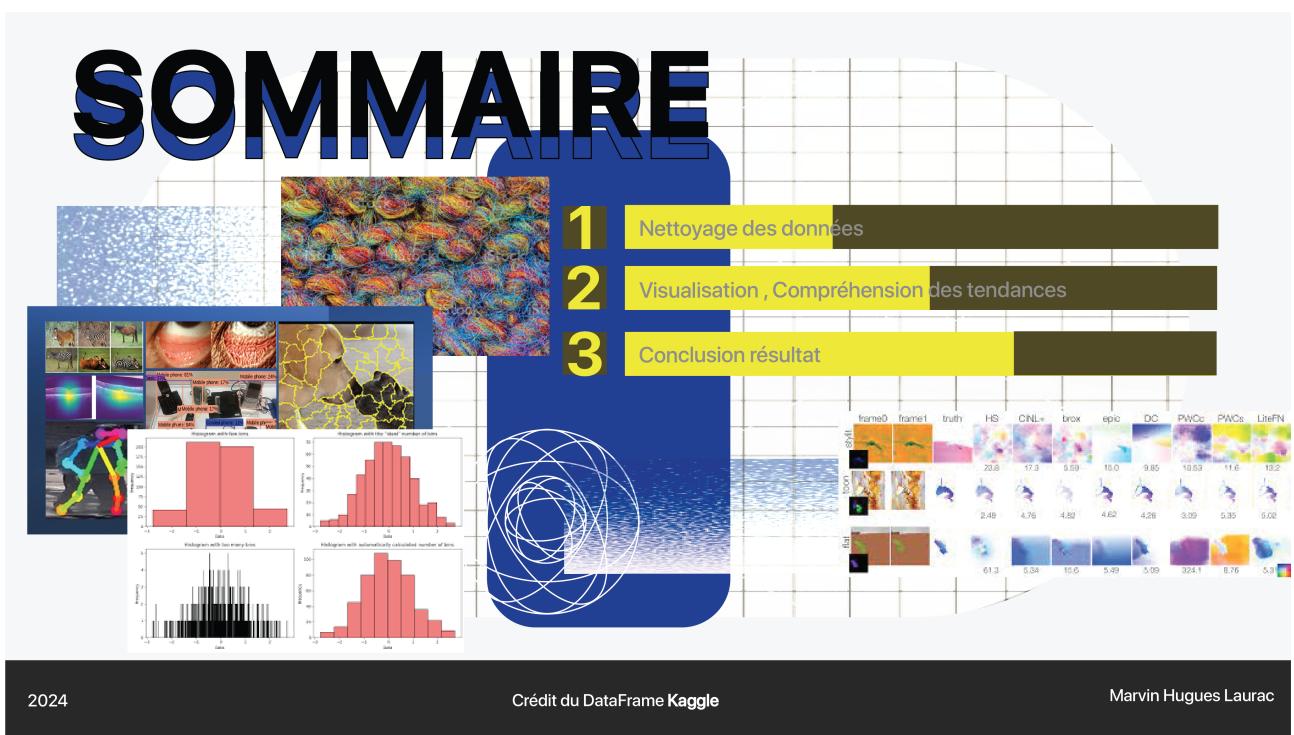




INTRODUCTION

Notre monde d'aujourd'hui évolue à une vitesse effrénée. Où la croissance économique se trouve au centre des préoccupations des entreprises et des gouvernements, tout en prenant conscience des impacts environnementaux. De plus, en Europe, nous sommes dans un virage au profit de la RSE, il est donc nécessaire d'identifier les actions mises en place.

Ce projet a pour but de créer des visualisations sur les actions mises en place au profit des énergies renouvelables, avec des méthodes d'affichage.



LOGICIEL PRIVILÉGIÉ

- Python

J'ai utilisé Python avec ses bibliothèques comme Pandas pour l'analyse de données et Matplotlib pour la visualisation.

VISUALISATION DES ÉNERGIES RENOUVELABLES PAR PAYS

CONTEXTE DU SUJET

Ce projet représente l'utilisation et la contribution des pays ou organisations aux énergies renouvelables dans l'approvisionnement total en énergie primaire TPES.

Le TPES est l'ensemble des ressources énergétiques mobilisées par un pays ou une région.

Ce sont l'énergie hydraulique, la géothermie, le solaire, l'éolien, ainsi que l'énergie provenant des marées et des vagues. Mais aussi les biocarburants, qui sont dérivés de la biomasse, et incluent également les déchets végétaux et animaux pour la production de chaleur et d'électricité.

