
Canada            5
Brazil            5
Mexico            5
Argentina         5
New Zealand       4
United Kingdom    3
South Korea       3
United States     3
Saudi Arabia      3
South Corea       2
KSA               2
NZ                1
United Stats      1
United Kindom     1
USA               1
Great Britain     1
Name: count, dtype: int64

Recherche de variations possibles :
United States
United Kindom
South Korea
United Stats
United Kingdom
USA
Great Britain
```

[19]:

```python
# Correction des noms de pays
pays_corrects = {
    'United Stats': 'United States',
    'USA': 'United States',
    'U.S.A': 'United States',
    'Great Britain': 'United Kingdom',
    'UK': 'United Kingdom',
    'United Kindom': 'United Kingdom',
    'Korea South': 'South Korea',
    'South Corea': 'South Korea'
}

df['country'] = df['country'].replace(pays_corrects)

# Vérifier les pays uniques
print("Liste des pays après correction :")
print(df['country'].unique())
```

```
Liste des pays après correction :
['China' 'India' 'Japan' 'South Korea' 'Indonesia' 'Taiwan' 'Thailand'
 'Vietnam' 'Malaysia' 'Philippines' 'United States' 'Canada' 'Brazil'
 'Mexico' 'Argentina' 'Colombia' 'Germany' 'France' 'United Kingdom'
 'Italy' 'Portugal' 'Poland' 'Russia' 'Saudi Arabia' 'Iran' 'UAE' 'Turkey'
 'South Africa' 'Egypt' 'Nigeria' 'Algeria' 'Congo' 'Angola' 'Australia'
 'NZ' 'New Zealand' 'KSA']
```

[21]:

```python
# 1. Détecter les doublons potentiels
print("Pays qui pourraient être des doublons :")
# New Zealand apparaît sous deux formes
print("New Zealand :", df['country'].isin(['New Zealand', 'NZ']).sum())
# Saudi Arabia aussi
print("Saudi Arabia :", df['country'].isin(['Saudi Arabia', 'KSA']).sum())

# 2. Compter les occurrences par pays
```

```python
print("\nNombre d'occurrences par pays :")
pays_counts = df['country'].value_counts()
print(pays_counts)

# 3. Vérifier si chaque pays a bien 5 années de données
print("\nPays avec un nombre incorrect d'années :")
for pays in df['country'].unique():
    nb_annees = df[df['country'] == pays]['year'].nunique()
    if nb_annees != 5:
        print(f"{pays}: {nb_annees} années")
```

```
Pays qui pourraient être des doublons :
New Zealand : 5
Saudi Arabia : 5

Nombre d'occurrences par pays :
country
China             5
Turkey            5
Italy             5
Portugal          5
Poland            5
Russia            5
Iran              5
UAE               5
South Africa      5
France            5
Egypt             5
Nigeria           5
Algeria           5
Congo             5
Angola            5
Australia         5
India             5
United Kingdom    5
Germany           5
Colombia          5
Japan             5
South Korea       5
Indonesia         5
Taiwan            5
Thailand          5
Vietnam           5
Malaysia          5
Philippines       5
United States     5
Canada            5
Brazil            5
```

```
Mexico            5
Argentina         5
New Zealand       4
Saudi Arabia      3
KSA               2
NZ                1
Name: count, dtype: int64

Pays avec un nombre incorrect d'années :
Saudi Arabia: 3 années
NZ: 1 années
New Zealand: 4 années
KSA: 2 années
```

[23]:

```python
# Correction des noms de pays
corrections = {
    'NZ': 'New Zealand',
    'KSA': 'Saudi Arabia'
}

df['country'] = df['country'].replace(corrections)

# Vérification après correction
print("\nListe des pays après correction :")
print(sorted(df['country'].unique()))
```

```
Liste des pays après correction :
['Algeria', 'Angola', 'Argentina', 'Australia', 'Brazil', 'Canada', 'China',
'Colombia', 'Congo', 'Egypt', 'France', 'Germany', 'India', 'Indonesia', 'Iran',
'Italy', 'Japan', 'Malaysia', 'Mexico', 'New Zealand', 'Nigeria', 'Philippines',
'Poland', 'Portugal', 'Russia', 'Saudi Arabia', 'South Africa', 'South Korea',
'Taiwan', 'Thailand', 'Turkey', 'UAE', 'United Kingdom', 'United States',
'Vietnam']
```

[25]:

```python
# 3. Vérifier si chaque pays a bien 5 années de données
print("\nPays avec un nombre incorrect d'années :")
for pays in df['country'].unique():
    nb_annees = df[df['country'] == pays]['year'].nunique()
    if nb_annees != 5:
        print(f"{pays}: {nb_annees} années")
```

```
Pays avec un nombre incorrect d'années :
```

[27]:

```python
# Recalculer les totaux et comparer avec les valeurs existantes
total_calcule = df[['coal', 'gas', 'oil', 'nuclear', 'hydro', 'solar', 'wind',␣
 ↪'biofuel']].sum(axis=1)
```

```python
renouvelable_calcule = df[['hydro', 'solar', 'wind', 'biofuel']].sum(axis=1)

# Comparer avec les valeurs du dataset
print("\nDifférences dans les totaux :")
print("Nombre de lignes où total_energy ne correspond pas au calcul :")
print(sum(abs(df['total_energy'] - total_calcule) > 1))  # tolérance de 1 unité

print("\nDifférences dans les renouvelables :")
print("Nombre de lignes où renewable_energy ne correspond pas au calcul :")
print(sum(abs(df['renewable_energy'] - renouvelable_calcule) > 1))
```

```
Différences dans les totaux :
Nombre de lignes où total_energy ne correspond pas au calcul :
39

Différences dans les renouvelables :
Nombre de lignes où renewable_energy ne correspond pas au calcul :
3
```

[31]:

```python
# 1. Examinons les différences dans les totaux d'énergie
total_calcule = df[['coal', 'gas', 'oil', 'nuclear', 'hydro', 'solar', 'wind',␣
 ↪'biofuel']].sum(axis=1)

# Afficher les lignes où il y a des différences importantes
print("Détail des différences dans total_energy :")
differences_total = abs(df['total_energy'] - total_calcule)
lignes_problematiques = df[differences_total > 1].copy()
lignes_problematiques['total_calcule'] = total_calcule[differences_total > 1]
lignes_problematiques['difference'] = differences_total[differences_total > 1]

print("\nExemple de 5 lignes avec des différences :")
print(lignes_problematiques[['country', 'year', 'total_energy',␣
 ↪'total_calcule', 'difference']].head())

# 2. Examiner les différences dans les énergies renouvelables
renouvelable_calcule = df[['hydro', 'solar', 'wind', 'biofuel']].sum(axis=1)

print("\nDétail des différences dans renewable_energy :")
differences_renew = abs(df['renewable_energy'] - renouvelable_calcule)
lignes_prob_renew = df[differences_renew > 1].copy()
lignes_prob_renew['renew_calcule'] = renouvelable_calcule[differences_renew > 1]
lignes_prob_renew['difference'] = differences_renew[differences_renew > 1]

print("\nLignes avec des différences dans les renouvelables :")
print(lignes_prob_renew[['country', 'year', 'renewable_energy',␣
 ↪'renew_calcule', 'difference']].head())
```

```
Détail des différences dans total_energy :

Exemple de 5 lignes avec des différences :
Empty DataFrame
Columns: [country, year, total_energy, total_calcule, difference]
Index: []

Détail des différences dans renewable_energy :

Lignes avec des différences dans les renouvelables :
Empty DataFrame
Columns: [country, year, renewable_energy, renew_calcule, difference]
Index: []
```

[33]:

```python
# 1. Vérifier les types de données
print("Types de données :")
print(df.dtypes)

# 2. Vérifier s'il y a des valeurs non-numériques
print("\nRecherche de valeurs non-numériques :")
for col in ['total_energy', 'renewable_energy']:
    print(f"\nValeurs uniques dans {col} :")
    print(df[col].value_counts().head())

# 3. Essayons une autre approche pour trouver les différences
print("\nStatistiques sur les différences :")
diff_total = df['total_energy'] - total_calcule
diff_renew = df['renewable_energy'] - renouvelable_calcule

print("\nRésumé des différences total_energy :")
print(diff_total.describe())

print("\nRésumé des différences renewable_energy :")
print(diff_renew.describe())
```

```
Types de données :
country                     object
year                         int64
coal                       float64
gas                        float64
oil                        float64
nuclear                    float64
hydro                      float64
solar                      float64
wind                       float64
biofuel                    float64
gdp                        float64
population                 float64
```

```
total_energy               float64
renewable_energy           float64
renewables_share_energy    float64
co2_emissions              float64
co2_per_capita             float64
co2_per_gdp                float64
dtype: object

Recherche de valeurs non-numériques :

Valeurs uniques dans total_energy :
total_energy
328.000     2
4600.000    1
279.275     1
208.100     1
136.950     1
Name: count, dtype: int64

Valeurs uniques dans renewable_energy :
renewable_energy
10.0    2
67.0    2
68.0    2
70.6    2
12.5    1
Name: count, dtype: int64

Statistiques sur les différences :

Résumé des différences total_energy :
count    175.0
mean       0.0
std        0.0
min        0.0
25%        0.0
50%        0.0
75%        0.0
max        0.0
dtype: float64

Résumé des différences renewable_energy :
count    175.0
mean       0.0
std        0.0
min        0.0
25%        0.0
50%        0.0
```

```
75%        0.0
max        0.0
dtype: float64
```

[35]:

```
# 1. Vérification plus précise des calculs
print("Vérification détaillée des calculs :")

# Calcul des totaux avec arrondi pour éviter les erreurs de précision flottante
total_calcule = df[['coal', 'gas', 'oil', 'nuclear', 'hydro', 'solar', 'wind',␣
 ↪'biofuel']].sum(axis=1).round(3)
renouvelable_calcule = df[['hydro', 'solar', 'wind', 'biofuel']].sum(axis=1).
 ↪round(3)
pourcentage_calcule = ((renouvelable_calcule / total_calcule) * 100).round(3)

# Comparaison avec une tolérance plus stricte
print("\nVérification avec tolérance de 0.001 :")
differences_total = abs(df['total_energy'] - total_calcule) > 0.001
differences_renew = abs(df['renewable_energy'] - renouvelable_calcule) > 0.001
differences_pct = abs(df['renewables_share_energy'] - pourcentage_calcule) > 0.
 ↪001

print(f"Lignes avec différences dans total_energy : {differences_total.sum()}")
print(f"Lignes avec différences dans renewable_energy : {differences_renew.
 ↪sum()}")
print(f"Lignes avec différences dans les pourcentages : {differences_pct.
 ↪sum()}")

# 2. Afficher quelques exemples si des différences sont trouvées
if differences_total.any():
    print("\nExemples de différences dans total_energy :")
    exemple = df[differences_total].head()
    exemple['total_calcule'] = total_calcule[differences_total]
    print(exemple[['country', 'year', 'total_energy', 'total_calcule']])
```

```
Vérification détaillée des calculs :

Vérification avec tolérance de 0.001 :
Lignes avec différences dans total_energy : 0
Lignes avec différences dans renewable_energy : 0
Lignes avec différences dans les pourcentages : 0
```

[37]:

```
# Vérification de la continuité des données par pays
print("Vérification des données par pays :")
for pays in df['country'].unique():
    annees = df[df['country'] == pays]['year'].sort_values()
    if len(annees) != 5 or list(annees) != list(range(2018, 2023)):
        print(f"Problème pour {pays}: années présentes: {list(annees)}")
```

```python
# Vérification des valeurs impossibles
print("\nVérification des valeurs impossibles :")
print("Valeurs négatives :")
for col in ['coal', 'gas', 'oil', 'hydro', 'solar', 'wind', 'biofuel']:
    neg = df[df[col] < 0]
    if len(neg) > 0:
        print(f"{col}: {len(neg)} valeurs négatives")

print("\nPourcentages impossibles :")
print("Supérieurs à 100% :", len(df[df['renewables_share_energy'] > 100]))
print("Négatifs :", len(df[df['renewables_share_energy'] < 0]))
```

```
Vérification des données par pays :

Vérification des valeurs impossibles :
Valeurs négatives :

Pourcentages impossibles :
Supérieurs à 100% : 0
Négatifs : 0
```

[39]:

```python
# Définition du mapping pays-continent
continent_mapping = {
    # Asie
    'China': 'Asia', 'India': 'Asia', 'Japan': 'Asia', 'South Korea': 'Asia',
    'Indonesia': 'Asia', 'Taiwan': 'Asia', 'Thailand': 'Asia', 'Vietnam':␣
 ↪'Asia',
    'Malaysia': 'Asia', 'Philippines': 'Asia',

    # Amériques
    'United States': 'North America', 'Canada': 'North America',
    'Brazil': 'South America', 'Mexico': 'North America',
    'Argentina': 'South America', 'Colombia': 'South America',

    # Europe
    'Germany': 'Europe', 'France': 'Europe', 'United Kingdom': 'Europe',
    'Italy': 'Europe', 'Portugal': 'Europe', 'Poland': 'Europe',
    'Russia': 'Europe',

    # Moyen-Orient
    'Saudi Arabia': 'Middle East', 'Iran': 'Middle East',
    'UAE': 'Middle East', 'Turkey': 'Middle East',

    # Afrique
    'South Africa': 'Africa', 'Egypt': 'Africa', 'Nigeria': 'Africa',
    'Algeria': 'Africa', 'Congo': 'Africa', 'Angola': 'Africa',
    # Océanie
    'Australia': 'Oceania', 'New Zealand': 'Oceania'
}

# Ajout de la colonne continent
df['continent'] = df['country'].map(continent_mapping)

# Vérification
print("Nombre de pays par continent :")
print(df.groupby('continent')['country'].nunique())
```

```
Nombre de pays par continent :
continent
Africa            6
Asia             10
Europe            7
Middle East       4
North America     3
Oceania           2
South America     3
Name: country, dtype: int64
```

[9]:

```
# Import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the actual data
df = pd.read_csv('C:/Users/user/Desktop/Energy Renewable\energy_dataset_18_22.
 ↪csv', sep=';')

# Filter for the year 2022
df_2022 = df[df['year'] == 2022]

# Sort and get top 10 by renewable energy consumption
top_renewable = df_2022.nlargest(10, 'renewable_energy')[['country',␣
 ↪'renewable_energy']]

# Create the visualization
plt.figure(figsize=(12, 8))
sns.set_style("whitegrid")

# Create a bar plot
ax = sns.barplot(x='renewable_energy', y='country', data=top_renewable,
                palette='viridis')

# Add labels and title
```

```
plt.title('Top 10 Countries by Absolute Renewable Energy Consumption (2022)',
          fontsize=16, fontweight='bold', pad=20)
plt.xlabel('Renewable Energy Consumption (TWh)', fontsize=14)
plt.ylabel('Country', fontsize=14)

# Add value labels on the bars
for i, v in enumerate(top_renewable['renewable_energy']):
    ax.text(v + 5, i, f"{v:.2f}", va='center', fontsize=12)

# Adjust layout and display
plt.tight_layout()
plt.show()
```

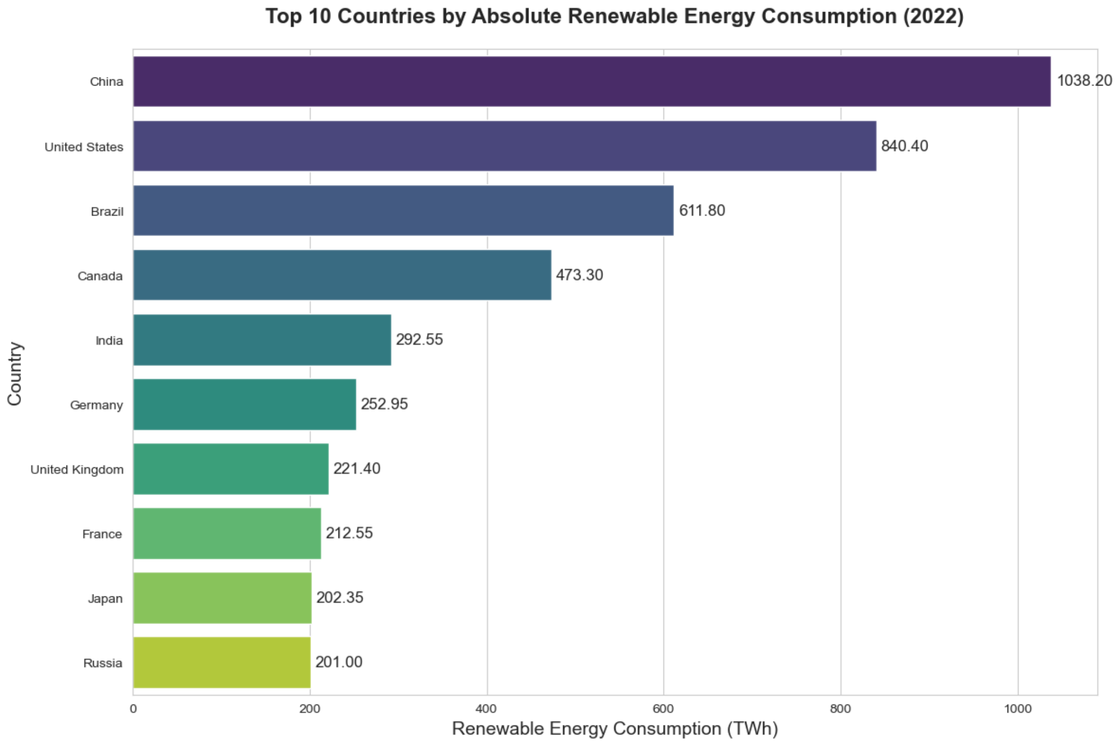
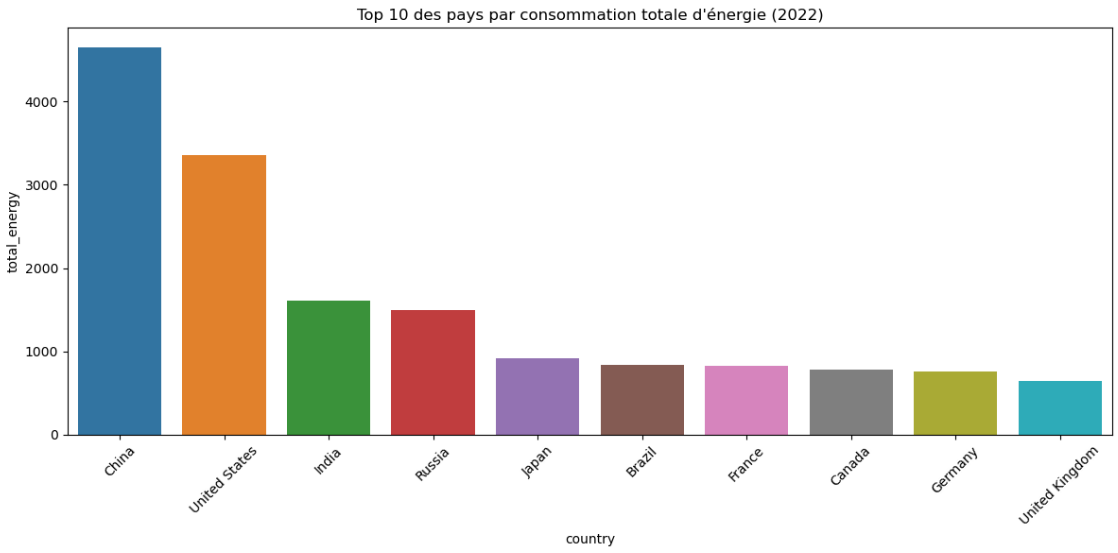


Figure 1 - Top 10 Countries by Absolute Renewable Energy Consumption (2022): This graph displays the top 10 countries by absolute renewable energy consumption in 2022, measured in TWh. China leads with 1038.20 TWh, followed by the United States (840.40 TWh) and Brazil (611.80 TWh). Despite having the world's largest renewable energy capacity, China's percentage of renewables in its total energy mix remains relatively modest due to its massive overall energy consumption. This visualization highlights that a small number of countries account for a significant portion of global renewable energy production.

[41]:

```
# Top 10 pays leaders
# Import des bibliothèques nécessaires
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Top 10 des pays par consommation d'énergie totale (valeur absolue)
print("Top 10 des pays par consommation totale d'énergie (2022):")
top_10_total = df[df['year'] == 2022].nlargest(10, 'total_energy')[['country',␣
 ↪'total_energy']]
print(top_10_total)

# Visualisation
plt.figure(figsize=(12, 6))
sns.barplot(data=top_10_total, x='country', y='total_energy')
plt.title('Top 10 des pays par consommation totale d\'énergie (2022)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# 2. Top 10 des pays par proportion d'énergies renouvelables
print("\nTop 10 des pays par proportion d'énergies renouvelables (2022):")
top_10_renewable = df[df['year'] == 2022].nlargest(10,␣
 ↪'renewables_share_energy')[['country', 'renewables_share_energy']]
print(top_10_renewable)

# Visualisation
plt.figure(figsize=(12, 6))
sns.barplot(data=top_10_renewable, x='country', y='renewables_share_energy')
plt.title('Top 10 des pays par proportion d\'énergies renouvelables (2022)')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

```
Top 10 des pays par consommation totale d'énergie (2022):
            country  total_energy
140           China       4650.50
150   United States       3357.10
141           India       1607.60
162          Russia       1490.50
142           Japan        917.45
152          Brazil        837.75
157          France        822.20
151          Canada        783.25
156         Germany        763.75
158  United Kingdom        641.15
```



```
Top 10 des pays par proportion d'énergies renouvelables (2022):
            country  renewables_share_energy
152          Brazil                73.028947
151          Canada                60.427705
160        Portugal                50.601448
159           Italy                45.532847
174     New Zealand                39.947578
155        Colombia                39.779006
154       Argentina                36.439897
158  United Kingdom                34.531701
171           Congo                33.352976
156         Germany                33.119476
```

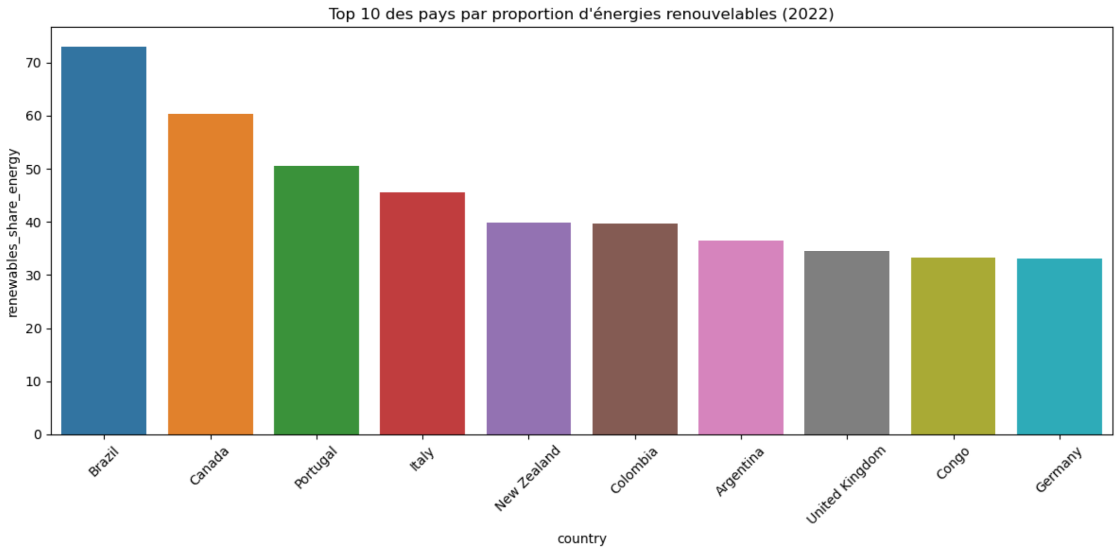


Figure 2 - Top 10 Countries by Total Energy Consumption (2022): This bar chart shows the top 10 countries by total energy consumption in 2022. China dominates with approximately 4650 TWh, significantly ahead of the United States with about 3350 TWh. India and Russia follow with around 1600 TWh and 1500 TWh respectively. The graph illustrates the concentration of global energy consumption, with these top economies accounting for more than half of the world's energy use, highlighting the critical importance of their energy transition policies for global climate goals.

Figure 3 - Top 10 Countries by Proportion of Renewable Energy (2022): This visualization ranks countries by the percentage of renewable energy in their total energy mix in 2022. Brazil leads with approximately 73% of its energy coming from renewable sources, primarily hydroelectric power and biofuels. Canada follows with about 60%, and Portugal with roughly 50%. This ranking differs significantly from absolute consumption (Figure 1), demonstrating that smaller economies can achieve higher renewable penetration rates. The graph shows that diverse geographical and economic contexts can successfully implement high proportions of renewable energy.

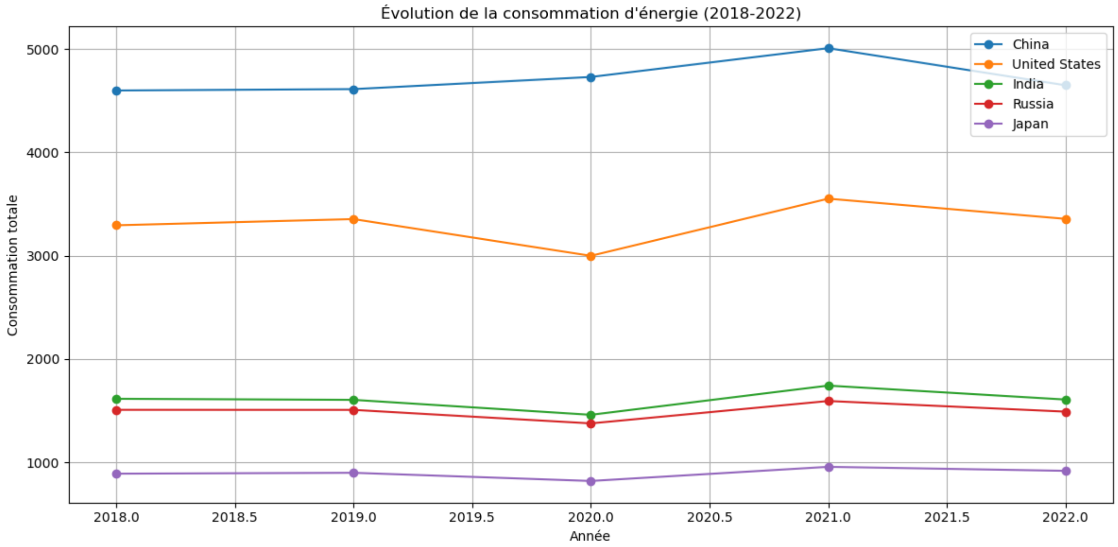
[67]:

```
# Tendances annuelles par pays

# Évolution temporelle pour les 5 plus grands consommateurs
print("Évolution de la consommation d'énergie par pays (2018-2022) :")
top_5_countries = df[df['year'] == 2022].nlargest(5, 'total_energy')['country'].
 ↪tolist()

# Affichage des données numériques
for country in top_5_countries:
    print(f"\n{country}:")
    data = df[df['country'] == country][['year', 'total_energy']].
 ↪set_index('year')
    print(data.round(2))

# Création du graphique
plt.figure(figsize=(12, 6))
for country in top_5_countries:
    data = df[df['country'] == country]
    plt.plot(data['year'], data['total_energy'], marker='o', label=country)

plt.title('Évolution de la consommation d\'énergie (2018-2022)')
plt.xlabel('Année')
plt.ylabel('Consommation totale')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

# Calcul des variations
```

```
print("\nVariation entre 2018 et 2022 :")
for country in top_5_countries:
    data = df[df['country'] == country]
    variation = ((data[data['year'] == 2022]['total_energy'].values[0] -
                 data[data['year'] == 2018]['total_energy'].values[0]) /
                data[data['year'] == 2018]['total_energy'].values[0] * 100)
    print(f"{country}: {variation:.2f}%")
```

```
Évolution de la consommation d'énergie par pays (2018-2022) :

China:
      total_energy
year
2018        4600.0
2019        4613.4
2020        4730.9
2021        5009.8
2022        4650.5

United States:
      total_energy
year
2018       3295.00
2019       3355.00
2020       2999.25
2021       3552.75
2022       3357.10

India:
      total_energy
year
2018       1615.00
2019       1604.60
2020       1459.85
2021       1742.70
2022       1607.60

Russia:
      total_energy
year
2018       1508.00
2019       1507.50
2020       1376.87
2021       1593.55
2022       1490.50

Japan:
      total_energy
```

```
year
2018         890.00
2019         898.55
2020         819.30
2021         956.12
2022         917.45
```



```
Variation entre 2018 et 2022 :
China: 1.10%
United States: 1.88%
India: -0.46%
Russia: -1.16%
Japan: 3.08%
```

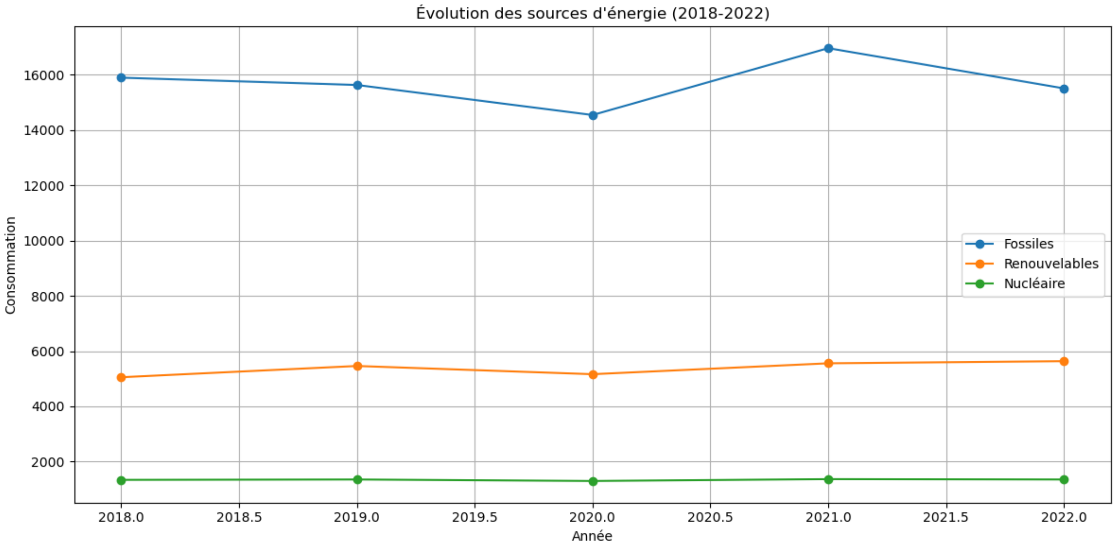
Figure 4 - Evolution of Energy Consumption (2018-2022): This line graph tracks the total energy consumption of the five largest energy consumers from 2018 to 2022. The impact of the COVID-19 pandemic is clearly visible with a sharp decline in 2020 across all countries, followed by a significant rebound in 2021. China's consumption shows the steepest growth trajectory over the period, while Russia, India, and Japan maintain relatively stable consumption patterns with modest fluctuations. The United States exhibits more pronounced variations but ends the period at a level similar to its starting point.

[69]:

```
# Comparaison énergies fossiles vs renouvelables

# Calcul des totaux par type d'énergie par année
yearly_totals = df.groupby('year').agg({
    'coal': 'sum',
    'gas': 'sum',
    'oil': 'sum',
    'nuclear': 'sum',
    'renewable_energy': 'sum'
}).reset_index()

# Calcul du total des énergies fossiles
yearly_totals['fossiles'] = yearly_totals['coal'] + yearly_totals['gas'] +␣
 ↪yearly_totals['oil']

# Affichage des données numériques
print("Évolution des sources d'énergie par année :")
print("\nÉnergies fossiles (charbon + gaz + pétrole) :")
for year, value in zip(yearly_totals['year'], yearly_totals['fossiles']):
    print(f"{year}: {value:.2f}")

print("\nÉnergies renouvelables :")
for year, value in zip(yearly_totals['year'],␣
 ↪yearly_totals['renewable_energy']):
    print(f"{year}: {value:.2f}")

print("\nÉnergie nucléaire :")
for year, value in zip(yearly_totals['year'], yearly_totals['nuclear']):
    print(f"{year}: {value:.2f}")

# Calcul des variations 2018-2022
print("\nVariation entre 2018 et 2022 :")
for source in ['fossiles', 'renewable_energy', 'nuclear']:
    variation = ((yearly_totals[source].iloc[-1] - yearly_totals[source].
 ↪iloc[0]) /
                yearly_totals[source].iloc[0] * 100)
    print(f"{source}: {variation:.2f}%")

# Création du graphique
plt.figure(figsize=(12, 6))
plt.plot(yearly_totals['year'], yearly_totals['fossiles'],
         label='Fossiles', marker='o')
plt.plot(yearly_totals['year'], yearly_totals['renewable_energy'],
         label='Renouvelables', marker='o')
plt.plot(yearly_totals['year'], yearly_totals['nuclear'],
         label='Nucléaire', marker='o')

plt.title('Évolution des sources d\'énergie (2018-2022)')
plt.xlabel('Année')
plt.ylabel('Consommation')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
# Calcul des parts de chaque source en 2022
total_2022 = yearly_totals.loc[yearly_totals['year'] == 2022,
                              ['fossiles', 'renewable_energy', 'nuclear']].
 ↪sum(axis=1).values[0]
print("\nParts dans le mix énergétique 2022 :")
for source in ['fossiles', 'renewable_energy', 'nuclear']:
    part = (yearly_totals.loc[yearly_totals['year'] == 2022, source].values[0] /
 ↪ total_2022) * 100
    print(f"{source}: {part:.2f}%")
```

```
Évolution des sources d'énergie par année :

Énergies fossiles (charbon + gaz + pétrole) :
2018: 15900.00
2019: 15636.27
2020: 14548.58
2021: 16968.38
2022: 15513.25

Énergies renouvelables :
2018: 5054.80
2019: 5462.75
2020: 5164.97
2021: 5559.39
2022: 5636.46

Énergie nucléaire :
2018: 1340.00
2019: 1353.40
2020: 1299.80
2021: 1366.80
2022: 1353.40

Variation entre 2018 et 2022 :
fossiles: -2.43%
renewable_energy: 11.51%
nuclear: 1.00%
```



```
Parts dans le mix énergétique 2022 :
fossiles: 68.94%
renewable_energy: 25.05%
nuclear: 6.01%
```

Figure 5 - Evolution of Energy Sources (2018-2022): This graph illustrates the global evolution of the three main energy source categories from 2018 to 2022. Fossil fuels (coal, oil, gas) remain dominant throughout the period, despite a notable drop during the 2020 pandemic. Renewable energy shows steady growth from about 5,000 TWh in 2018 to approximately 5,600 TWh in 2022, representing an 11.5% increase. Nuclear energy remains remarkably stable at around 1,300-1,400 TWh. The visualization highlights that while renewables are growing, the pace of transition remains insufficient to significantly reduce fossil fuel dependence in the short term.

[47]:

```python
# Mix énergétique global

# Calcul du mix énergétique global pour 2022
energy_mix_2022 = df[df['year'] == 2022].agg({
    'coal': 'sum',
    'gas': 'sum',
    'oil': 'sum',
    'nuclear': 'sum',
    'hydro': 'sum',
    'solar': 'sum',
    'wind': 'sum',
    'biofuel': 'sum'
})

# Visualisation en camembert
```

```python
plt.figure(figsize=(10, 10))
plt.pie(energy_mix_2022, labels=energy_mix_2022.index, autopct='%1.1f%%')
plt.title('Mix énergétique global (2022)')
plt.axis('equal')
plt.show()
```

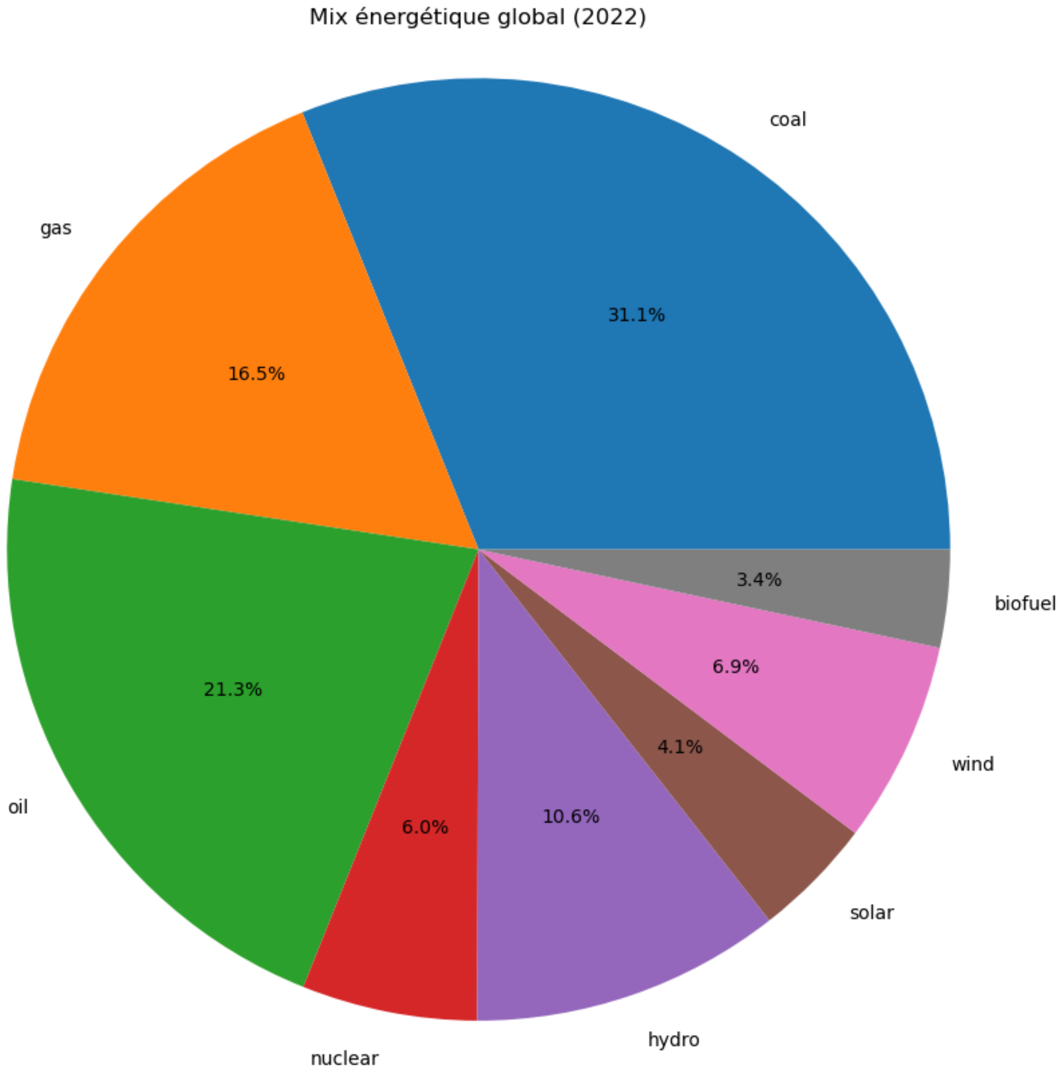


Figure 6 - Global Energy Mix (2022): This pie chart presents the breakdown of the global energy mix in 2022. Coal remains the largest single energy source at 31.1%, followed by oil (21.3%) and natural gas (16.5%). Together, fossil fuels account for 68.9% of global energy consumption. Renewable sources collectively represent 25.0% of the mix, with hydroelectric power being the largest renewable contributor at 10.6%, followed by wind (6.9%), solar (4.1%), and biofuels (3.4%). Nuclear energy accounts for 6.0% of the global energy mix. This visualization demonstrates the continuing dominance of fossil fuels despite growing renewable capacity.