Pour ne pas rendre les choses trop compliquées, je retiens seulement le terme énergie primaire.

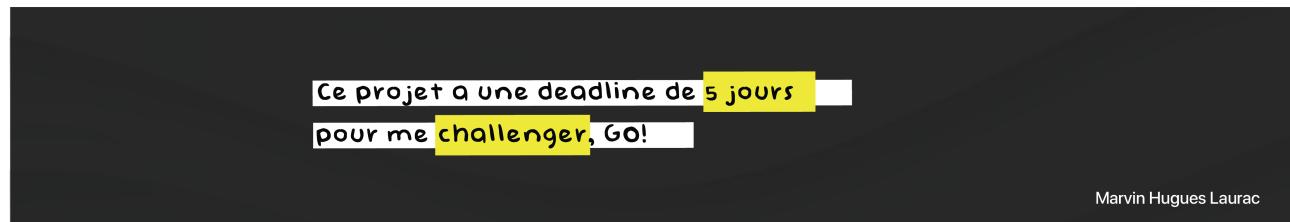
À noter : Ces informations ont été indiquées par l'auteur. De mon côté, j'ai travaillé sur ce **DataFrame** en prenant en compte ces indications, en les reformulant et en faisant mes recherches de mon côté

DATAFRAME INFO

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15904 entries, 0 to 15903
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   LOCATION    15904 non-null   object 
 1   INDICATOR   15904 non-null   object 
 2   SUBJECT     15904 non-null   object 
 3   MEASURE     15904 non-null   object 
 4   FREQUENCY   15904 non-null   object 
 5   TIME        15904 non-null   int64  
 6   Value       12817 non-null   float64
 7   Flag Codes  3887 non-null   object  
dtypes: float64(1), int64(1), object(6)
memory usage: 994.1+ KB
```

LICENCE

kaggle (Database Contents License (DbCL) v1.0)



```
Entrée [139]: import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

Entrée [140]: #je commence par importer le DataFrame
df = pd.read_csv("/Users/octoberone/Desktop/DataFrame GitHub/DataFrame/Visualisation des énergies renouvelables par

Entrée [141]: #affichage des 5 premières lignes
df.head()

Out[141]:   LOCATION INDICATOR SUBJECT MEASURE FREQUENCY TIME   Value Flag Codes
0      AUS  RENEWABLE      TOT    KTOE        A  1960  4436.932      NaN
1      AUS  RENEWABLE      TOT    KTOE        A  1961  4490.510      NaN
2      AUS  RENEWABLE      TOT    KTOE        A  1962  4407.097      NaN
3      AUS  RENEWABLE      TOT    KTOE        A  1963  4628.738      NaN
4      AUS  RENEWABLE      TOT    KTOE        A  1964  4497.396      NaN

Entrée [142]: #voici le nombre de lignes et de colonnes sans le détail
df.shape

Out[142]: (15904, 8)
```

VISUALISATION DES ÉNERGIES RENOUVELABLES PAR PAYS

Entrée [143]: #voici les détails des colonnes avec beaucoup plus d'informations (variables, types, le nombre de valeurs nulles)
df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15904 entries, 0 to 15903
Data columns (total 8 columns):
 #   Column    Non-Null Count  Dtype  
--- 
 0   LOCATION   15904 non-null   object 
 1   INDICATOR  15904 non-null   object 
 2   SUBJECT    15904 non-null   object 
 3   MEASURE    15904 non-null   object 
 4   FREQUENCY  15904 non-null   object 
 5   TIME       15904 non-null   int64  
 6   Value      12017 non-null   float64
 7   Flag Codes 3887 non-null   object  
dtypes: float64(1), int64(1), object(6)
memory usage: 994.1+ KB
```