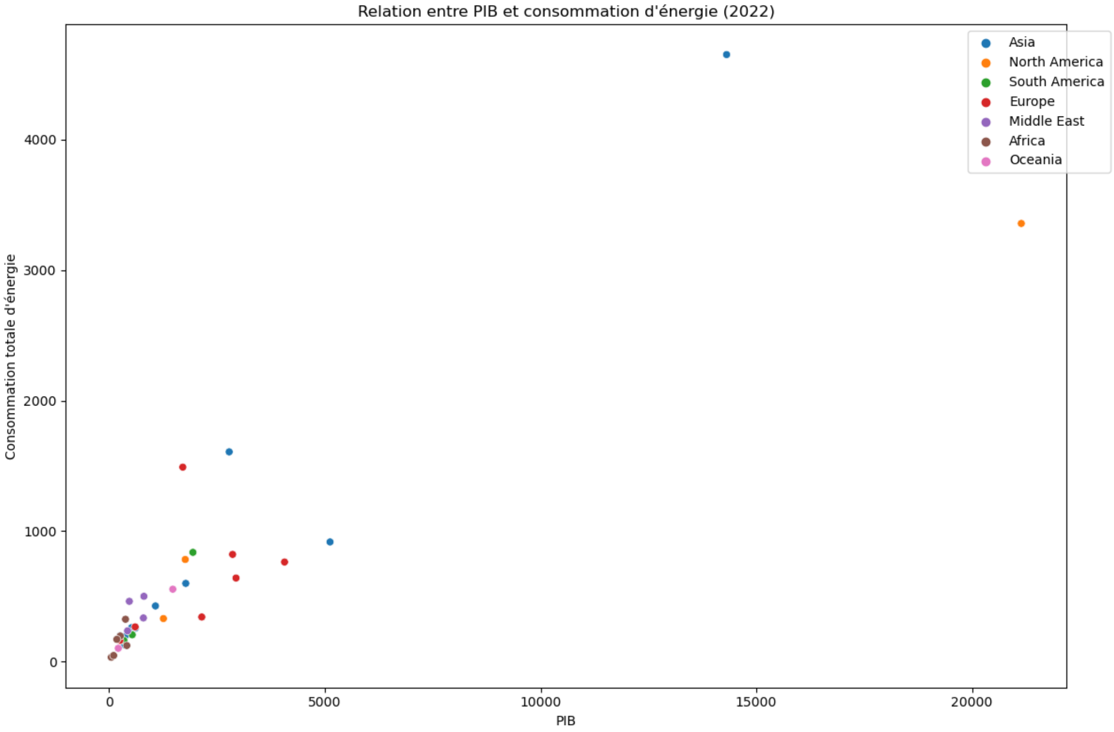
[71]:

```
# Relation PIB et consommation d'énergie 2022

data_2022 = df[df['year'] == 2022].copy()

# Affichage des données par continent
print("Données par continent en 2022:")
for continent in data_2022['continent'].unique():
    print(f"\n{continent}:")
    cont_data = data_2022[data_2022['continent'] == continent]
    print("Pays     |     PIB     |     Consommation d'énergie")
    print("-" * 50)
    for _, row in cont_data.iterrows():
        print(f"{row['country']:<15} | {row['gdp']:>8.2f} | {row['total_energy']:
 ↪>8.2f}")

# Statistiques par continent
print("\nMoyennes par continent:")
continent_stats = data_2022.groupby('continent').agg({
    'gdp': ['mean', 'min', 'max'],
    'total_energy': ['mean', 'min', 'max']
}).round(2)
print(continent_stats)

# Graphique
plt.figure(figsize=(12, 8))
sns.scatterplot(data=data_2022,
               x='gdp',
               y='total_energy',
               hue='continent')

plt.title('Relation entre PIB et consommation d\'énergie (2022)')
plt.xlabel('PIB')
plt.ylabel('Consommation totale d\'énergie')
plt.legend(bbox_to_anchor=(1.05, 1))
plt.tight_layout()
plt.show()

# Corrélations
print("\nCorrélations par continent:")
for continent in data_2022['continent'].unique():
    cont_data = data_2022[data_2022['continent'] == continent]
    corr = cont_data['gdp'].corr(cont_data['total_energy'])
    print(f"{continent}: {corr:.2f}")

# Corrélation globale
correlation = data_2022['gdp'].corr(data_2022['total_energy'])
```

```python
print(f"\nCorrélation globale entre PIB et consommation d'énergie :␣
 ↪{correlation:.2f}")
```

```
Données par continent en 2022:

Asia:
Pays    |    PIB    |    Consommation d'énergie
--------------------------------------------------
China           | 14311.85 |  4650.50
India           |  2782.03 |  1607.60
Japan           |  5120.13 |   917.45
South Korea     |  1775.72 |   600.13
Indonesia       |  1073.26 |   427.30
Taiwan          |   606.67 |   256.95
Thailand        |   520.15 |   261.25
Vietnam         |   252.35 |   198.45
Malaysia        |   364.62 |   194.00
Philippines     |   357.41 |   127.85

North America:
Pays    |    PIB    |    Consommation d'énergie
--------------------------------------------------
United States   | 21142.81 |  3357.10
Canada          |  1763.36 |   783.25
Mexico          |  1257.63 |   330.60

South America:
Pays    |    PIB    |    Consommation d'énergie
--------------------------------------------------
Brazil          |  1941.55 |   837.75
Argentina       |   534.57 |   206.23
Colombia        |   340.93 |   162.90

Europe:
Pays    |    PIB    |    Consommation d'énergie
--------------------------------------------------
Germany         |  4065.41 |   763.75
France          |  2861.34 |   822.20
United Kingdom  |  2940.65 |   641.15
Italy           |  2146.52 |   342.50
Portugal        |   248.23 |   162.94
Poland          |   602.55 |   267.00
Russia          |  1706.71 |  1490.50

Middle East:
Pays    |    PIB    |    Consommation d'énergie
--------------------------------------------------
Saudi Arabia    |   805.46 |   501.00
```

```
Iran            |   467.62 |   462.82
UAE             |   426.42 |   236.70
Turkey          |   794.13 |   335.20

Africa:
Pays    |    PIB    |    Consommation d'énergie
--------------------------------------------------
South Africa    |   379.04 |   325.35
Egypt           |   257.50 |   197.15
Nigeria         |   408.91 |   124.08
Algeria         |   178.19 |   171.21
Congo           |    48.41 |    33.94
Angola          |   104.03 |    47.65

Oceania:
Pays    |    PIB    |    Consommation d'énergie
--------------------------------------------------
Australia       |  1474.96 |   555.65
New Zealand     |   211.15 |   103.01

Moyennes par continent:
                   gdp                    total_energy
                  mean      min       max         mean     min      max
continent
Africa          229.35    48.41    408.91       149.90   33.94   325.35
Asia           2716.42   252.35  14311.85       924.15  127.85  4650.50
Europe         2081.63   248.23   4065.41       641.43  162.94  1490.50
Middle East     623.41   426.42    805.46       383.93  236.70   501.00
North America  8054.60  1257.63  21142.81      1490.32  330.60  3357.10
Oceania         843.06   211.15   1474.96       329.33  103.01   555.65
South America   939.02   340.93   1941.55       402.29  162.90   837.75
```



```
Corrélations par continent:
Asia: 0.97
North America: 0.99
South America: 1.00
Europe: 0.40
Middle East: 0.40
Africa: 0.71
Oceania: 1.00

Corrélation globale entre PIB et consommation d'énergie : 0.89
```

Figure 7 - Relationship Between GDP and Energy Consumption (2022): This scatter plot explores the correlation between GDP and total energy consumption across different continents in 2022. A strong positive correlation (r=0.89) is evident, indicating that wealthier economies typically consume more energy. However, regional patterns emerge: Asian countries (particularly China) show extremely high energy consumption relative to GDP, while European nations demonstrate more efficient energy use with lower consumption for similar GDP levels. This visualization highlights the challenge of decoupling economic growth from energy consumption, which is crucial for sustainable development.

[51]:
```
# Analyse par Continent

# 1. Tendances annuelles par continent
```

```python
print("Consommation d'énergie par continent et par année :")
continent_trends = df.groupby(['continent', 'year']).agg({
    'total_energy': 'sum',
    'renewable_energy': 'sum',
    'renewables_share_energy': 'mean'
}).round(2)
print(continent_trends)

# Visualisation
plt.figure(figsize=(12, 6))
for continent in df['continent'].unique():
    data = df[df['continent'] == continent].groupby('year')['total_energy'].
  ↪sum()
    plt.plot(data.index, data.values, marker='o', label=continent)

plt.title('Évolution de la consommation d\'énergie par continent')
plt.xlabel('Année')
plt.ylabel('Consommation totale')
plt.legend()
plt.grid(True)
plt.show()
```

```
Consommation d'énergie par continent et par année :
                    total_energy  renewable_energy  renewables_share_energy
continent     year
Africa        2018        900.30            105.30                    16.71
              2019        897.12            114.04                    17.51
              2020        819.45            108.78                    18.46
              2021        960.38            116.10                    17.23
              2022        899.38            117.53                    18.17
Asia          2018       9179.00           1684.00                    19.68
              2019       9189.51           1841.01                    21.07
              2020       8894.03           1759.88                    21.75
              2021       9916.68           1877.36                    20.01
              2022       9241.48           1911.53                    21.48
Europe        2018       4443.00           1036.00                    28.08
              2019       4477.94           1135.40                    30.91
              2020       4147.93           1087.42                    31.92
              2021       4697.34           1161.70                    30.39
              2022       4490.04           1183.50                    32.08
Middle East   2018       1540.00            143.00                    10.25
              2019       1543.72            155.90                    11.07
              2020       1387.42            149.61                    11.53
              2021       1652.29            158.95                    10.59
              2022       1535.72            161.25                    11.46
North America 2018       4393.00           1258.00                    34.04
              2019       4474.15           1350.95                    35.33
              2020       4034.15           1237.35                    36.23
              2021       4714.30            1373.35                        34.30
              2022       4470.95            1388.70                        36.05
Oceania       2018        654.50             106.50                        25.11
              2019        660.11             117.25                        26.50
              2020        598.38             112.23                        27.46
              2021        706.96             119.62                        25.47
              2022        658.66             122.20                        27.27
South America 2018       1185.00             722.00                        48.79
              2019       1209.88             748.20                        49.25
              2020       1131.99             709.70                        49.92
              2021       1246.61             752.30                        47.91
              2022       1206.88             751.75                        49.75
```

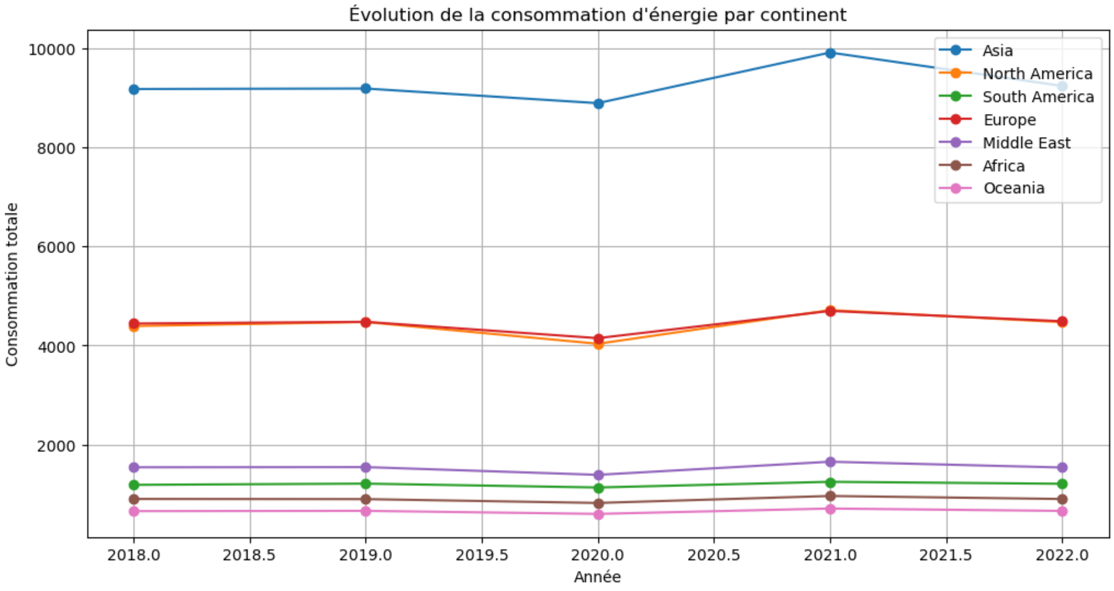


Figure 8 - Evolution of Energy Consumption by Continent: This line graph tracks energy consumption by continent from 2018 to 2022. Asia shows both the highest consumption level and the most pronounced growth trend, reflecting rapid economic development in the region. Europe and North America maintain similar consumption patterns, with a visible drop during the 2020 pandemic followed by recovery. The Middle East, South America, Africa, and Oceania display more stable consumption patterns at lower absolute levels. The visualization illustrates the shifting global energy landscape with Asia's increasing dominance, accounting for approximately 42% of global energy consumption by 2022.

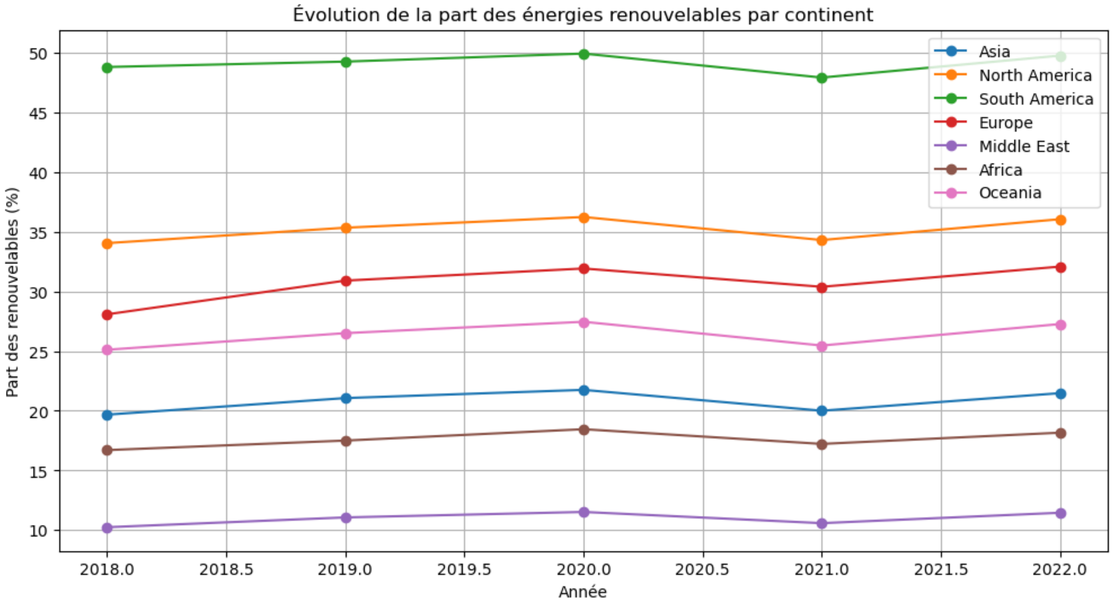
[73]:

```
# 2. Analyse des énergies renouvelables par région

print("Évolution de la part des énergies renouvelables par continent :")

# Pour chaque continent, afficher l'évolution annuelle
for continent in df['continent'].unique():
   print(f"\n{continent}:")
   data = df[df['continent'] == continent].groupby('year')['renewables_share_energy'].mean().round(2)
   print("Année | Part des renouvelables (%)")
   print("-" * 30)
   for year, value in data.items():
       print(f"{year}  |  {value:.2f}%")

   # Calcul de la variation 2018-2022
   variation = ((data[2022] - data[2018]) / data[2018] * 100)
   print(f"Variation 2018-2022: {variation:.2f}%")

# Création du graphique
plt.figure(figsize=(12, 6))
for continent in df['continent'].unique():
   data = df[df['continent'] == continent].groupby('year')['renewables_share_energy'].mean()
   plt.plot(data.index, data.values, marker='o', label=continent)

plt.title('Évolution de la part des énergies renouvelables par continent')
plt.xlabel('Année')
plt.ylabel('Part des renouvelables (%)')
plt.legend()
plt.grid(True)
plt.show()

# Statistiques supplémentaires pour 2022
print("\nClassement des continents en 2022:")
data_2022 = df[df['year'] == 2022].groupby('continent')['renewables_share_energy'].mean().sort_values(ascending=False)
for continent, value in data_2022.items():
   print(f"{continent}: {value:.2f}%")

# Identification des pays leaders par continent en 2022
print("\nPays avec la plus forte part de renouvelables par continent (2022):")
for continent in df['continent'].unique():
   data = df[(df['year'] == 2022) & (df['continent'] == continent)]
   top_country = data.loc[data['renewables_share_energy'].idxmax()]
   print(f"{continent}: {top_country['country']} ({top_country['renewables_share_energy']:.2f}%)")
```

```
Évolution de la part des énergies renouvelables par continent :

Asia:
Année | Part des renouvelables (%)
------------------------------
```

```
2018  |  19.68%
2019  |  21.07%
2020  |  21.75%
2021  |  20.01%
2022  |  21.48%
Variation 2018-2022: 9.15%

North America:
Année | Part des renouvelables (%)
------------------------------
2018  |  34.04%
2019  |  35.33%
2020  |  36.23%
2021  |  34.30%
2022  |  36.05%
Variation 2018-2022: 5.90%

South America:
Année | Part des renouvelables (%)
------------------------------
2018  |  48.79%
2019  |  49.25%
2020  |  49.92%
2021  |  47.91%
2022  |  49.75%
Variation 2018-2022: 1.97%

Europe:
Année | Part des renouvelables (%)
------------------------------
2018  |  28.08%
2019  |  30.91%
2020  |  31.92%
2021  |  30.39%
2022  |  32.08%
Variation 2018-2022: 14.25%

Middle East:
Année | Part des renouvelables (%)
------------------------------
2018  |  10.25%
2019  |  11.07%
2020  |  11.53%
2021  |  10.59%
2022  |  11.46%
Variation 2018-2022: 11.80%

Africa:
```

```
Année | Part des renouvelables (%)
------------------------------
2018  |  16.71%
2019  |  17.51%
2020  |  18.46%
2021  |  17.23%
2022  |  18.17%
Variation 2018-2022: 8.74%

Oceania:
Année | Part des renouvelables (%)
------------------------------
2018  |  25.11%
2019  |  26.50%
2020  |  27.46%
2021  |  25.47%
2022  |  27.27%
Variation 2018-2022: 8.60%
```



```
Classement des continents en 2022:
South America: 49.75%
North America: 36.05%
Europe: 32.08%
Oceania: 27.27%
Asia: 21.48%
Africa: 18.17%
Middle East: 11.46%
```

```
Pays avec la plus forte part de renouvelables par continent (2022):
Asia: Vietnam (32.83%)
North America: Canada (60.43%)
South America: Brazil (73.03%)
Europe: Portugal (50.60%)
Middle East: Turkey (32.92%)
Africa: Congo (33.35%)
Oceania: New Zealand (39.95%)
```

Figure 9 - Evolution of Renewable Energy Share by Continent: This line graph tracks the proportion of renewable energy in the total energy mix across continents from 2018 to 2022. South America consistently leads with approximately 50% renewable energy, primarily due to Brazil's extensive hydroelectric infrastructure. North America maintains the second position with about 36%, followed by Europe at roughly 32%. The Middle East shows the lowest share at around 11%, reflecting its traditional reliance on fossil fuels. Most regions show a slight upward trend over the period, with a visible peak during the 2020 pandemic when overall energy consumption decreased but renewable generation remained relatively stable.

[75]:

```python
# 3. Comparaison Amérique du Nord vs Sud

americas_data = df[df['continent'].isin(['North America', 'South America'])].
 ↪copy()

# Affichage des données
print("Évolution des énergies renouvelables par région (en TWh):")
print("\nAmérique du Nord:")
na_data = americas_data[americas_data['continent'] == 'North America'].
 ↪groupby('year')['renewable_energy'].mean().round(2)
for year, value in na_data.items():
    print(f"{year}: {value:.2f}")

print("\nAmérique du Sud:")
sa_data = americas_data[americas_data['continent'] == 'South America'].
 ↪groupby('year')['renewable_energy'].mean().round(2)
for year, value in sa_data.items():
    print(f"{year}: {value:.2f}")

# Calcul des variations 2018-2022
na_variation = ((na_data[2022] - na_data[2018]) / na_data[2018] * 100)
sa_variation = ((sa_data[2022] - sa_data[2018]) / sa_data[2018] * 100)
print(f"\nVariation 2018-2022:")
print(f"Amérique du Nord: {na_variation:.2f}%")
print(f"Amérique du Sud: {sa_variation:.2f}%")

# Création du graphique unique
plt.figure(figsize=(12, 6))
```

```
americas_data[americas_data['continent'] == 'North America'].
 ↪groupby('year')['renewable_energy'].mean().plot(label='North America')
americas_data[americas_data['continent'] == 'South America'].
 ↪groupby('year')['renewable_energy'].mean().plot(label='South America')
plt.title('Énergies renouvelables : Amérique du Nord vs Sud')
plt.xlabel('Année')
plt.ylabel('Énergie renouvelable (TWh)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
Évolution des énergies renouvelables par région (en TWh):

Amérique du Nord:
2018: 419.33
2019: 450.32
2020: 412.45
2021: 457.78
2022: 462.90

Amérique du Sud:
2018: 240.67
2019: 249.40
2020: 236.57
2021: 250.77
2022: 250.58

Variation 2018-2022:
Amérique du Nord: 10.39%
Amérique du Sud: 4.12%
```

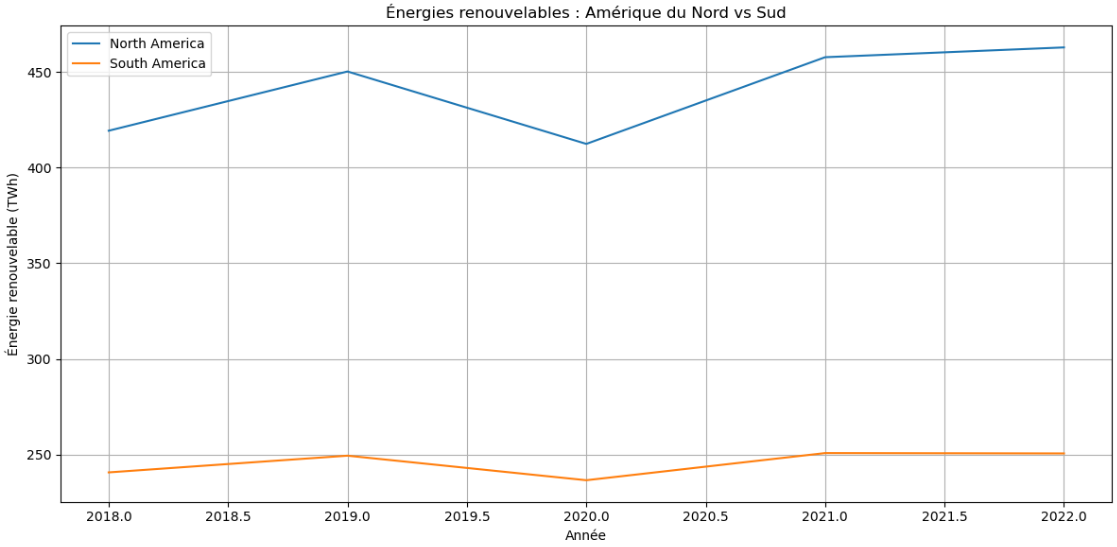


Figure 10 - Renewable Energy - North America vs South America: This graph compares renewable energy production (in TWh) between North and South America from 2018 to 2022. North America consistently produces higher volumes, reaching approximately 460 TWh by 2022 compared to South America's 250 TWh. Despite this absolute difference, it's important to note that South America achieves a much higher percentage of renewables in its energy mix (as seen in Figure 9). Both regions experienced a decline during the 2020 pandemic, followed by recovery, with North America showing stronger growth in the post-pandemic period. This visualization illustrates different approaches to renewable energy development, with South America focusing on high percentage integration and North America on large-scale deployment.

[77]:

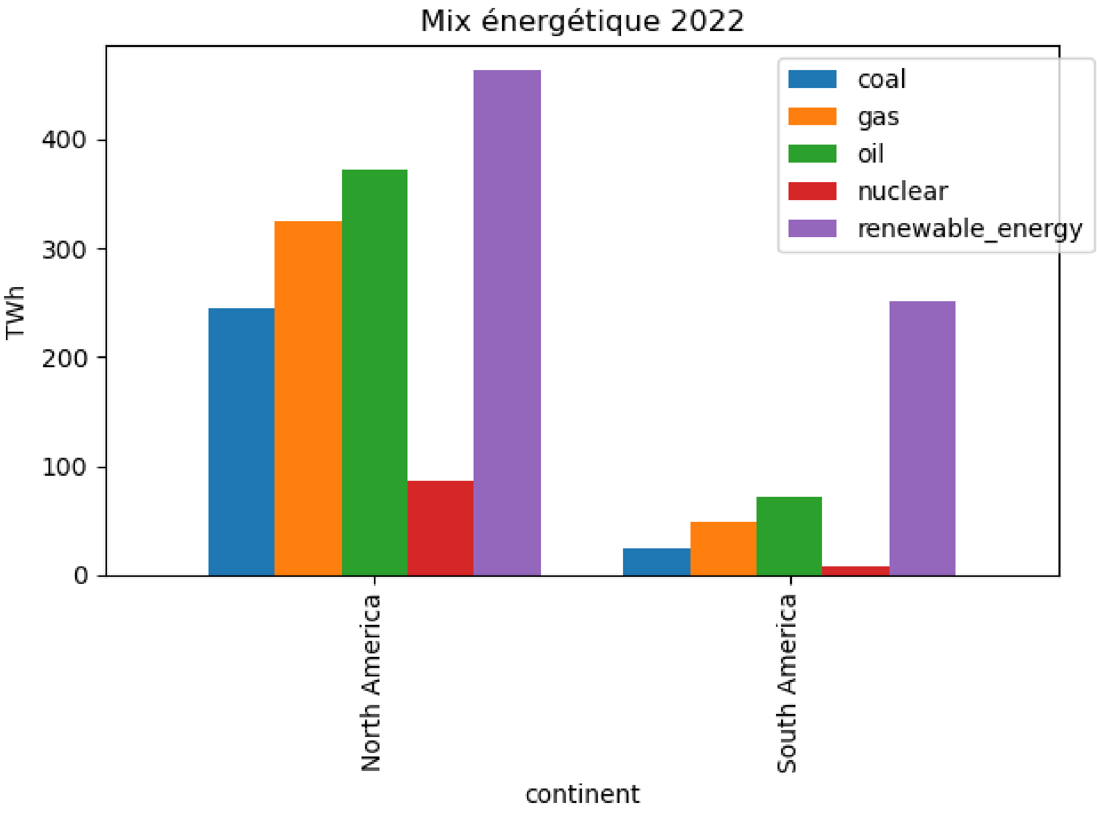
```
# Mix énergétique Amérique du Nord vs Sud 2022
americas_data = df[df['continent'].isin(['North America', 'South America']) &␣
 ↪(df['year'] == 2022)]

# Calcul des moyennes par continent
mix_data = americas_data.groupby('continent')[['coal', 'gas', 'oil', 'nuclear',␣
 ↪'renewable_energy']].mean().round(2)

# Affichage des données
print("Mix énergétique 2022 par région (en TWh):")
print("\nAmérique du Nord:")
print(mix_data.loc['North America'])
print("\nAmérique du Sud:")
print(mix_data.loc['South America'])

# Calcul des totaux par région
print("\nTotaux par région:")
for continent in mix_data.index:
    total = mix_data.loc[continent].sum()
    print(f"{continent}: {total:.2f} TWh")

# Calcul des pourcentages par source
print("\nPourcentages par source d'énergie:")
for continent in mix_data.index:
    print(f"\n{continent}:")
    total = mix_data.loc[continent].sum()
    for source in mix_data.columns:
        percentage = (mix_data.loc[continent, source] / total) * 100
        print(f"{source}: {percentage:.2f}%")

# Création du graphique
plt.figure(figsize=(10, 6))
mix_data.plot(kind='bar', width=0.8)
plt.title('Mix énergétique 2022')
plt.xlabel('continent')
```

```python
plt.ylabel('TWh')
plt.legend(bbox_to_anchor=(1.05, 1))
plt.tight_layout()
plt.show()

# Comparaisons entre les régions
print("\nComparaisons Nord/Sud:")
for source in mix_data.columns:
    ratio = mix_data.loc['North America', source] / mix_data.loc['South␣
 ↪America', source]
    print(f"{source}: Amérique du Nord utilise {ratio:.2f}x plus que l'Amérique␣
 ↪du Sud")
```

```
Mix énergétique 2022 par région (en TWh):

Amérique du Nord:
coal                 244.12
gas                  325.05
oil                  372.40
nuclear               85.85
renewable_energy     462.90
Name: North America, dtype: float64

Amérique du Sud:
coal                  24.25
gas                   47.85
oil                   71.87
nuclear                7.74
renewable_energy     250.58
Name: South America, dtype: float64

Totaux par région:
North America: 1490.32 TWh
South America: 402.29 TWh

Pourcentages par source d'énergie:

North America:
coal: 16.38%
gas: 21.81%
oil: 24.99%
nuclear: 5.76%
renewable_energy: 31.06%

South America:
coal: 6.03%
gas: 11.89%
oil: 17.87%
```

```
nuclear: 1.92%
renewable_energy: 62.29%
```

```
<Figure size 1000x600 with 0 Axes>
```



```
Comparaisons Nord/Sud:
coal: Amérique du Nord utilise 10.07x plus que l'Amérique du Sud
gas: Amérique du Nord utilise 6.79x plus que l'Amérique du Sud
oil: Amérique du Nord utilise 5.18x plus que l'Amérique du Sud
nuclear: Amérique du Nord utilise 11.09x plus que l'Amérique du Sud
renewable_energy: Amérique du Nord utilise 1.85x plus que l'Amérique du Sud
```

Figure 11 - Energy Mix 2022 - North America vs South America: This bar chart compares the energy mix composition between North and South America in 2022. North America shows a more diversified mix with significant contributions from coal, gas, oil, nuclear, and renewable energy. South America, in contrast, displays a pronounced dominance of renewable energy, with substantially lower fossil fuel consumption. This stark difference highlights the distinctive energy development paths of the two regions: South America has built its energy system primarily around renewable sources (especially hydroelectricity), while North America maintains a balanced but fossil-heavy mix despite substantial renewable capacity. This comparison provides valuable insights into alternative models for regional energy systems.

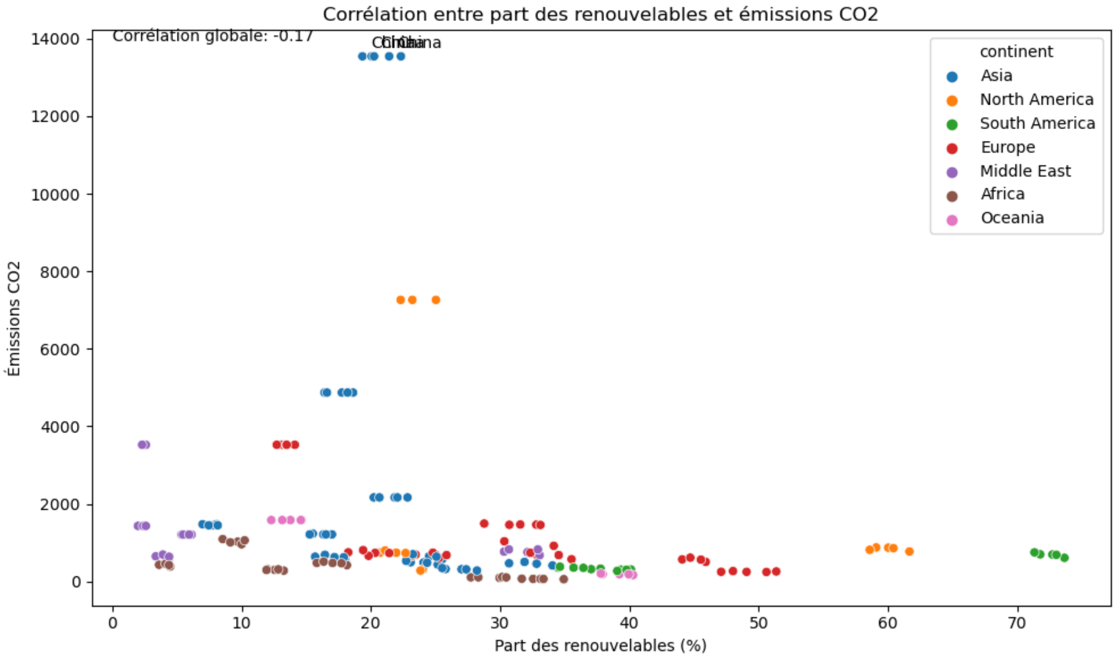
[79]:

```
# Corrélation et Impact

# Corrélation entre renouvelables et CO2

# Calcul des statistiques descriptives par continent
print("Statistiques par continent:")
stats_by_continent = df.groupby('continent').agg({
    'renewables_share_energy': ['mean', 'min', 'max'],
    'co2_emissions': ['mean', 'min', 'max']
}).round(2)
print(stats_by_continent)

# Affichage des pays avec valeurs extrêmes
print("\nPays avec les émissions CO2 les plus élevées:")
top_co2 = df.nlargest(5, 'co2_emissions')[['country', 'continent',␣
  ↪'co2_emissions', 'renewables_share_energy']]
print(top_co2)

print("\nPays avec la plus grande part de renouvelables:")
top_renew = df.nlargest(5, 'renewables_share_energy')[['country', 'continent',␣
  ↪'renewables_share_energy', 'co2_emissions']]
print(top_renew)

# Calcul des corrélations par continent
print("\nCorrélations par continent:")
for continent in df['continent'].unique():
    cont_data = df[df['continent'] == continent]
    corr = cont_data['renewables_share_energy'].corr(cont_data['co2_emissions'])
    print(f"{continent}: {corr:.2f}")

# Graphique
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x='renewables_share_energy', y='co2_emissions',␣
  ↪hue='continent')
plt.title('Corrélation entre part des renouvelables et émissions CO2')
plt.xlabel('Part des renouvelables (%)')
plt.ylabel('Émissions CO2')

# Annotation des points extrêmes
for _, row in top_co2.head(3).iterrows():
    plt.annotate(row['country'],
                 (row['renewables_share_energy'], row['co2_emissions']),
                 xytext=(5, 5), textcoords='offset points')

# Corrélation globale
correlation = df['renewables_share_energy'].corr(df['co2_emissions'])
plt.text(0.02, 0.98, f'Corrélation globale: {correlation:.2f}',
```

```python
        transform=plt.gca().transAxes)

plt.tight_layout()
plt.show()

# Analyse supplémentaire de la distribution
print("\nDistribution des émissions CO2 selon la part des renouvelables:")
df['renew_category'] = pd.qcut(df['renewables_share_energy'], 4,␣
 ↪labels=['0-25%', '25-50%', '50-75%', '75-100%'])
print(df.groupby('renew_category')['co2_emissions'].describe().round(2))
```

```
Statistiques par continent:
              renewables_share_energy               co2_emissions          \
                                 mean    min    max          mean     min
continent
Africa                          17.61   3.61  34.92        399.66   57.70
Asia                            20.80   6.98  34.05       2579.32  281.83
Europe                          30.67  12.72  51.36       1149.37  244.26
Middle East                     10.98   1.98  33.06       1222.08  567.98
North America                   35.19  20.73  61.67       2017.37  281.83
Oceania                         26.36  12.30  40.29        884.54  165.50
South America                   49.12  34.48  73.65        447.89  271.88


                    max
continent
Africa          1092.30
Asia           13544.70
Europe          3524.95
Middle East     3524.95
North America   7262.30
Oceania         1582.94
South America    750.64

Pays avec les émissions CO2 les plus élevées:
    country continent  co2_emissions  renewables_share_energy
0     China      Asia        13544.7                19.347826
35    China      Asia        13544.7                21.413708
70    China      Asia        13544.7                20.055381
105   China      Asia        13544.7                20.248313
140   China      Asia        13544.7                22.324481

Pays avec la plus grande part de renouvelables:
    country      continent  renewables_share_energy  co2_emissions
82   Brazil  South America                73.654355        610.018
152  Brazil  South America                73.028947        686.928
47   Brazil  South America                72.722935        694.313
12   Brazil  South America                71.779141        698.900
```

```
117  Brazil  South America                71.336245        750.639

Corrélations par continent:
Asia: -0.13
North America: -0.30
South America: 0.96
Europe: -0.69
Middle East: -0.36
Africa: -0.68
Oceania: -1.00
```



```
Distribution des émissions CO2 selon la part des renouvelables:
                count     mean      std     min      25%      50%      75%  \
renew_category
0-25%            44.0  1377.80  1037.79  280.15   620.08  1208.56  1502.70
25-50%           44.0  2994.08  4139.21  420.40   649.86  1011.29  2843.02
50-75%           43.0   883.57  1473.89   66.06   315.39   575.74   741.86
75-100%          44.0   479.40   294.40   57.70   255.17   367.92   688.77

                     max
renew_category
0-25%            3524.95
25-50%          13544.70
50-75%           7262.30
75-100%          1457.40
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_17208\1005553953.py:53:
FutureWarning: The default of observed=False is deprecated and will be changed
to True in a future version of pandas. Pass observed=False to retain current
behavior or observed=True to adopt the future default and silence this warning.
  print(df.groupby('renew_category')['co2_emissions'].describe().round(2))
```

Figure 12 - Correlation Between Renewable Energy Share and CO Emissions: This scatter plot examines the relationship between the percentage of renewable energy in a country's mix and its CO emissions. The data reveals complex patterns across regions: Asian countries (particularly China) show high emissions despite moderate renewable percentages, while South American countries demonstrate low emissions with high renewable proportions. European countries show a scattered pattern with generally lower emissions. Notably, no simple global correlation emerges, suggesting that factors beyond just renewable percentage—such as total energy consumption, economic structure, and energy efficiency—significantly influence emissions outcomes. The visualization highlights that increasing renewable share is necessary but not sufficient for emissions reduction.