Entrée [144]: #dans cette partie, j'ai affiché les pays pour pouvoir visualiser quels étaient les pays présents dans ces data
#après observation, je constate qu'il n'y a pas tous les pays, dont la suisse ou l'allemande
print(df["LOCATION"].unique())

```
['AUS' 'AUT' 'BEL' 'CAN' 'CZE' 'DNK' 'FIN' 'FRA' 'DEU' 'GRC' 'HUN' 'ISL'
 'IRL' 'ITA' 'JPN' 'KOR' 'LUX' 'MEX' 'NLD' 'NZL' 'NOR' 'POL' 'PRT' 'SVK'
 'ESP' 'SWE' 'CHE' 'TUR' 'GBR' 'USA' 'OEU' 'ALB' 'DZA' 'ARG' 'ARM' 'AZE'
 'BGD' 'BLR' 'BIH' 'BRA' 'BRN' 'BGR' 'KHM' 'CHL' 'CHN' 'COL' 'HRV' 'CYP'
 'EGY' 'EST' 'ETH' 'GEO' 'GHA' 'HTI' 'HKG' 'IND' 'IDN' 'IRN' 'ISR' 'KAZ'
 'LVA' 'LTU' 'MKD' 'MYS' 'MLT' 'MDA' 'MOZ' 'NGA' 'PAK' 'PRY' 'PER' 'PHL'
 'ROU' 'RUS' 'SAU' 'SGP' 'SVN' 'ZAF' 'SDN' 'TWN' 'TZA' 'THA' 'UKR' 'ARE'
 'URY' 'VNM' 'ZMB' 'WLD' 'SRB' 'MNE' 'G20' 'EU28' 'OECD' 'AGO' 'BHR' 'BEN'
 'BOL' 'BWA' 'CMR' 'COG' 'CRI' 'CIV' 'CUB' 'PRK' 'COD' 'DOM' 'ECU' 'SLV'
 'ERI' 'GAB' 'GTM' 'HND' 'IRQ' 'JAM' 'JOR' 'KEN' 'KWT' 'KGZ' 'LBN' 'LBY'
 'MNG' 'MAR' 'MMR' 'NAM' 'NPL' 'NIC' 'NER' 'OMN' 'PAN' 'QAT' 'SEN' 'LKA'
 'SYR' 'TJK' 'TGO' 'TT0' 'TUN' 'TKM' 'UZB' 'VEN' 'YEM' 'ZWE']
```

Entrée [145]: #j'ai utilisé cette commande pour être sûr qu'il n'y a pas de valeurs manquantes
df.isnull().sum()
#je constate que les colonnes "Value" et "Flag Codes" disposent de None

Out[145]:

Column	0
LOCATION	0
INDICATOR	0
SUBJECT	0
MEASURE	0
FREQUENCY	0
TIME	0
Value	3887
Flag Codes	12017
dtype: int64	

Entrée [146]: #dans cette partie, je décide de supprimer les valeurs None pour avoir des données exploitables et justes
df.dropna(subset=["Value"], inplace=True)

Entrée [147]: #je décide d'afficher les données dans la colonne "Flag Codes" car j'ai rencontré un problème lors de l'affichage
#en supprimant les données None, cela fausse tous les résultats suivants
print(df["Flag Codes"])
#j'observe qu'il y a beaucoup de données None et donc cette colonne risque de ne pas être pertinente pour pouvoir ê

```
135    NaN
136    NaN
137    NaN
138    NaN
139    NaN
140    NaN
141    NaN
142    NaN
143    NaN
144    NaN
145    NaN
146    NaN
147    NaN
148    NaN
149    NaN
150    NaN
151    NaN
152    NaN
153    NaN
154    NaN
```

VISUALISATION DES ÉNERGIES RENOUVELABLES PAR PAYS

```
Entrée [148]: #je vérifie par la suite si les données de la colonne "Value" ont bien été supprimées
#les données ont bien été supprimées car nous avons 0
df.isnull().sum()
```

```
Out[148]: LOCATION      0
INDICATOR      0
SUBJECT       0
MEASURE        0
FREQUENCY      0
TIME          0
Value          0
Flag Codes    12017
dtype: int64
```

```
Entrée [149]: #Afin d'avoir plus de détails statistiques, j'affiche les colonnes numériques
```

```
#count%:nombre de valeurs nulles
#mean%:la moyenne des valeurs
#std%:l'écart type (dispersion valeurs)
#min%:le min de la colonne
#25%:1er quartile à moins de 25
#50%:2er quartile à moins de 25
#75%:3er quartile à moins de 25
#max:le maxi de la colonne

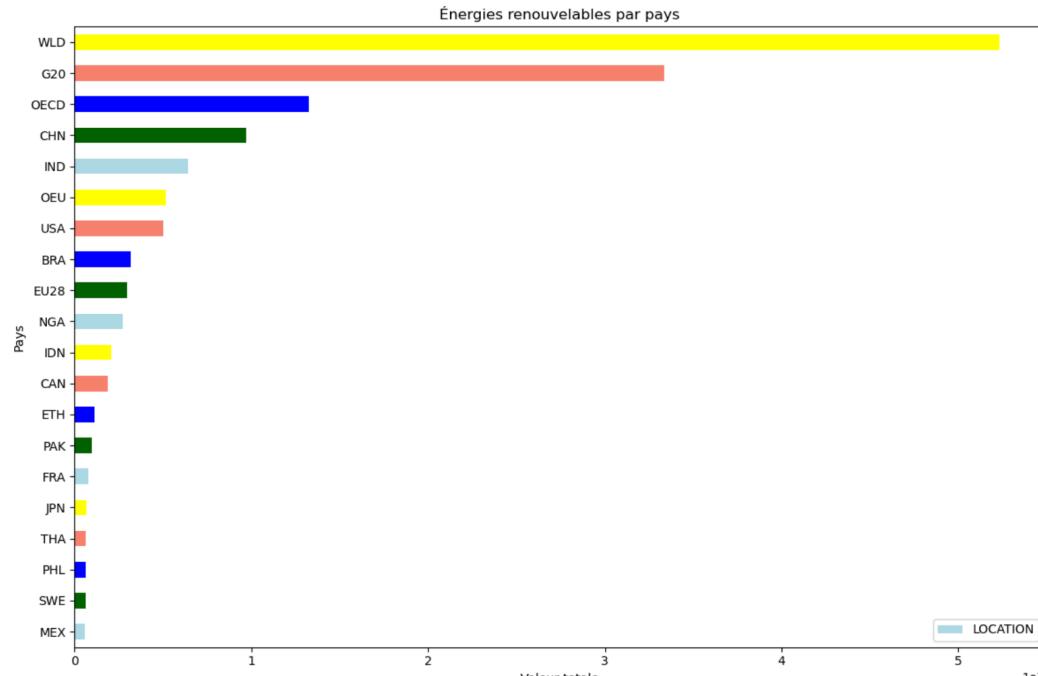
print(df.describe())
```