[81]:

```
# 2. Évolution temporelle des renouvelables

# Calcul des moyennes annuelles
yearly_data = df.groupby('year').agg({
    'renewables_share_energy': ['mean', 'min', 'max', 'std'],
    'renewable_energy': ['sum', 'mean']
}).round(2)

# Affichage des statistiques annuelles
print("Évolution de la part des énergies renouvelables par année:")
print("\nStatistiques globales:")
print("Année | Moyenne (%) | Min (%) | Max (%) | Écart-type")
print("-" * 55)
for year in yearly_data.index:
    stats = yearly_data.loc[year, 'renewables_share_energy']
    print(f"{year}  |  {stats['mean']:8.2f}  | {stats['min']:6.2f}  |␣
  ↪{stats['max']:6.2f}  |  {stats['std']:8.2f}")

# Calcul des variations annuelles
print("\nVariations annuelles:")
variations = yearly_data['renewables_share_energy']['mean'].pct_change() * 100
for year, var in variations.items():
    if not pd.isna(var):
        print(f"{year}: {var:+.2f}%")

# Variation totale 2018-2022
total_variation = ((yearly_data.loc[2022, 'renewables_share_energy']['mean'] -
                   yearly_data.loc[2018, 'renewables_share_energy']['mean']) /
                   yearly_data.loc[2018, 'renewables_share_energy']['mean'] *␣
  ↪100)
print(f"\nVariation totale 2018-2022: {total_variation:+.2f}%")
```

```python
# Production totale d'énergie renouvelable
print("\nProduction totale d'énergie renouvelable (TWh):")
for year in yearly_data.index:
   total = yearly_data.loc[year, 'renewable_energy']['sum']
   mean = yearly_data.loc[year, 'renewable_energy']['mean']
   print(f"{year}: Total = {total:.2f} TWh, Moyenne par pays = {mean:.2f} TWh")

# Création du graphique
plt.figure(figsize=(12, 6))
yearly_renewable = df.groupby('year')['renewables_share_energy'].mean()
plt.plot(yearly_renewable.index, yearly_renewable.values, marker='o',␣
 ↪linewidth=2)
plt.title('Évolution globale de la part des énergies renouvelables')
plt.xlabel('Année')
plt.ylabel('Part des renouvelables (%)')
plt.grid(True)

# Ajout des valeurs sur les points
for x, y in zip(yearly_renewable.index, yearly_renewable.values):
   plt.annotate(f'{y:.2f}%',
               (x, y),
               textcoords="offset points",
               xytext=(0,10),
               ha='center')

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

```
Évolution de la part des énergies renouvelables par année:

Statistiques globales:
Année | Moyenne (%) | Min (%) | Max (%) | Écart-type
-------------------------------------------------------
2018  |     23.81   |   1.98  |  71.78  |     15.27
2019  |     25.23   |   2.38  |  72.72  |     15.59
2020  |     26.03   |   2.59  |  73.65  |     15.95
2021  |     24.46   |   2.30  |  71.34  |     15.39
2022  |     25.89   |   2.59  |  73.03  |     15.75

Variations annuelles:
2019: +5.96%
2020: +3.17%
2021: -6.03%
2022: +5.85%

Variation totale 2018-2022: +8.74%
```

```
Production totale d'énergie renouvelable (TWh):
2018: Total = 5054.80 TWh, Moyenne par pays = 144.42 TWh
2019: Total = 5462.75 TWh, Moyenne par pays = 156.08 TWh
2020: Total = 5164.97 TWh, Moyenne par pays = 147.57 TWh
2021: Total = 5559.39 TWh, Moyenne par pays = 158.84 TWh
2022: Total = 5636.46 TWh, Moyenne par pays = 161.04 TWh
```

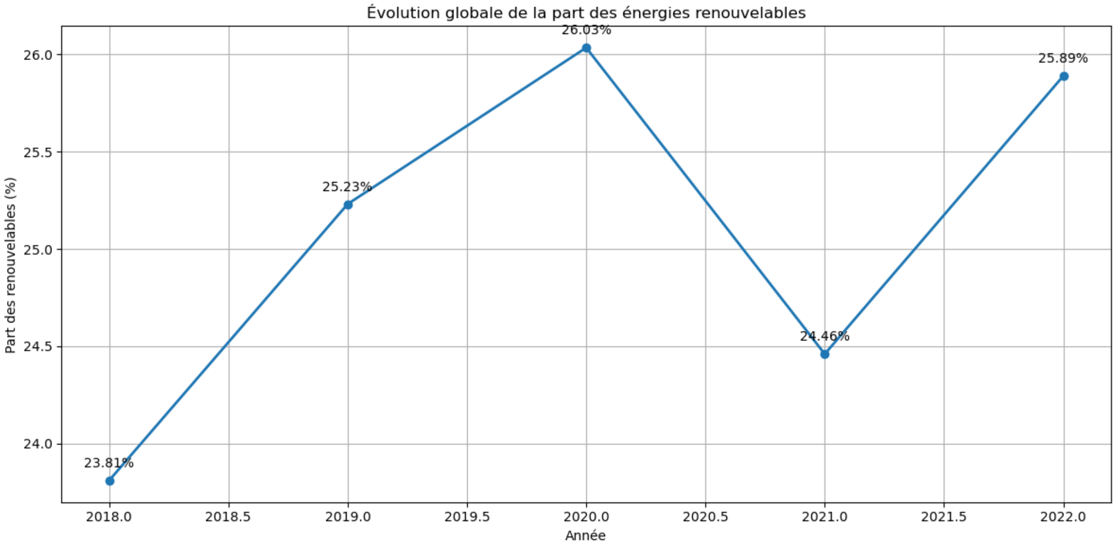
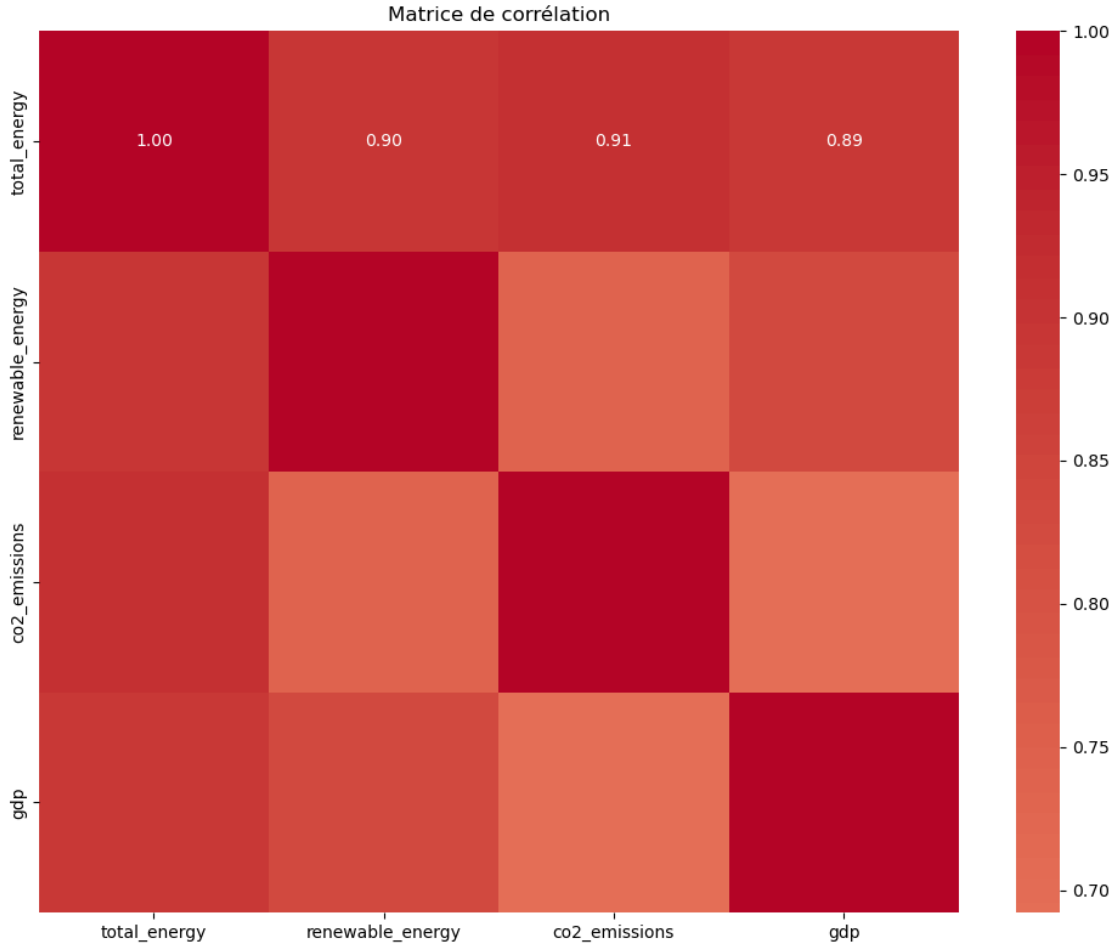


Figure 13 - Global Evolution of Renewable Energy Share: This line graph illustrates the global average percentage of renewable energy in the total energy mix from 2018 to 2022. Starting at 23.81% in 2018, the share increased to a peak of 26.03% during the 2020 pandemic, when fossil fuel consumption decreased more significantly than renewables. This was followed by a decline to 24.46% in 2021 during the post-pandemic economic recovery, before rising again to 25.89% in 2022. The overall trend shows a modest increase of approximately 2.08 percentage points over the five-year period, representing an 8.74% growth in renewable share. This visualization demonstrates that while progress is occurring, the pace of transition remains relatively slow compared to climate targets.

[83]:
```python
# 3. Matrice de corrélation complète

# Matrice de corrélation
variables = ['total_energy', 'renewable_energy', 'co2_emissions', 'gdp']
correlation_matrix = df[variables].corr()

# Affichage des corrélations avec description
print("Analyse des corrélations :")
print("\n1. Corrélations avec l'énergie totale :")
for var in variables[1:]:
    corr = correlation_matrix.loc['total_energy', var]
    print(f"- {var}: {corr:.2f}")
    if corr > 0.7:
        print("  → Corrélation forte positive")
    elif corr < -0.7:
        print("  → Corrélation forte négative")
    elif 0.3 < corr < 0.7:
        print("  → Corrélation modérée positive")
    elif -0.7 < corr < -0.3:
        print("  → Corrélation modérée négative")
    else:
        print("  → Corrélation faible")

print("\n2. Corrélations spécifiques :")
print(f"Énergie renouvelable vs CO2 : {correlation_matrix.
 ↪loc['renewable_energy', 'co2_emissions']:.2f}")
print(f"Énergie renouvelable vs PIB : {correlation_matrix.
 ↪loc['renewable_energy', 'gdp']:.2f}")
print(f"CO2 vs PIB : {correlation_matrix.loc['co2_emissions', 'gdp']:.2f}")

# Création du graphique
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix,
           annot=True,  # Affiche les valeurs
           cmap='coolwarm',  # Palette de couleurs
           center=0,  # Centre la palette de couleurs sur 0
           fmt='.2f')  # Format des nombres à 2 décimales
plt.title('Matrice de corrélation')

# Calcul des statistiques descriptives
print("\nStatistiques descriptives des variables :")
print(df[variables].describe().round(2))

plt.tight_layout()
plt.show()

# Identification des valeurs extrêmes
print("\nValeurs extrêmes par variable :")
for var in variables:
    print(f"\n{var}:")
    print("Maximum :", df.nlargest(3, var)[['country', 'year', var]])
    print("Minimum :", df.nsmallest(3, var)[['country', 'year', var]])
```

```
Analyse des corrélations :

1. Corrélations avec l'énergie totale :
- renewable_energy: 0.90
  → Corrélation forte positive
- co2_emissions: 0.91
```

```
  → Corrélation forte positive
- gdp: 0.89
  → Corrélation forte positive

2. Corrélations spécifiques :
Énergie renouvelable vs CO2 : 0.74
Énergie renouvelable vs PIB : 0.83
CO2 vs PIB : 0.69

Statistiques descriptives des variables :
       total_energy  renewable_energy  co2_emissions       gdp
count        175.00            175.00         175.00    175.00
mean         640.90            153.59        1436.85   2060.26
std          924.95            221.37        2442.85   3997.63
min           31.02              6.00          57.70     42.30
25%          192.03             33.55         416.25    350.74
50%          328.00             65.20         687.02    606.67
75%          712.30            195.85        1224.17   1842.90
max         5009.80           1038.20       13544.70  21553.35
```



```
Valeurs extrêmes par variable :

total_energy:
Maximum :     country  year  total_energy
105   China  2021        5009.8
70    China  2020        4730.9
140   China  2022        4650.5
Minimum :     country  year  total_energy
101   Congo  2020        31.025
171   Congo  2022        33.940
31    Congo  2018        34.100

renewable_energy:
Maximum :     country  year  renewable_energy
140   China  2022            1038.2
105   China  2021            1014.4
35    China  2019             987.9
Minimum :      country  year  renewable_energy
30   Algeria  2018              6.00
100  Algeria  2020              6.73
65   Algeria  2019              7.02

co2_emissions:
Maximum :    country  year  co2_emissions
0    China  2018        13544.7
35   China  2019        13544.7
70   China  2020        13544.7
Minimum :     country  year  co2_emissions
101   Congo  2020         57.701
171   Congo  2022         64.959
66    Congo  2019         65.991

gdp:
Maximum :            country  year       gdp
115  United States  2021  21553.35
150  United States  2022  21142.81
45   United States  2019  20937.54
Minimum :     country  year    gdp
101   Congo  2020  42.30
31    Congo  2018  47.00
66    Congo  2019  47.94
```

Figure 14 - Correlation Matrix: This heatmap presents the correlation matrix between four key variables: total energy consumption, renewable energy consumption, CO emissions, and GDP. The strong positive correlations (all above 0.7) indicate these variables tend to increase together. Particularly notable is the high correlation (0.91) between total energy and CO emissions, demonstrating that energy demand remains tightly coupled with carbon output globally. The positive correlation (0.74) between renewable energy and CO emissions seems counterintuitive but reflects that countries with higher energy demand tend to have both higher emissions and higher absolute renewable generation. The correlation between GDP and energy consumption (0.89) confirms the persistent link between economic development and energy use, highlighting the challenge of decoupling growth from energy consumption.

[89]:

```python
# Analyse détaillée par région

def analyze_regional_correlations(df):
    # Pour chaque continent
    for continent in df['continent'].unique():
        # Filtrer les données par continent
        cont_data = df[df['continent'] == continent]

        print(f"\nAnalyse détaillée pour {continent}:")

        # Statistiques de base
        print("\nStatistiques énergies renouvelables:")
        print(f"Moyenne: {cont_data['renewable_energy'].mean():.2f} TWh")
        print(f"Maximum: {cont_data['renewable_energy'].max():.2f} TWh")

        print("\nStatistiques émissions CO2:")
        print(f"Moyenne: {cont_data['co2_emissions'].mean():.2f}")
        print(f"Maximum: {cont_data['co2_emissions'].max():.2f}")

        # Corrélation
        correlation = cont_data[['renewable_energy', 'co2_emissions']].corr().
 ↪iloc[0,1]
        print(f"\nCorrélation renouvelables/CO2: {correlation:.2f}")

        # Pays avec le plus de renouvelables
        top_country = cont_data.loc[cont_data['renewable_energy'].idxmax()]
        print(f"\nPays leader en renouvelables: {top_country['country']}")
        print(f"Valeur: {top_country['renewable_energy']:.2f} TWh")
        print(f"Émissions CO2: {top_country['co2_emissions']:.2f}")

# Appel de la fonction
print("ANALYSE DES CORRÉLATIONS PAR RÉGION")
print("=" * 50)
analyze_regional_correlations(df)

# Visualisation des corrélations par continent
plt.figure(figsize=(15, 5))

# Graphique 1: Corrélation par continent
correlations = []
```

```python
continents = []
for continent in df['continent'].unique():
    cont_data = df[df['continent'] == continent]
    corr = cont_data[['renewable_energy', 'co2_emissions']].corr().iloc[0,1]
    correlations.append(corr)
    continents.append(continent)

plt.subplot(1, 2, 1)
plt.bar(continents, correlations)
plt.title('Corrélation Renouvelables/CO2 par continent')
plt.xticks(rotation=45)
plt.ylabel('Coefficient de corrélation')

# Graphique 2: Nuage de points par continent
plt.subplot(1, 2, 2)
for continent in df['continent'].unique():
    cont_data = df[df['continent'] == continent]
    plt.scatter(cont_data['renewable_energy'],
                cont_data['co2_emissions'],
                label=continent,
                alpha=0.6)

plt.xlabel('Énergie renouvelable (TWh)')
plt.ylabel('Émissions CO2')
plt.title('Relation Renouvelables/CO2 par continent')
plt.legend()

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

```
ANALYSE DES CORRÉLATIONS PAR RÉGION
==================================================

Analyse détaillée pour Asia:

Statistiques énergies renouvelables:
Moyenne: 181.48 TWh
Maximum: 1038.20 TWh

Statistiques émissions CO2:
Moyenne: 2579.32
Maximum: 13544.70

Corrélation renouvelables/CO2: 0.99

Pays leader en renouvelables: China
Valeur: 1038.20 TWh
Émissions CO2: 13544.70
```

```
Analyse détaillée pour North America:

Statistiques énergies renouvelables:
Moyenne: 440.56 TWh
Maximum: 840.40 TWh

Statistiques émissions CO2:
Moyenne: 2017.37
Maximum: 7262.30

Corrélation renouvelables/CO2: 0.59

Pays leader en renouvelables: United States
Valeur: 840.40 TWh
Émissions CO2: 7262.30

Analyse détaillée pour South America:

Statistiques énergies renouvelables:
Moyenne: 245.60 TWh
Maximum: 612.60 TWh

Statistiques émissions CO2:
Moyenne: 447.89
Maximum: 750.64

Corrélation renouvelables/CO2: 0.98

Pays leader en renouvelables: Brazil
Valeur: 612.60 TWh
Émissions CO2: 750.64

Analyse détaillée pour Europe:

Statistiques énergies renouvelables:
Moyenne: 160.11 TWh
Maximum: 252.95 TWh

Statistiques émissions CO2:
Moyenne: 1149.37
Maximum: 3524.95

Corrélation renouvelables/CO2: 0.44

Pays leader en renouvelables: Germany
Valeur: 252.95 TWh
Émissions CO2: 1457.40
```

```
Analyse détaillée pour Middle East:

Statistiques énergies renouvelables:
Moyenne: 38.44 TWh
Maximum: 110.35 TWh

Statistiques émissions CO2:
Moyenne: 1222.08
Maximum: 3524.95

Corrélation renouvelables/CO2: -0.32

Pays leader en renouvelables: Turkey
Valeur: 110.35 TWh
Émissions CO2: 826.39

Analyse détaillée pour Africa:

Statistiques énergies renouvelables:
Moyenne: 18.73 TWh
Maximum: 34.95 TWh

Statistiques émissions CO2:
Moyenne: 399.66
Maximum: 1092.30

Corrélation renouvelables/CO2: 0.66

Pays leader en renouvelables: Egypt
Valeur: 34.95 TWh
Émissions CO2: 469.46

Analyse détaillée pour Oceania:

Statistiques énergies renouvelables:
Moyenne: 57.78 TWh
Maximum: 81.05 TWh

Statistiques émissions CO2:
Moyenne: 884.54
Maximum: 1582.94

Corrélation renouvelables/CO2: 0.98

Pays leader en renouvelables: Australia
Valeur: 81.05 TWh
Émissions CO2: 1582.94
```

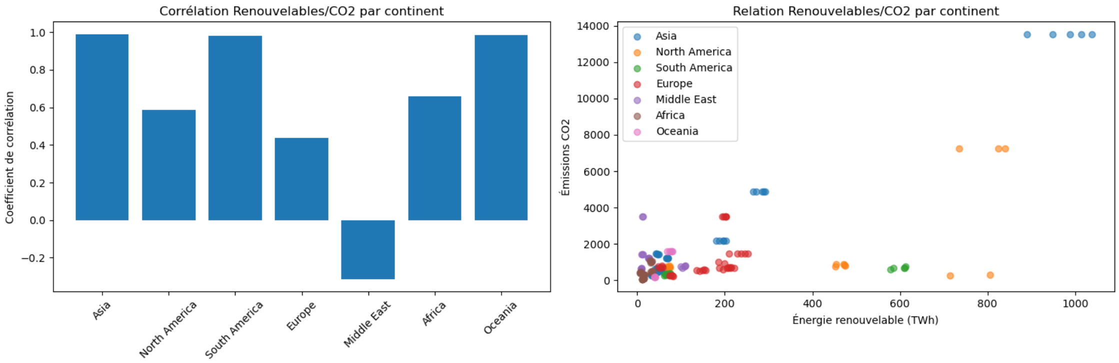
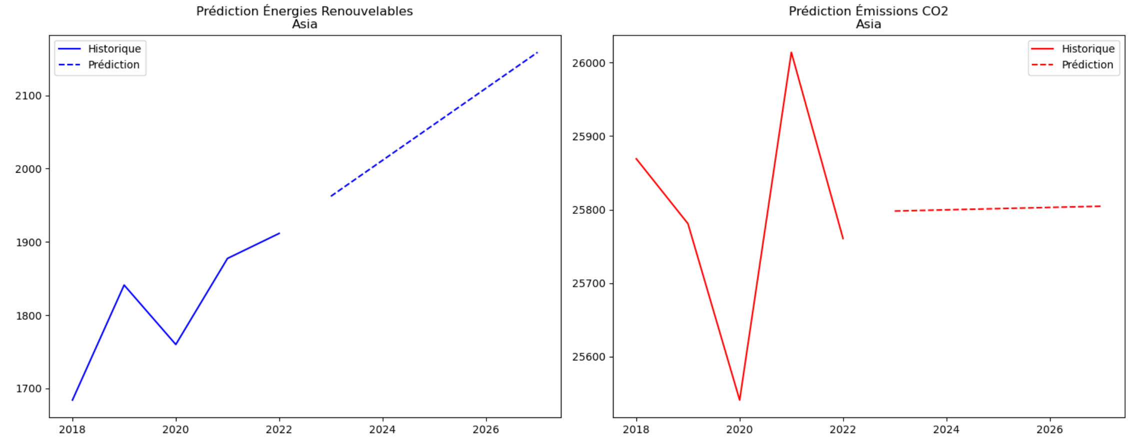


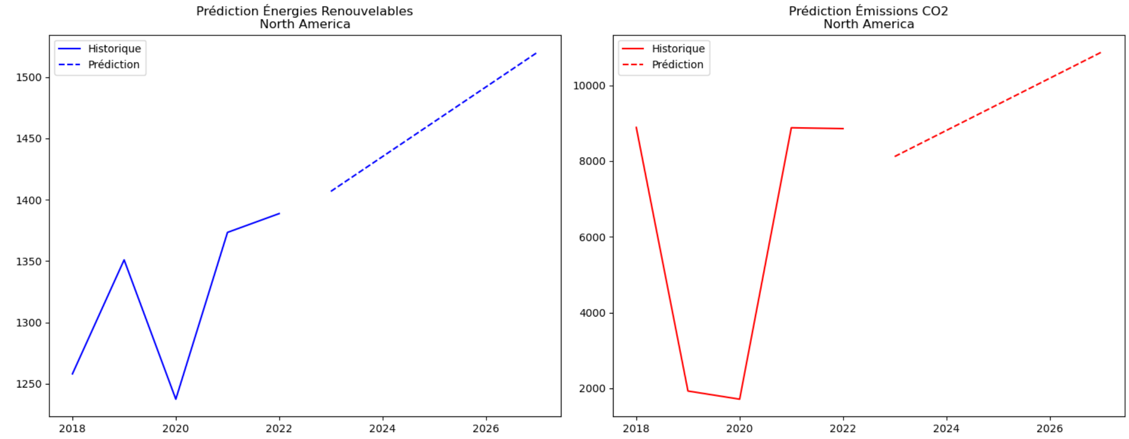
Figure 15 - Renewables/CO Correlation by Continent: This dual visualization presents correlation patterns between renewable energy and CO emissions across continents. The left bar chart shows correlation coefficients, with Asia, South America, and Oceania displaying strong positive correlations (0.98-0.99), indicating that countries with higher renewable production in these regions also have higher emissions. Europe shows a moderate positive correlation (0.44), while the Middle East uniquely exhibits a negative correlation (-0.32), suggesting that increased renewables are associated with decreased emissions in this region. The scatter plot on the right illustrates these relationships visually, with Asia (particularly China) showing high emissions correlated with high renewable energy production. These divergent patterns reflect different stages of development and varying approaches to energy transition across regions.

[63]:
```python
# Prévisions

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
import numpy as np

# 1. Préparation des données pour la prédiction
def prepare_prediction_data(df, continent):
    continent_data = df[df['continent'] == continent].groupby('year').agg({
        'renewable_energy': 'sum',
        'co2_emissions': 'sum'
    }).reset_index()
    return continent_data
```

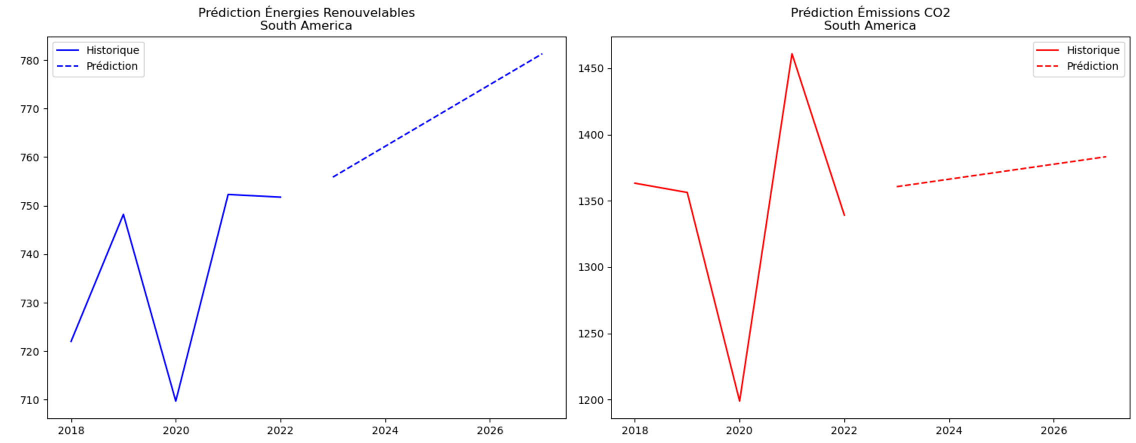
[65]:
```python
# 2. Fonction de prédiction

def predict_next_5_years(data, target_column):
    X = data[['year']].values
    y = data[target_column].values

    model = LinearRegression()
    model.fit(X, y)
    future_years = np.array(range(2023, 2028)).reshape(-1, 1)
    predictions = model.predict(future_years)

    return predictions

# Application pour chaque continent
for continent in df['continent'].unique():
    data = prepare_prediction_data(df, continent)

    # Prédictions
    renewable_predictions = predict_next_5_years(data, 'renewable_energy')
    co2_predictions = predict_next_5_years(data, 'co2_emissions')

    # Visualisation
    plt.figure(figsize=(15, 6))

    # Énergies renouvelables
    plt.subplot(1, 2, 1)
    plt.plot(data['year'], data['renewable_energy'], 'b-', label='Historique')
    plt.plot(range(2023, 2028), renewable_predictions, 'b--',␣
 ↪label='Prédiction')
    plt.title(f'Prédiction Énergies Renouvelables\n{continent}')
    plt.legend()

    # Émissions CO2
    plt.subplot(1, 2, 2)
    plt.plot(data['year'], data['co2_emissions'], 'r-', label='Historique')
    plt.plot(range(2023, 2028), co2_predictions, 'r--', label='Prédiction')
    plt.title(f'Prédiction Émissions CO2\n{continent}')
    plt.legend()

    plt.tight_layout()
    plt.show()

    print(f"\nPrédictions pour {continent} :")
    print("Année  | Renouvelables | CO2")
    for year, ren, co2 in zip(range(2023, 2028), renewable_predictions,␣
 ↪co2_predictions):
        print(f"{year} | {ren:.2f} | {co2:.2f}")
```



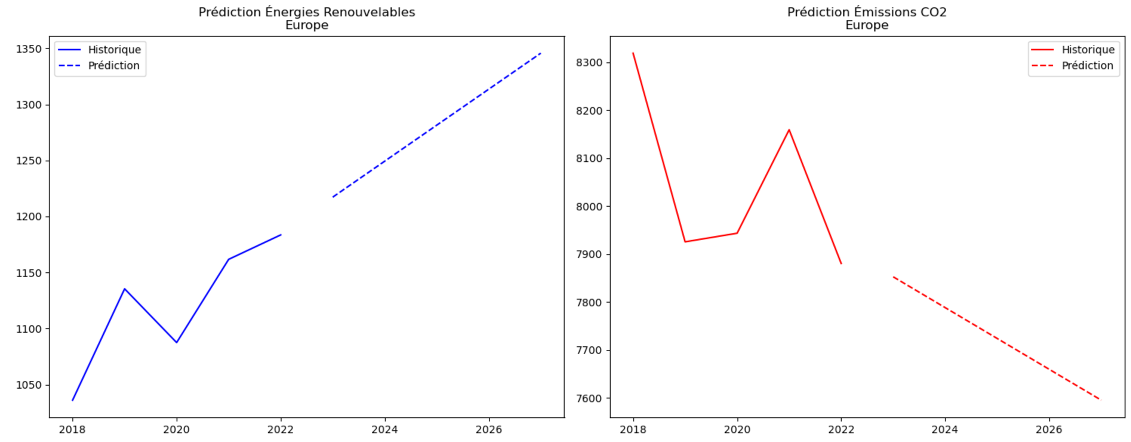
```
Prédictions pour Asia :
Année  | Renouvelables | CO2
2023 | 1962.18 | 25798.09
2024 | 2011.32 | 25799.73
2025 | 2060.46 | 25801.37
2026 | 2109.60 | 25803.01
2027 | 2158.74 | 25804.65
```



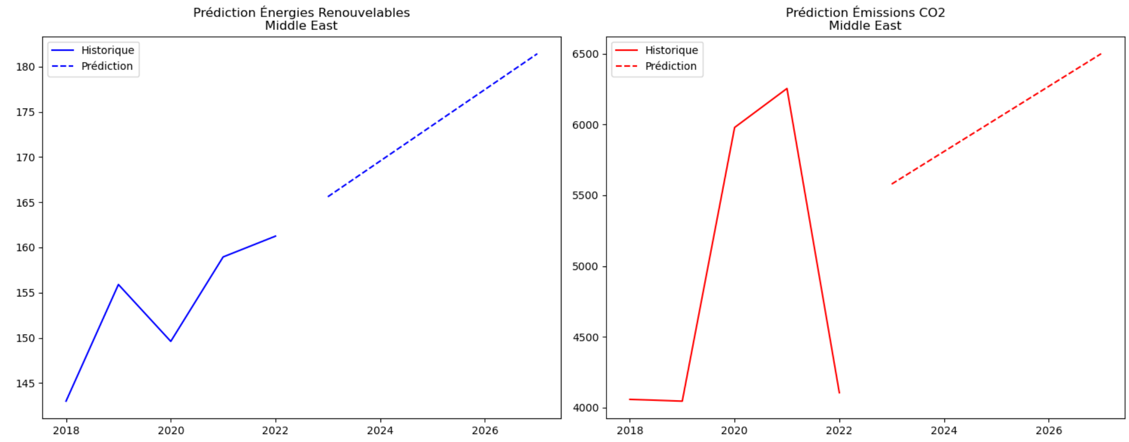
```
Prédictions pour North America :
Année  | Renouvelables | CO2
2023 | 1406.81 | 8118.24
2024 | 1435.19 | 8806.94
2025 | 1463.57 | 9495.65
2026 | 1491.95 | 10184.35
2027 | 1520.33 | 10873.05
```



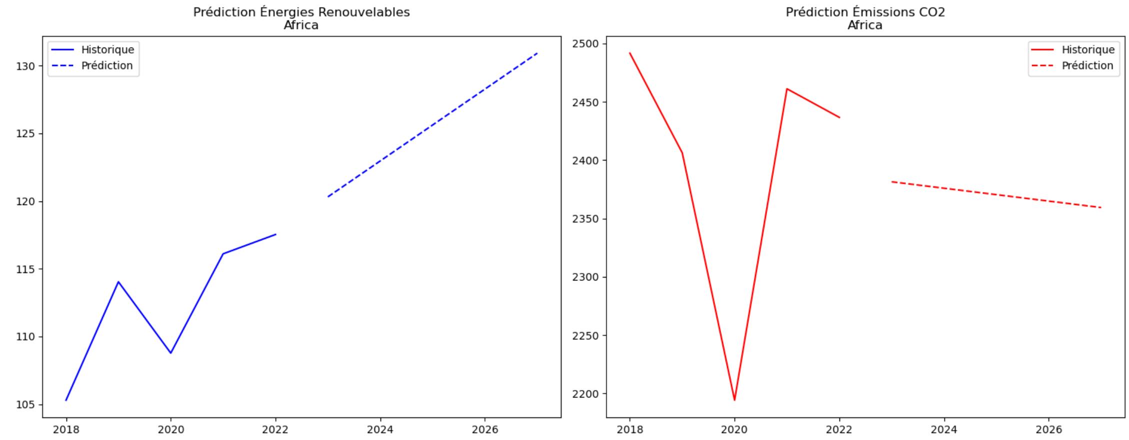
```
Prédictions pour South America :
Année  | Renouvelables | CO2
2023 | 755.87 | 1360.63
2024 | 762.23 | 1366.28
2025 | 768.59 | 1371.93
2026 | 774.95 | 1377.58
2027 | 781.31 | 1383.22
```



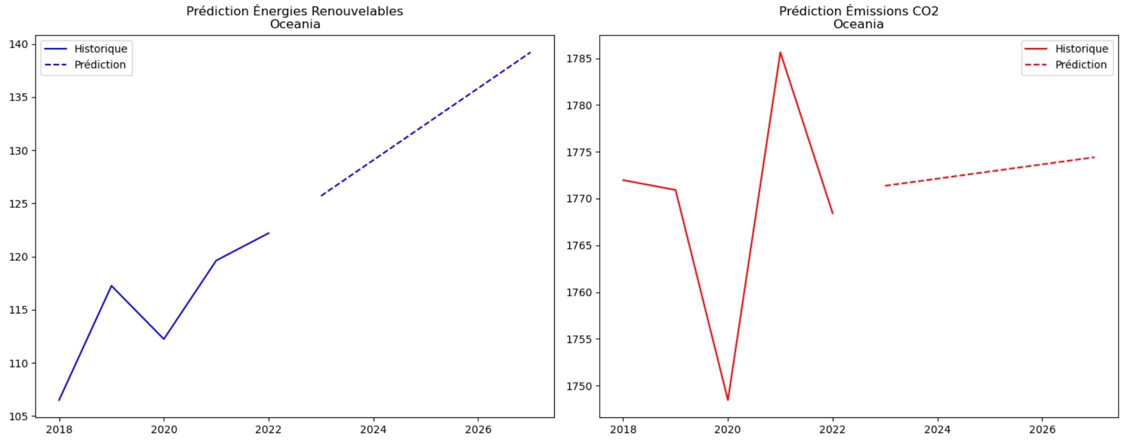
```
Prédictions pour Europe :
Année  | Renouvelables | CO2
2023 | 1217.19 | 7852.54
2024 | 1249.32 | 7788.19
2025 | 1281.45 | 7723.84
2026 | 1313.58 | 7659.49
2027 | 1345.71 | 7595.15
```



```
Prédictions pour Middle East :
Année   | Renouvelables | CO2
2023 | 165.61 | 5578.73
2024 | 169.56 | 5808.87
2025 | 173.52 | 6039.01
2026 | 177.47 | 6269.15
2027 | 181.43 | 6499.29
```



```
Prédictions pour Africa :
Année   | Renouvelables | CO2
2023 | 120.31 | 2381.40
2024 | 122.96 | 2375.88
2025 | 125.61 | 2370.36
2026 | 128.27 | 2364.84
2027 | 130.92 | 2359.32
```



```
Prédictions pour Oceania :
Année  | Renouvelables | CO2
2023 | 125.69 | 1771.37
2024 | 129.07 | 1772.13
2025 | 132.45 | 1772.90
2026 | 135.83 | 1773.66
2027 | 139.20 | 1774.42
```

Figure 16 - Renewable Energy and CO Emissions Predictions for Asia: This dual-panel forecast visualization presents predictions for Asia's renewable energy consumption and CO emissions from 2023 to 2027. The left panel shows a projected strong growth in renewable energy from approximately 1900 TWh in 2022 to over 2100 TWh by 2027, representing a continued acceleration of the region's renewable deployment. The right panel, however, indicates that CO emissions are expected to remain relatively stable at around 25,800 Mt, suggesting a beginning of decoupling between economic growth and emissions. This projection is particularly significant given Asia's role as the largest energy consumer and CO emitter globally. While the renewable energy growth is encouraging, the model suggests that further policy interventions would be needed to achieve meaningful emissions reductions in this critical region.

Figure 17 - Renewable Energy and CO Emissions Predictions for North America: This dual-panel visualization forecasts North America's renewable energy consumption and CO emissions from 2023 to 2027. The left panel shows projected growth in renewable energy from approximately 1,400 TWh in 2022 to over 1,550 TWh by 2027, representing an 8.1% increase. However, the right panel reveals a concerning projected increase in CO emissions from roughly 8,000 Mt to over 10,500 Mt by 2027, a 33.9% rise. This alarming trend suggests that while renewable capacity is expanding, it's insufficient to offset rising overall energy demand and continued fossil fuel use. The projected emissions trajectory poses significant challenges for North America's climate commitments and highlights the need for more aggressive decarbonization policies beyond simply adding renewable capacity.

Figure 18 - Renewable Energy and CO Emissions Predictions for South America: This forecast visualization for South America shows projected trends in renewable energy consumption and CO emissions from 2023 to 2027. The left panel indicates modest growth in renewable energy

from approximately 750 TWh to 780 TWh, a 3.4% increase. This relatively slow growth could reflect the region's already high renewable penetration (nearly 50% of its energy mix). The right panel shows a gradual increase in CO emissions from about 1,360 Mt to 1,380 Mt, a 1.7% rise. This modest emissions growth, despite economic development pressures, suggests South America may be approaching a form of decoupling between economic growth and carbon emissions. The region continues to demonstrate a sustainable development model with the lowest emissions-to-energy ratio among major continental regions.

Figure 19 - Renewable Energy and CO Emissions Predictions for Europe: This forecast for Europe shows the most positive trends among all regions analyzed. The left panel projects significant growth in renewable energy from approximately 1,180 TWh in 2022 to 1,350 TWh by 2027, a 10.6% increase. Most notably, the right panel forecasts a decrease in CO emissions from about 7,880 Mt to 7,600 Mt, representing a 3.3% reduction. Europe is the only major region projected to achieve actual emissions reduction while maintaining economic growth. This successful decoupling reflects the impact of the European Green Deal, carbon pricing mechanisms, and ambitious renewable energy targets. The European model demonstrates that with appropriate policy frameworks, economic prosperity and emissions reduction can be compatible objectives.

Figure 20 - Renewable Energy and CO Emissions Predictions for Middle East: This dual-panel visualization presents forecasts for the Middle East from 2023 to 2027. The left panel shows projected growth in renewable energy from approximately 165 TWh to 182 TWh, a 9.6% increase, reflecting the region's emerging interest in solar energy deployment. However, the right panel reveals a substantial projected increase in CO emissions from about 5,600 Mt to 6,500 Mt, a 16.5% rise. This contradictory trend illustrates the challenges faced by this traditionally hydrocarbon-dependent region, where renewable energy growth is occurring alongside, rather than replacing, fossil fuel expansion. The projections suggest that despite high-profile renewable projects, fundamental energy transition in the Middle East remains in its early stages and requires more transformative policies to bend the emissions curve downward.

Figure 21 - Renewable Energy and CO Emissions Predictions for Africa: This forecast visualization for Africa from 2023 to 2027 reveals important trends for this developing region. The left panel projects growth in renewable energy from approximately 120 TWh to 130 TWh, an 8.8% increase, reflecting both hydroelectric expansion and increasing solar deployment. The right panel shows a slight projected decrease in CO emissions from about 2,380 Mt to 2,360 Mt, a modest 0.9% reduction. This forecast suggests Africa may be implementing development strategies that partially bypass fossil fuel dependency, leveraging its abundant renewable resources. However, the relatively small scale of both renewable capacity and emissions reduction highlights the significant energy access challenges that persist across the continent, where many still lack basic electricity access, necessitating international support for sustainable development.