	TIME	Value
count	12017.000000	1.201700e+04
mean	1992.454273	1.322108e+04
std	13.756033	9.203502e+04
min	1960.000000	0.000000e+00
25%	1981.000000	9.580000e+00
50%	1993.000000	7.486000e+01
75%	2004.000000	1.816018e+03
max	2015.000000	1.894019e+06

```
Entrée [163]: #dans cette première visualisation, je compte les 20 premiers pays avec le plus de données pertinentes en comptant
#de plus, pour avoir une meilleure visualisation, je les classe par ordre décroissant
#j'en déduis que nous avons WLD, que je suppose représenter le monde
#G20 pour les pays du G20
#OECD est une organisation par pays
#ces 3 données sont des données regroupées
```

```
#mais nous pouvons observer que la Chine est en première position, l'Inde en 2ème, les USA en 4ème, le Canada en 9ème
df_top = df.groupby("LOCATION")["Value"].sum().nlargest(20).sort_values(ascending=True)
```

```
plt.figure(figsize=(12, 8))
df_top.plot(kind="barh", color=["lightblue", "darkgreen", "blue", "salmon", "yellow"], label = "LOCATION")
plt.title("Énergies renouvelables par pays")
plt.ylabel("Pays")
plt.xlabel("Valeur totale")
plt.tight_layout()
plt.legend()
plt.show()
```



VISUALISATION DES ÉNERGIES RENOUVELABLES PAR PAYS

```
Entrée [134]: #afin de me concentrer seulement sur les pays voisins de la France, je crée un nouveau dataframe regroupant ces voisins
#malheureusement, tous les pays voisins ne sont pas enregistrés dans la data
location_euro = ["FRA", "ESP", "BEL", "ITA"]
df_proche_france = df[df['LOCATION'].isin(location_euro)]
df_proche_france.head()
```

Out[134]:	LOCATION	INDICATOR	SUBJECT	MEASURE	FREQUENCY	TIME	Value	Flag Codes
112	BEL	RENEWABLE	TOT	KTOE	A	1960	14.792	NaN
113	BEL	RENEWABLE	TOT	KTOE	A	1961	16.254	NaN
114	BEL	RENEWABLE	TOT	KTOE	A	1962	14.018	NaN
115	BEL	RENEWABLE	TOT	KTOE	A	1963	12.384	NaN
116	BEL	RENEWABLE	TOT	KTOE	A	1964	9.890	NaN

```
Entrée [135]: #dans cette dernière partie, j'affiche les pays que j'ai regroupés afin de visualiser les pays proches de la France
#j'en déduis que la France est le pays qui utilise le plus d'énergie renouvelable contrairement à ses confrères
plt.figure(figsize=(12, 8))
sns.barplot(data=df_proche_france, x="Value", y="LOCATION")
plt.title("Énergies renouvelables par pays (autour de la France)")
plt.xlabel("Valeur totale")
plt.ylabel("Pays")
plt.tight_layout()
plt.show()
```