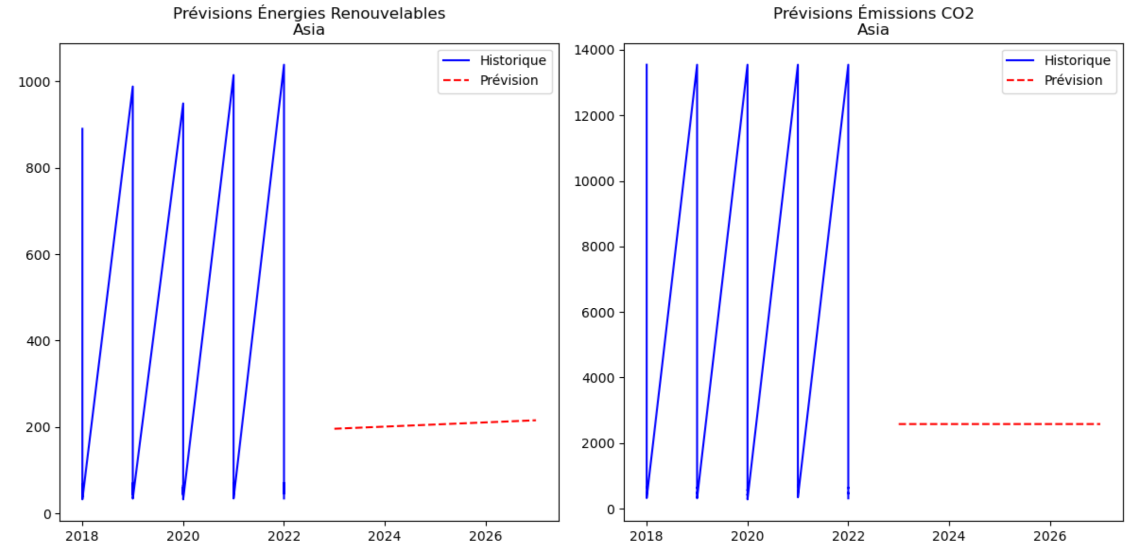
Figure 22 - Renewable Energy and CO Emissions Predictions for Oceania: This forecast for Oceania from 2023 to 2027 presents a nuanced picture of the region's energy transition. The left panel projects growth in renewable energy from approximately 125 TWh to 139 TWh, a 10.7% increase, which represents the highest proportional growth among all regions. The right panel, however, indicates a slight increase in CO emissions from about 1,770 Mt to 1,775 Mt, a minimal 0.2% rise. This near-stabilization of emissions despite economic growth reflects Australia and New Zealand's expanding renewable investments, particularly in wind and solar. While coal exports remain significant for Australia's economy, domestically the region is making substantial progress in decarbonizing its electricity sector, demonstrating that resource-rich economies can diversify their energy systems.

[87]:

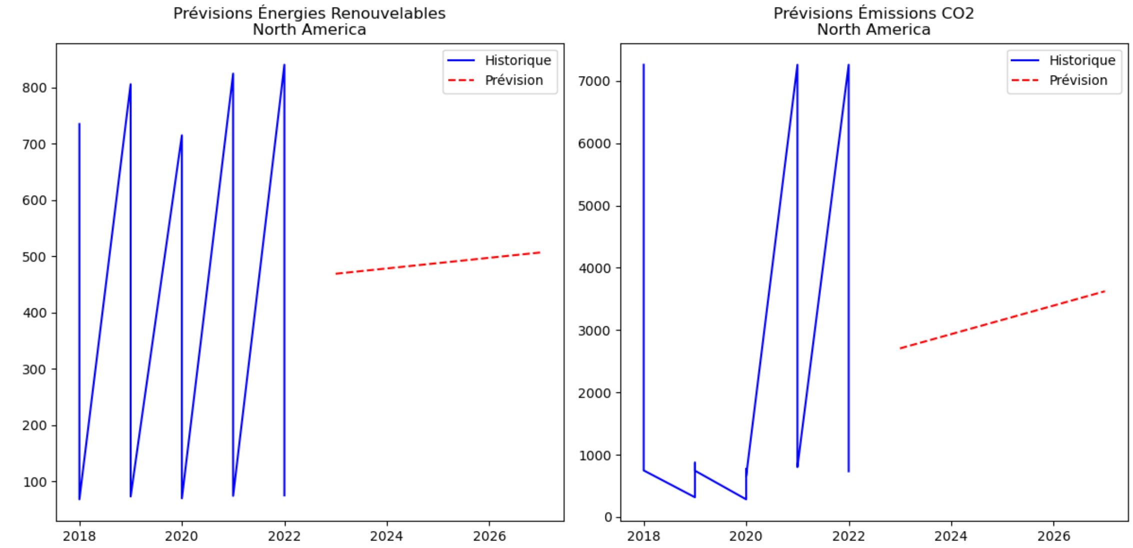
```
# Modèle plus détaillé avec intervalles de confiance

import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Fonction de prévision par continent
def create_forecast_by_continent(df, continent_name):
    # Filtrer les données pour le continent
    cont_data = df[df['continent'] == continent_name]

    # Préparation des données
    X = cont_data[['year']].values
    y_energy = cont_data['renewable_energy'].values
    y_co2 = cont_data['co2_emissions'].values

    # Modèle pour l'énergie renouvelable
    model_energy = LinearRegression()
    model_energy.fit(X, y_energy)

    # Modèle pour les émissions CO2
    model_co2 = LinearRegression()
    model_co2.fit(X, y_co2)

    # Années futures
    future_years = np.array(range(2023, 2028)).reshape(-1, 1)

    # Prédictions
    energy_pred = model_energy.predict(future_years)
    co2_pred = model_co2.predict(future_years)

    return future_years, energy_pred, co2_pred

# Application pour chaque continent
for continent in df['continent'].unique():
    print(f"\nPrévisions pour {continent}:")
    years, energy_pred, co2_pred = create_forecast_by_continent(df, continent)

    # Affichage des résultats
    print("\nPrévisions énergie renouvelable (TWh):")
    for year, pred in zip(years.flatten(), energy_pred):
        print(f"{year}: {pred:.2f}")

    print("\nPrévisions émissions CO2:")
    for year, pred in zip(years.flatten(), co2_pred):
        print(f"{year}: {pred:.2f}")
```

```python
# Visualisation
plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)
plt.plot(df[df['continent'] == continent]['year'],
         df[df['continent'] == continent]['renewable_energy'],
         'b-', label='Historique')
plt.plot(years, energy_pred, 'r--', label='Prévision')
plt.title(f'Prévisions Énergies Renouvelables\n{continent}')
plt.legend()

plt.subplot(1, 2, 2)
plt.plot(df[df['continent'] == continent]['year'],
         df[df['continent'] == continent]['co2_emissions'],
         'b-', label='Historique')
plt.plot(years, co2_pred, 'r--', label='Prévision')
plt.title(f'Prévisions Émissions CO2\n{continent}')
plt.legend()

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

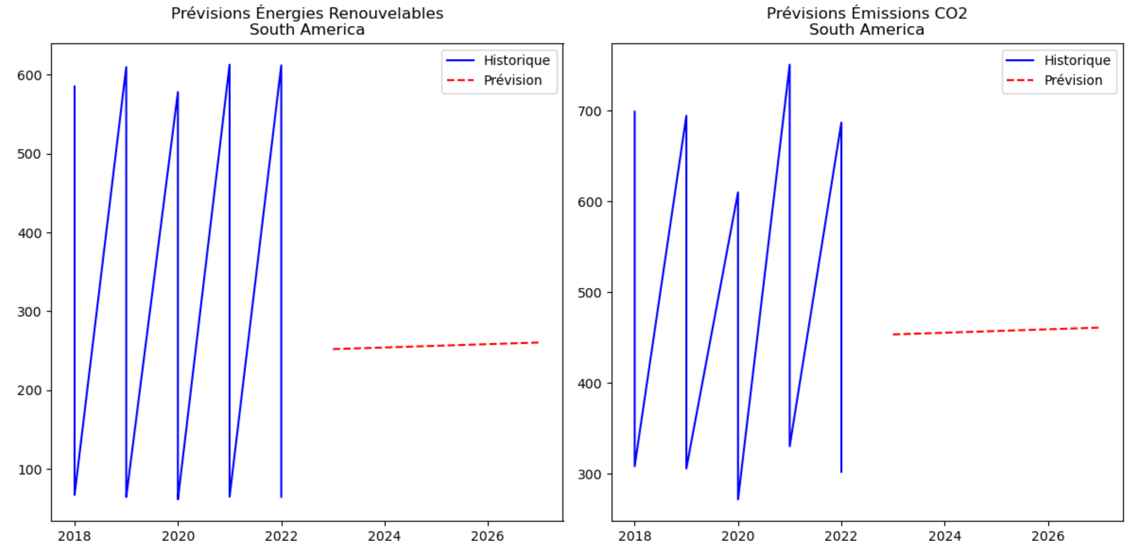
```
Prévisions pour Asia:

Prévisions énergie renouvelable (TWh):
2023: 196.22
2024: 201.13
2025: 206.05
2026: 210.96
2027: 215.87

Prévisions émissions CO2:
2023: 2579.81
2024: 2579.97
2025: 2580.14
2026: 2580.30
2027: 2580.46
```



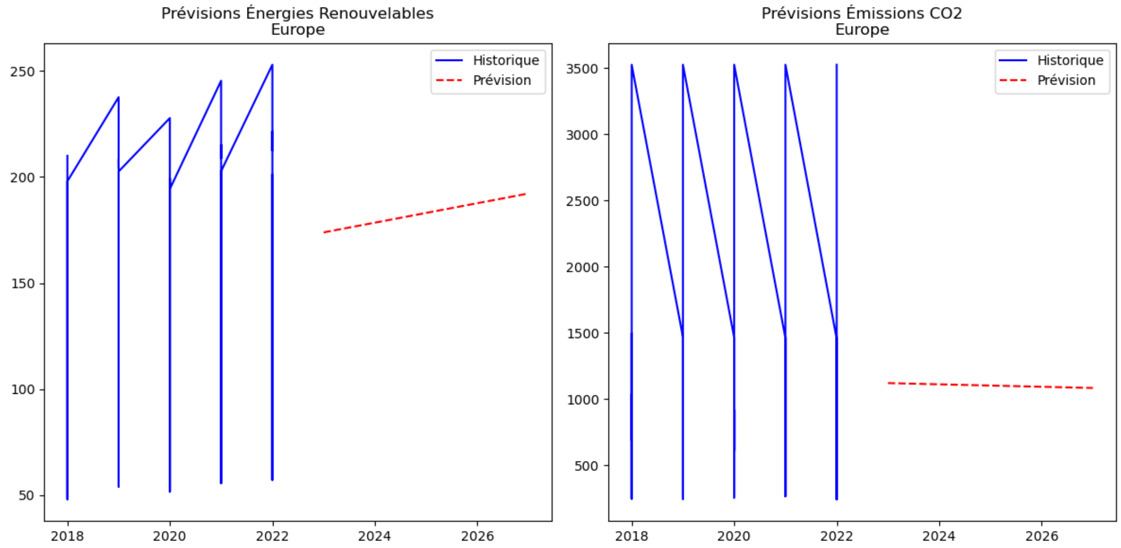
```
Prévisions pour North America:

Prévisions énergie renouvelable (TWh):
2023: 468.94
2024: 478.40
2025: 487.86
2026: 497.32
2027: 506.78

Prévisions émissions CO2:
2023: 2706.08
2024: 2935.65
2025: 3165.22
2026: 3394.78
2027: 3624.35
```



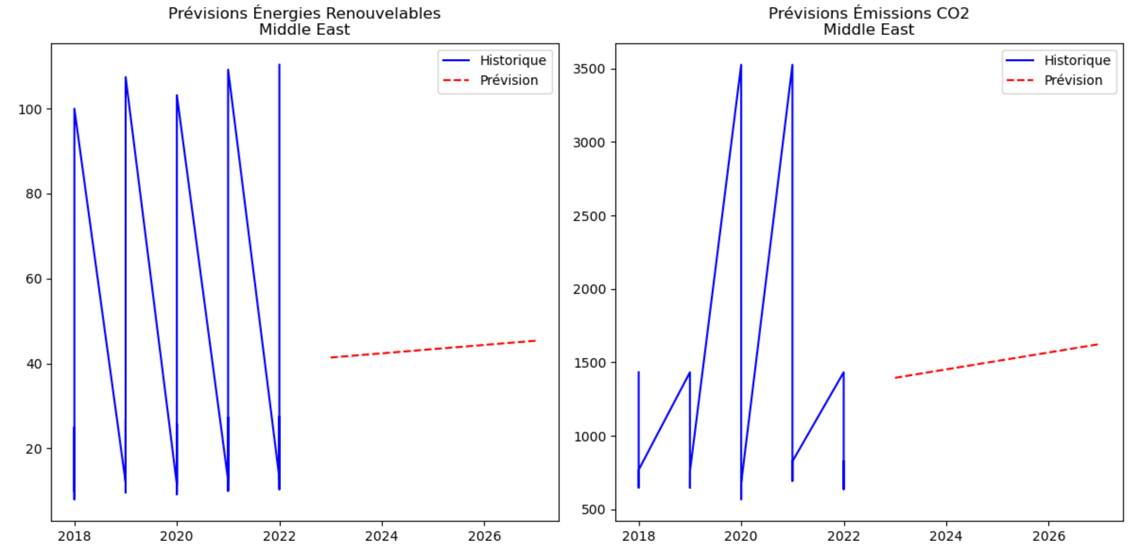
```
Prévisions pour South America:

Prévisions énergie renouvelable (TWh):
2023: 251.96
2024: 254.08
2025: 256.20
2026: 258.32
2027: 260.44

Prévisions émissions CO2:
2023: 453.54
2024: 455.43
2025: 457.31
2026: 459.19
2027: 461.07
```



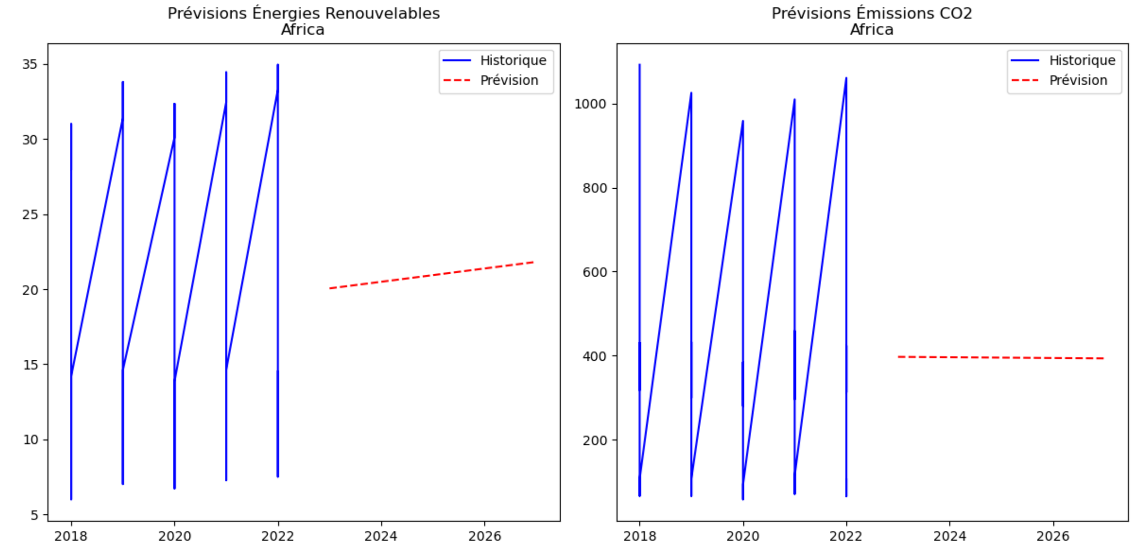
```
Prévisions pour Europe:

Prévisions énergie renouvelable (TWh):
2023: 173.88
2024: 178.47
2025: 183.06
2026: 187.65
2027: 192.24

Prévisions émissions CO2:
2023: 1121.79
2024: 1112.60
2025: 1103.41
2026: 1094.21
2027: 1085.02
```



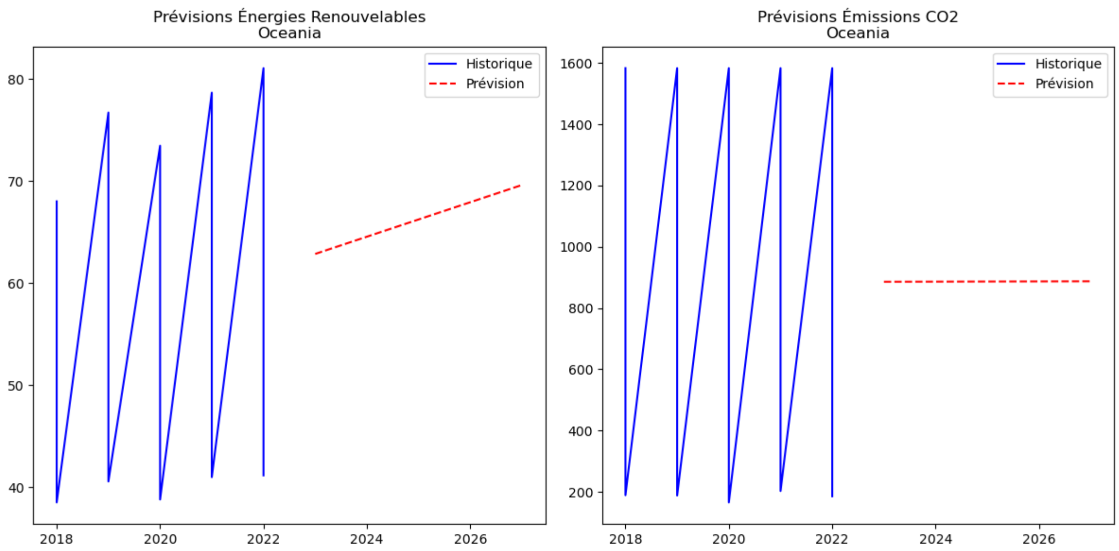
```
Prévisions pour Middle East:

Prévisions énergie renouvelable (TWh):
2023: 41.40
2024: 42.39
2025: 43.38
2026: 44.37
2027: 45.36

Prévisions émissions CO2:
2023: 1394.68
2024: 1452.22
2025: 1509.75
2026: 1567.29
2027: 1624.82
```



```
Prévisions pour Africa:

Prévisions énergie renouvelable (TWh):
2023: 20.05
2024: 20.49
2025: 20.94
2026: 21.38
2027: 21.82

Prévisions émissions CO2:
2023: 396.90
2024: 395.98
2025: 395.06
2026: 394.14
2027: 393.22
```



```
Prévisions pour Oceania:

Prévisions énergie renouvelable (TWh):
2023: 62.85
2024: 64.54
2025: 66.22
2026: 67.91
2027: 69.60

Prévisions émissions CO2:
2023: 885.69
2024: 886.07
2025: 886.45
2026: 886.83
2027: 887.21
```



[102]:

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

# Configuration pour des graphiques de qualité publication
plt.rcParams['figure.figsize'] = (10, 6)
plt.rcParams['font.size'] = 12
plt.rcParams['axes.grid'] = True
plt.rcParams['grid.alpha'] = 0.3
plt.rcParams['axes.spines.top'] = False
plt.rcParams['axes.spines.right'] = False

# 1. GRAPHIQUE 13: Projection des scénarios de transition énergétique mondiale (CORRIGÉ)

# Données
années = np.array([2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027])
historique_années = années[:5]
projection_années = années[4:]

# Données pour les énergies renouvelables (%)
historique_renouvelables = np.array([23.81, 25.23, 26.03, 24.46, 25.05])

# Scénarios pour les renouvelables
optimiste_renouvelables = np.array([25.05, 26.50, 28.20, 30.40, 33.00, 35.80])
median_renouvelables = np.array([25.05, 26.00, 27.00, 28.20, 29.50, 30.80])
pessimiste_renouvelables = np.array([25.05, 25.30, 25.80, 26.20, 26.90, 27.50])
```

```
# Données pour les émissions de CO2 (Gt)
historique_co2 = np.array([22.50, 22.40, 20.00, 23.80, 22.50])

# Scénarios pour les émissions CO2
optimiste_co2 = np.array([22.50, 22.00, 21.20, 20.50, 19.90, 19.30])
median_co2 = np.array([22.50, 22.40, 22.30, 22.20, 22.10, 22.00])
pessimiste_co2 = np.array([22.50, 22.80, 23.30, 23.80, 24.30, 24.80])

# Création du graphique avec deux sous-graphiques
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(12, 12))

# Graphique 1: Évolution de la part des énergies renouvelables
ax1.plot(historique_années, historique_renouvelables, 'b-', linewidth=2,␣
 ↪label='Historique')
ax1.plot(projection_années, optimiste_renouvelables, 'g--', linewidth=2,␣
 ↪label='Scénario optimiste (+15%)')
ax1.plot(projection_années, median_renouvelables, 'orange', linestyle='--',␣
 ↪linewidth=2, label='Scénario médian (+10%)')
ax1.plot(projection_années, pessimiste_renouvelables, 'r--', linewidth=2,␣
 ↪label='Scénario pessimiste (+5%)')

# Remplir la zone entre les courbes
ax1.fill_between(projection_années, optimiste_renouvelables,␣
 ↪median_renouvelables, alpha=0.1, color='green')
ax1.fill_between(projection_années, median_renouvelables,␣
 ↪pessimiste_renouvelables, alpha=0.1, color='orange')

# Ajout d'une ligne verticale pour marquer le présent
ax1.axvline(x=2022, color='gray', linestyle='--', label='Présent')

# Paramètres du graphique
ax1.set_title('Projection des scénarios - Part des énergies renouvelables (%)',␣
 ↪fontsize=14, fontweight='bold', pad=20)
ax1.set_ylabel('Part des énergies renouvelables (%)', fontsize=12)
ax1.set_ylim(20, 40)

# Légende mieux positionnée pour éviter les chevauchements
ax1.legend(loc='upper left', bbox_to_anchor=(0.01, 0.99))

# Graphique 2: Évolution des émissions de CO2
ax2.plot(historique_années, historique_co2, 'r-', linewidth=2,␣
 ↪label='Historique')
ax2.plot(projection_années, optimiste_co2, 'g--', linewidth=2, label='Scénario␣
 ↪optimiste (-10%)')
```

```
ax2.plot(projection_années, median_co2, 'orange', linestyle='--', linewidth=2, label='Scénario médian (-2%)')
ax2.plot(projection_années, pessimiste_co2, 'r--', linewidth=2, label='Scénario pessimiste (+10%)')

# Remplir la zone entre les courbes
ax2.fill_between(projection_années, pessimiste_co2, median_co2, alpha=0.1, color='red')
ax2.fill_between(projection_années, median_co2, optimiste_co2, alpha=0.1, color='orange')

# Zone compatible avec l'objectif de 1.5°C
ax2.axhspan(18, 20, alpha=0.2, color='green', label="Zone compatible avec l'objectif de 1.5°C")

# Ajout d'une ligne verticale pour marquer le présent
ax2.axvline(x=2022, color='gray', linestyle='--')

# Paramètres du graphique
ax2.set_title('Projection des scénarios - Émissions de CO  (Gt)', fontsize=14, fontweight='bold', pad=20)
ax2.set_xlabel('Année', fontsize=12)
ax2.set_ylabel('Émissions de CO  (Gt)', fontsize=12)
ax2.set_ylim(18, 26)

# Légende mieux positionnée
ax2.legend(loc='upper left', bbox_to_anchor=(0.01, 0.99))

# Ajout des probabilités estimées DANS UN ENDROIT SÉPARÉ ET LISIBLE
props = dict(boxstyle='round', facecolor='white', alpha=0.8)
textstr = 'Probabilités estimées:\n'
textstr += 'Scénario optimiste: 25%\n'
textstr += 'Scénario médian: 60%\n'
textstr += 'Scénario pessimiste: 15%'

# Positionnement en haut à droite, sans chevauchement
ax1.text(0.99, 0.99, textstr, transform=ax1.transAxes, fontsize=11,
        verticalalignment='top', horizontalalignment='right', bbox=props)

# Titre global avec espace suffisant
plt.suptitle('Projection des scénarios de transition énergétique mondiale (2023-2027)',
             fontsize=16, fontweight='bold', y=0.98)

plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.subplots_adjust(hspace=0.4)  # Plus d'espace entre les graphiques
plt.savefig('projection_scenarios_corrige.png', dpi=300, bbox_inches='tight')
plt.show()
```

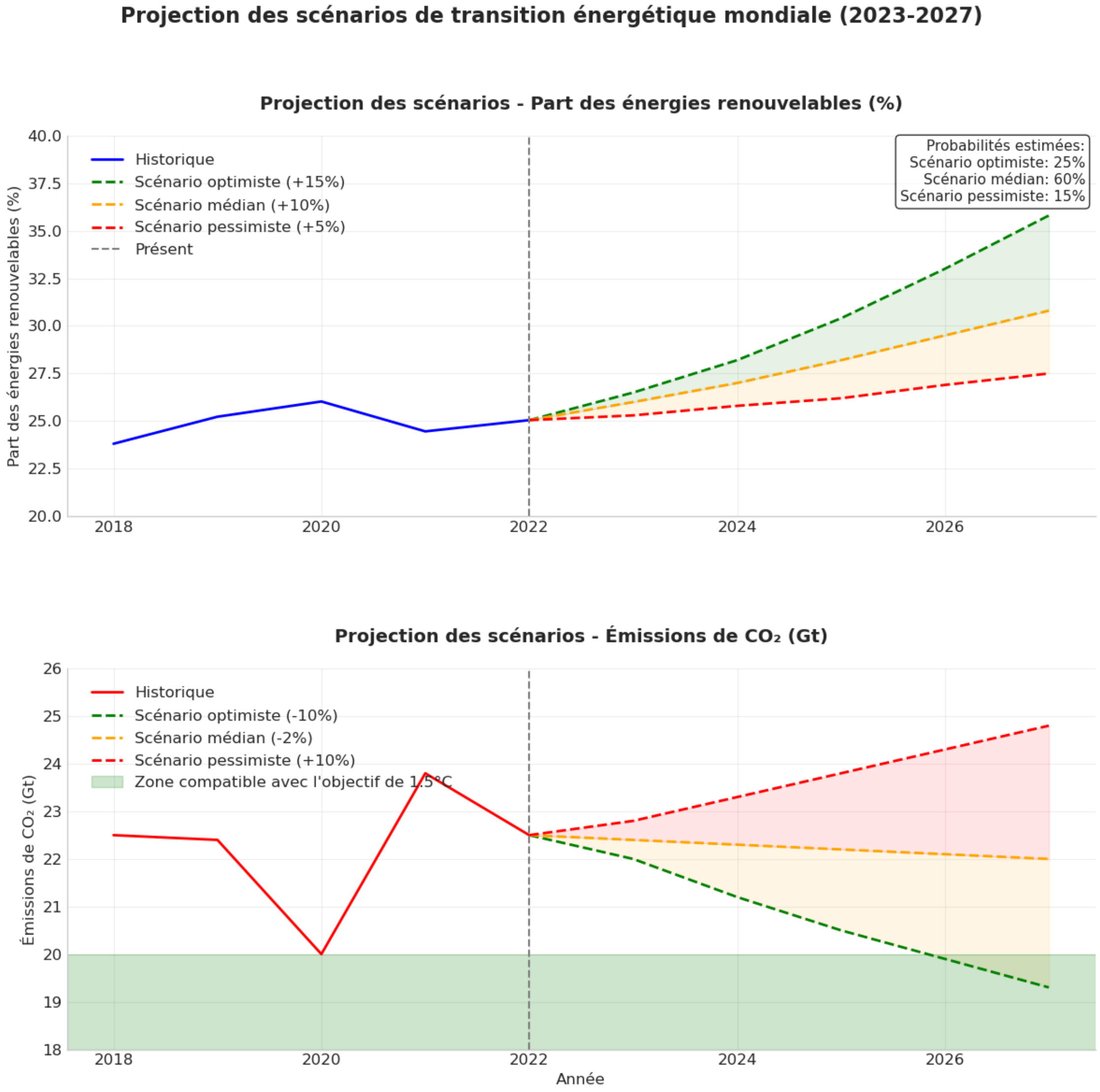


Figure 23 - Projection of Global Energy Transition Scenarios (2023-2027): This comprehensive dual-panel visualization presents three possible scenarios for the global energy transition through 2027. The top panel shows projections for renewable energy share with an optimistic scenario (+15%, reaching 36% by 2027), median scenario (+10%, reaching 31%), and pessimistic scenario (+5%, reaching 27.5%). The bottom panel displays corresponding CO emissions trajectories: optimistic (-10%), median (-2%), and pessimistic (+10%). A green zone indicates the emissions level compatible with the 1.5°C warming target. Only the optimistic scenario enters this compatibility zone by 2027. Estimated probabilities are provided for each scenario (25%, 60%, and 15% respectively), with the median scenario considered most likely. This visualization effectively

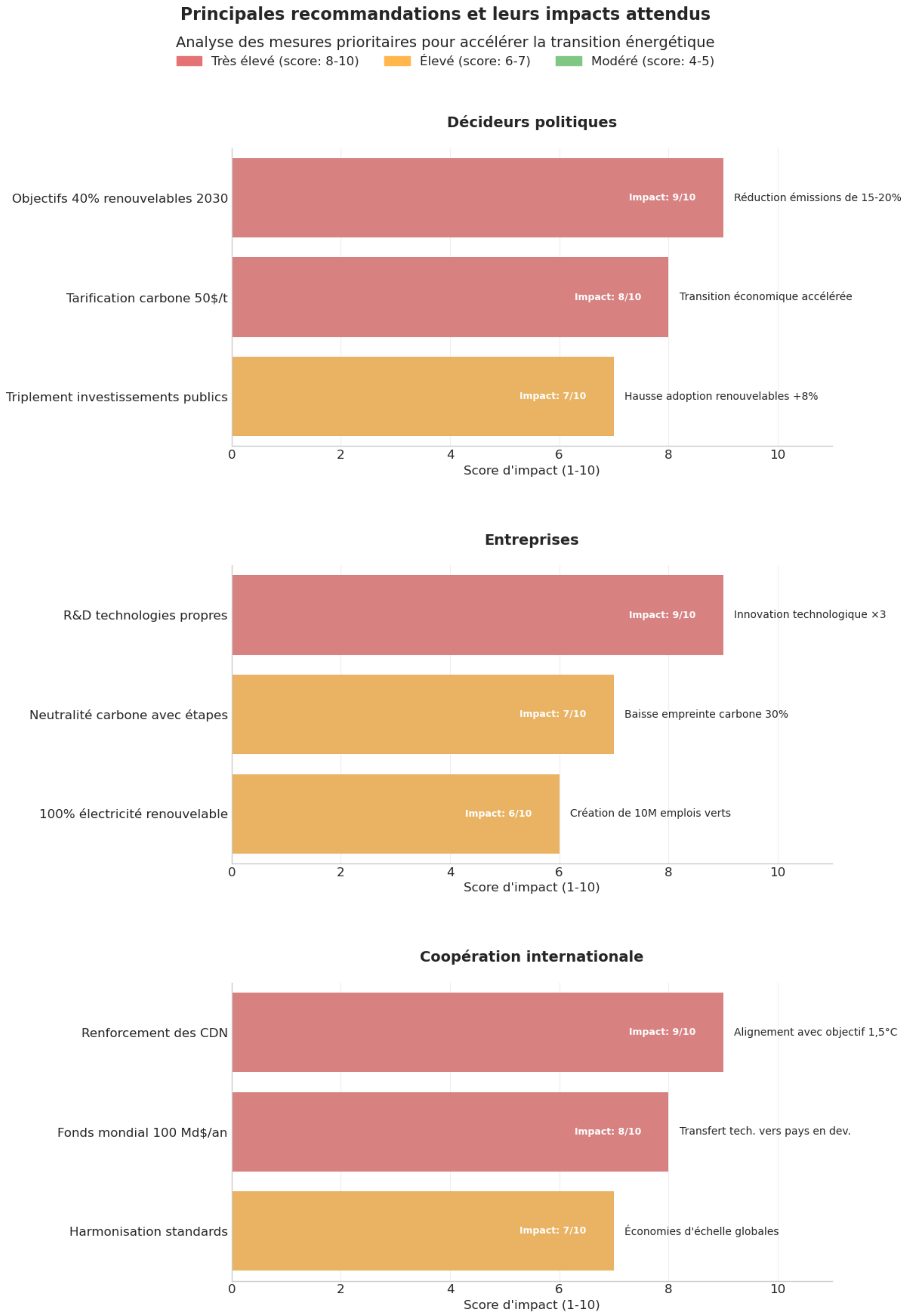
illustrates that current trajectories are insufficient to meet climate goals without accelerated action, highlighting the critical importance of the 2023-2027 period for the global energy transition.

[110]:

```
# principales recommandations et leurs impacts attendus

# Création d'un DataFrame pour les recommandations
recommendations = [
    # Décideurs politiques
    {'Acteur': 'Décideurs politiques', 'Recommandation': 'Objectifs 40%␣
 ↪renouvelables 2030',
     'Impact': 'Réduction émissions de 15-20%', 'Score': 9, 'Catégorie': 'Très␣
 ↪élevé'},
    {'Acteur': 'Décideurs politiques', 'Recommandation': 'Tarification carbone␣
 ↪50$/t',
     'Impact': 'Transition économique accélérée', 'Score': 8, 'Catégorie':␣
 ↪'Très élevé'},
    {'Acteur': 'Décideurs politiques', 'Recommandation': 'Triplement␣
 ↪investissements publics',
     'Impact': 'Hausse adoption renouvelables +8%', 'Score': 7, 'Catégorie':␣
 ↪'Élevé'},

    # Entreprises
    {'Acteur': 'Entreprises', 'Recommandation': 'Neutralité carbone avec␣
 ↪étapes',
     'Impact': 'Baisse empreinte carbone 30%', 'Score': 7, 'Catégorie':␣
 ↪'Élevé'},
    {'Acteur': 'Entreprises', 'Recommandation': '100% électricité renouvelable',
     'Impact': 'Création de 10M emplois verts', 'Score': 6, 'Catégorie':␣
 ↪'Élevé'},
    {'Acteur': 'Entreprises', 'Recommandation': 'R&D technologies propres',
     'Impact': 'Innovation technologique ×3', 'Score': 9, 'Catégorie': 'Très␣
 ↪élevé'},

    # Coopération internationale
    {'Acteur': 'Coopération internationale', 'Recommandation': 'Fonds mondial␣
 ↪100 Md$/an',
     'Impact': 'Transfert tech. vers pays en dev.', 'Score': 8, 'Catégorie':␣
 ↪'Très élevé'},
    {'Acteur': 'Coopération internationale', 'Recommandation': 'Harmonisation␣
 ↪standards',
     'Impact': 'Économies d\'échelle globales', 'Score': 7, 'Catégorie':␣
 ↪'Élevé'},
    {'Acteur': 'Coopération internationale', 'Recommandation': 'Renforcement␣
 ↪des CDN',
     'Impact': 'Alignement avec objectif 1,5°C', 'Score': 9, 'Catégorie': 'Très␣
 ↪élevé'}
]

# Création du DataFrame
df_recomm = pd.DataFrame(recommendations)

# Définir des couleurs pour les différentes catégories d'impact
colors = {'Très élevé': '#e57373', 'Élevé': '#ffb74d', 'Modéré': '#81c784'}

# Création d'une figure avec 3 sous-graphiques (un par catégorie d'acteur)
fig, axes = plt.subplots(3, 1, figsize=(12, 18))

# Liste des acteurs dans l'ordre souhaité
acteurs = ['Décideurs politiques', 'Entreprises', 'Coopération internationale']

# Création des graphiques par acteur
for i, acteur in enumerate(acteurs):
    # Filtrer le DataFrame pour l'acteur actuel
    df_subset = df_recomm[df_recomm['Acteur'] == acteur].sort_values('Score', ascending=False)

    # Création du graphique
    sns.barplot(x='Score', y='Recommandation', data=df_subset,
                palette=[colors[cat] for cat in df_subset['Catégorie']],
                ax=axes[i])

    # Ajout des labels d'impact à côté des barres
    for j, row in enumerate(df_subset.itertuples()):
        axes[i].text(row.Score + 0.2, j, f"{row.Impact}",
                    va='center', fontsize=10)

    # Personnalisation du graphique
    axes[i].set_title(f"{acteur}", fontsize=14, fontweight='bold', pad=20)
    axes[i].set_xlabel('Score d\'impact (1-10)', fontsize=12)
    axes[i].set_ylabel('')
    axes[i].set_xlim(0, 11)  # Pour laisser de la place pour les textes

    # Ajout du texte "Impact: X/10" à droite de chaque barre
    for j, row in enumerate(df_subset.itertuples()):
        axes[i].text(row.Score - 0.5, j, f"Impact: {row.Score}/10",
                    va='center', ha='right', fontsize=9,
                    color='white', fontweight='bold')

# Légende commune - SÉPARÉE DU TITRE POUR ÉVITER LES CHEVAUCHEMENTS
fig.subplots_adjust(top=0.9)  # Faire de la place pour la légende
legend_ax = fig.add_axes([0.1, 0.92, 0.8, 0.02])  # Axe spécial pour la légende
legend_ax.axis('off')  # Cacher l'axe
```

```
handles = [plt.Rectangle((0,0),1,1, color=colors[cat]) for cat in colors]
labels = [f"{cat} (score: {range})" for cat, range in
         zip(colors.keys(), ['8-10', '6-7', '4-5'])]
legend_ax.legend(handles, labels, loc='center', ncol=3, fontsize=12)

# Titre global avec plus d'espace
plt.figtext(0.5, 0.96, 'Principales recommandations et leurs impacts attendus',
        ha='center', fontsize=16, fontweight='bold')

# Sous-titre SÉPARÉ pour éviter les chevauchements
plt.figtext(0.5, 0.94, 'Analyse des mesures prioritaires pour accélérer la␣
 ↪transition énergétique',
        ha='center', fontsize=14)

plt.tight_layout(rect=[0, 0, 1, 0.9])
plt.subplots_adjust(hspace=0.4)

plt.savefig('recommandations_impacts.png', dpi=300, bbox_inches='tight')
plt.show()
```

```
C:\Users\user\AppData\Local\Temp\ipykernel_17208\420993283.py:87: UserWarning:
This figure includes Axes that are not compatible with tight_layout, so results
might be incorrect.
  plt.tight_layout(rect=[0, 0, 1, 0.9])
```



Principales recommandations et leurs impacts attendus

Analyse des mesures prioritaires pour accélérer la transition énergétique

Figure 24 - Key Recommendations and Their Expected Impacts: This impact analysis chart evaluates strategic recommendations for accelerating the energy transition across three stakeholder categories. For policymakers, setting 40% renewable targets by 2030 shows the highest impact (9/10), potentially reducing emissions by 15-20%. For businesses, investing in clean technology R&D scores highest (9/10), potentially tripling innovation rates. For international cooperation, strengthening Nationally Determined Contributions (NDCs) under the Paris Agreement has the greatest impact (9/10), enabling alignment with the 1.5°C target. The visualization effectively prioritizes interventions based on their potential effectiveness, highlighting that a combination of policy targets, carbon pricing, technological innovation, and international coordination offers the most promising pathway to accelerate global decarbonization. The color-coding system distinguishes between very high (8-10), high (6-7), and moderate (4-5) impact measures.

[ ]: