In [1]:
```
"""
Created on Wed Oct  9 18:51:52 2024

@author: Thierry ALLEM
"""
```

Out[1]:
```
'\nCreated on Wed Oct  9 18:51:52 2024\n\n@author: Thierry ALLEM\n'
```

In [2]:
```
# ************************************** CALCULS DES CONSOMMATIONS CUMULEE - AJOUT DES DONNEES METEOROLOGIQUES ET D'INDICATEURS ****
```

In [3]:
```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

In [4]:
```
# Importation du fichier eco2mix avec consommations
# Lecture du fichier CSV
df_eco_2_cumuls_delta = pd.read_csv("df_eco2_powerbi.csv", sep=';', encoding='latin-1')
df_eco_2_cumuls_delta.head()
```

Out[4]:

	Unnamed: 0	region_abr	region	date_heure	annee	annee_mois	mois	mois_nom	semaine	jour	...	tch_consommation	tch_consommation_isol	tch_6f_rte	tch_consomma
0	0	CVL	CENTRE VAL DE LOIRE	2013-01-01 00:30:00	2013	2013-01	1	Jan	0	1	...	19.36	76.19	566.23	
1	1	PDL	PAYS DE LA LOIRE	2013-01-01 00:30:00	2013	2013-01	1	Jan	0	1	...	131.30	12.53	10.48	
2	2	GES	GRAND EST	2013-01-01 00:30:00	2013	2013-01	1	Jan	0	1	...	26.18	63.80	58.65	
3	3	IDF	ILE DE FRANCE	2013-01-01 00:30:00	2013	2013-01	1	Jan	0	1	...	377.13	34.89	50.93	
4	4	OCC	OCCITANIE	2013-01-01 00:30:00	2013	2013-01	1	Jan	0	1	...	50.96	38.55	39.55	

5 rows × 65 columns

In [5]:
```
df_eco_2_cumuls_delta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2121396 entries, 0 to 2121395
Data columns (total 65 columns):
 #   Column                       Dtype
---  ------                       -----
 0   Unnamed: 0                   int64
 1   region_abr                   object
 2   region                       object
 3   date_heure                   object
 4   annee                        int64
 5   annee_mois                   object
 6   mois                         int64
 7   mois_nom                     object
 8   semaine                      int64
 9   jour                         int64
 10  jour_nom                     object
 11  jour_numero                  int64
 12  heure                        object
 13  jour_fractionnel             float64
 14  consommation                 float64
 15  thermique                    float64
 16  nucleaire                    float64
 17  eolien                       float64
 18  solaire                      float64
 19  hydraulique                  float64
 20  bioenergies                  float64
 21  pompage                      float64
 22  ech_physiques                float64
 23  export                       float64
 24  import                       float64
 25  stockage_batterie            float64
 26  destockage_batterie          float64
 27  tch_thermique_rte            float64
 28  tch_nucleaire_rte            float64
 29  tch_eolien_rte               float64
 30  tch_solaire_rte              float64
 31  tch_hydraulique_rte          float64
 32  tch_bioenergies_rte          float64
 33  cap_prod_max_thermique       float64
 34  thermique_p_disp             float64
 35  cap_prod_max_nucleaire       float64
 36  nucleaire_p_disp             float64
 37  eolien_p_disp                float64
 38  solaire_p_disp               float64
 39  cap_prod_max_hydraulique     float64
 40  ind_prod_hydraulique         float64
 41  hydraulique_p_disp           float64
 42  cap_prod_max_bioenergies     float64
 43  bioenergies_p_disp           float64
 44  region_p_max                 float64
 45  cmax_thermique_rte           float64
```

```
 46  cmax_nucleaire_rte          float64
 47  cmax_eolien_rte             float64
 48  cmax_solaire_rte            float64
 49  cmax_hydraulique_rte        float64
 50  cmax_bioenergies_rte        float64
 51  jour_ferie                  bool
 52  tco_6f                      float64
 53  consommation_isol           float64
 54  tch_6f                      float64
 55  tch_consommation            float64
 56  tch_consommation_isol       float64
 57  tch_6f_rte                  float64
 58  tch_consommation_rte        float64
 59  saison                      object
 60  clst_tch_6f                 object
 61  clst_tch_consommation       object
 62  clst_tch_consommation_isol  object
 63  date_heure_modifiee         int64
 64  reference                   object
dtypes: bool(1), float64(45), int64(7), object(12)
memory usage: 1.0+ GB
```

In [6]:

```python
# Suppression de colonnes non utiles dans cette étude
colonnes_a_supprimer = [
    'Unnamed: 0', 'region_abr','mois_nom','tch_6f_rte','tch_consommation_rte',
    'tch_thermique_rte','tch_nucleaire_rte','tch_eolien_rte','tch_solaire_rte','tch_hydraulique_rte','tch_bioenergies_rte','cap_prod_max_thermique',
    'thermique_p_disp','cap_prod_max_nucleaire',
    'nucleaire_p_disp','eolien_p_disp','solaire_p_disp','ind_prod_hydraulique','hydraulique_p_disp','cap_prod_max_bioenergies',
    'cap_prod_max_hydraulique','bioenergies_p_disp','region_p_max','cmax_thermique_rte','cmax_nucleaire_rte',
    'cmax_eolien_rte','cmax_solaire_rte','cmax_hydraulique_rte','cmax_bioenergies_rte','jour_ferie', 'saison','clst_tch_6f',
    'clst_tch_consommation','clst_tch_consommation_isol','date_heure_modifiee']
df_eco_2_cumuls_delta.drop(columns=colonnes_a_supprimer, inplace=True)
df_eco_2_cumuls_delta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2121396 entries, 0 to 2121395
Data columns (total 30 columns):
 #   Column                 Dtype
---  ------                 -----
 0   region                 object
 1   date_heure             object
 2   annee                  int64
 3   annee_mois             object
 4   mois                   int64
 5   semaine                int64
 6   jour                   int64
 7   jour_nom               object
 8   jour_numero            int64
 9   heure                  object
 10  jour_fractionnel       float64
 11  consommation           float64
 12  thermique              float64
 13  nucleaire              float64
 14  eolien                 float64
 15  solaire                float64
 16  hydraulique            float64
 17  bioenergies            float64
 18  pompage                float64
 19  ech_physiques          float64
 20  export                 float64
 21  import                 float64
 22  stockage_batterie      float64
 23  destockage_batterie    float64
 24  tco_6f                 float64
 25  consommation_isol      float64
 26  tch_6f                 float64
 27  tch_consommation       float64
 28  tch_consommation_isol  float64
 29  reference              object
dtypes: float64(19), int64(5), object(6)
memory usage: 485.5+ MB
```

In [7]:
```python
# Conversion de la colonne 'date_heure' en datetime
df_eco_2_cumuls_delta['date_heure'] = pd.to_datetime(df_eco_2_cumuls_delta['date_heure'])
```

In [8]:
```python
# Ajour d'une colonne de nom de mois, en français

# Dictionnaire de correspondance des noms de mois en français (avec majuscule)
mois_mapping = {
    1: 'Janvier', 2: 'Février', 3: 'Mars', 4: 'Avril', 5: 'Mai', 6: 'Juin',
    7: 'Juillet', 8: 'Août', 9: 'Septembre', 10: 'Octobre', 11: 'Novembre', 12: 'Décembre'
}
```

```
# Ajout de la colonne 'mois_nom'
df_eco_2_cumuls_delta['mois_nom'] = df_eco_2_cumuls_delta['date_heure'].dt.month.map(mois_mapping)
```

In [9]:
```
# Extraction de la date uniquement (sans les heures)
df_eco_2_cumuls_delta['date'] = df_eco_2_cumuls_delta['date_heure'].dt.strftime('%Y-%m-%d')
```

In [10]:
```
# Ajout d'une colonne 'heure'
df_eco_2_cumuls_delta['heure'] = df_eco_2_cumuls_delta['date_heure'].dt.strftime('%H:%M')
df_eco_2_cumuls_delta['heure'].unique()
df_eco_2_cumuls_delta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2121396 entries, 0 to 2121395
Data columns (total 32 columns):
 #   Column                 Dtype
---  ------                 -----
 0   region                 object
 1   date_heure             datetime64[ns]
 2   annee                  int64
 3   annee_mois             object
 4   mois                   int64
 5   semaine                int64
 6   jour                   int64
 7   jour_nom               object
 8   jour_numero            int64
 9   heure                  object
 10  jour_fractionnel       float64
 11  consommation           float64
 12  thermique              float64
 13  nucleaire              float64
 14  eolien                 float64
 15  solaire                float64
 16  hydraulique            float64
 17  bioenergies            float64
 18  pompage                float64
 19  ech_physiques          float64
 20  export                 float64
 21  import                 float64
 22  stockage_batterie      float64
 23  destockage_batterie    float64
 24  tco_6f                 float64
 25  consommation_isol      float64
 26  tch_6f                 float64
 27  tch_consommation       float64
 28  tch_consommation_isol  float64
 29  reference              object
 30  mois_nom               object
 31  date                   object
dtypes: datetime64[ns](1), float64(19), int64(5), object(7)
memory usage: 517.9+ MB
```

In [11]:
```
# Filtrage du dataframe sur les années 2013 à 2022

df_eco_2_cumuls_delta= df_eco_2_cumuls_delta[(df_eco_2_cumuls_delta['annee']>=2013) & (df_eco_2_cumuls_delta['annee']<=2022)]
```

In [12]:
```
# ================================================================================
# Importation du fichier des relevés de températures et de vitesse du vent
```

In [13]:
```
df_temperatures = pd.read_csv("df_meteo_interpolated.csv", sep=',', encoding='latin-1')
df_temperatures.head()
```

Out[13]:

	date_heure	region	Vitesse du vent moyen 10 mn	temperature_C	temperature_ressentie
0	2013-01-01 00:00:00	AUVERGNE RHONE ALPES	8.475000	7.375000	3.347231
1	2013-01-01 00:30:00	AUVERGNE RHONE ALPES	8.475000	7.375000	3.347231
2	2013-01-01 01:00:00	AUVERGNE RHONE ALPES	8.475000	7.375000	3.347231
3	2013-01-01 01:30:00	AUVERGNE RHONE ALPES	8.154167	7.416667	3.548545
4	2013-01-01 02:00:00	AUVERGNE RHONE ALPES	7.833333	7.458333	3.749860

In [14]:
```
df_temperatures.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2103540 entries, 0 to 2103539
Data columns (total 5 columns):
 #   Column                      Dtype  
---  ------                      -----  
 0   date_heure                  object 
 1   region                      object 
 2   Vitesse du vent moyen 10 mn  float64
 3   temperature_C               float64
 4   temperature_ressentie       float64
dtypes: float64(3), object(2)
memory usage: 80.2+ MB
```

In [15]:
```
# Conversion des dates
df_temperatures['date_heure'] = pd.to_datetime(df_temperatures['date_heure'])
```

In [16]:
```
# Renommage des colonnes dans 'df_temperatures' pour correspondre aux noms dans 'df_eco_2_cumuls_delta'
df_temperatures.rename(columns={ 'Vitesse du vent moyen 10 mn': 'vent_10m_vitesse_moyen'}, inplace=True)
df_temperatures.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2103540 entries, 0 to 2103539
Data columns (total 5 columns):
 #   Column                  Dtype
---  ------                  -----
 0   date_heure              datetime64[ns]
 1   region                  object
 2   vent_10m_vitesse_moyen  float64
 3   temperature_C           float64
 4   temperature_ressentie   float64
dtypes: datetime64[ns](1), float64(3), object(1)
memory usage: 80.2+ MB
```

In [17]:

```python
#=======================FUSION ============================

# Fusion avec le DataFrame des températures
df_conso_temp = pd.merge(
    df_eco_2_cumuls_delta,
    df_temperatures[['date_heure','region','vent_10m_vitesse_moyen', 'temperature_C', 'temperature_ressentie']],
    on=['region', 'date_heure'],
    how='left')
df_conso_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2103540 entries, 0 to 2103539
Data columns (total 35 columns):
 #   Column                  Dtype
---  ------                  -----
 0   region                  object
 1   date_heure              datetime64[ns]
 2   annee                   int64
 3   annee_mois              object
 4   mois                    int64
 5   semaine                 int64
 6   jour                    int64
 7   jour_nom                object
 8   jour_numero             int64
 9   heure                   object
 10  jour_fractionnel        float64
 11  consommation            float64
 12  thermique               float64
 13  nucleaire               float64
 14  eolien                  float64
 15  solaire                 float64
 16  hydraulique             float64
 17  bioenergies             float64
 18  pompage                 float64
 19  ech_physiques           float64
 20  export                  float64
 21  import                  float64
 22  stockage_batterie       float64
 23  destockage_batterie     float64
 24  tco_6f                  float64
 25  consommation_isol       float64
 26  tch_6f                  float64
 27  tch_consommation        float64
 28  tch_consommation_isol   float64
 29  reference               object
 30  mois_nom                object
 31  date                    object
 32  vent_10m_vitesse_moyen  float64
 33  temperature_C           float64
 34  temperature_ressentie   float64
dtypes: datetime64[ns](1), float64(22), int64(5), object(7)
memory usage: 561.7+ MB
```

In [18]:
```
# Recherche de valeurs manquantes
```

In [19]:
```
df_conso_temp.isna().sum()
```

Out[19]:
```
region                    0
date_heure                0
annee                     0
annee_mois                0
mois                      0
semaine                   0
jour                      0
jour_nom                  0
jour_numero               0
heure                     0
jour_fractionnel          0
consommation              0
thermique                 0
nucleaire                 0
eolien                    0
solaire                   0
hydraulique               0
bioenergies               0
pompage                   0
ech_physiques             0
export                    0
import                    0
stockage_batterie         0
destockage_batterie       0
tco_6f                    0
consommation_isol         0
tch_6f                    0
tch_consommation          0
tch_consommation_isol     0
reference                 0
mois_nom                  0
date                      0
vent_10m_vitesse_moyen    12
temperature_C             12
temperature_ressentie     12
dtype: int64
```

In [20]:
```
df_conso_temp_na = df_conso_temp[df_conso_temp.isna().any(axis=1)]
```

In [21]:
```
# Les valeurs manquantes correspondent aux relevés du 2022-12-31 23:30:00; elles peuvent être supprimmées sans nuire à l'étude
df_conso_temp.dropna(inplace = True)
df_conso_temp.isna().sum()
```

Out[21]:
```
region                   0
date_heure               0
annee                    0
annee_mois               0
mois                     0
semaine                  0
jour                     0
jour_nom                 0
jour_numero              0
heure                    0
jour_fractionnel         0
consommation             0
thermique                0
nucleaire                0
eolien                   0
solaire                  0
hydraulique              0
bioenergies              0
pompage                  0
ech_physiques            0
export                   0
import                   0
stockage_batterie        0
destockage_batterie      0
tco_6f                   0
consommation_isol        0
tch_6f                   0
tch_consommation         0
tch_consommation_isol    0
reference                0
mois_nom                 0
date                     0
vent_10m_vitesse_moyen   0
temperature_C            0
temperature_ressentie    0
dtype: int64
```

In [22]:
```
# Liste des colonnes de production pour lesquelles effectuer les calculs
colonnes_production = df_eco_2_cumuls_delta.columns[3:16]
```

In [23]:
```
# Affichage des informations pour vérification
df_conso_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2103528 entries, 0 to 2103527
Data columns (total 35 columns):
 #   Column                  Dtype
---  ------                  -----
 0   region                  object
 1   date_heure              datetime64[ns]
 2   annee                   int64
 3   annee_mois              object
 4   mois                    int64
 5   semaine                 int64
 6   jour                    int64
 7   jour_nom                object
 8   jour_numero             int64
 9   heure                   object
 10  jour_fractionnel        float64
 11  consommation            float64
 12  thermique               float64
 13  nucleaire               float64
 14  eolien                  float64
 15  solaire                 float64
 16  hydraulique             float64
 17  bioenergies             float64
 18  pompage                 float64
 19  ech_physiques           float64
 20  export                  float64
 21  import                  float64
 22  stockage_batterie       float64
 23  destockage_batterie     float64
 24  tco_6f                  float64
 25  consommation_isol       float64
 26  tch_6f                  float64
 27  tch_consommation        float64
 28  tch_consommation_isol   float64
 29  reference               object
 30  mois_nom                object
 31  date                    object
 32  vent_10m_vitesse_moyen  float64
 33  temperature_C           float64
 34  temperature_ressentie   float64
dtypes: datetime64[ns](1), float64(22), int64(5), object(7)
memory usage: 577.8+ MB
```

In [24]:
```
# Suppression des lignes où les températures et vitesse du vent sont des NaN
df_conso_temp = df_conso_temp.dropna(subset=['vent_10m_vitesse_moyen', 'temperature_C', 'temperature_ressentie'])
```

In [25]:
```
# ==================== Ajout des colonnes 'delta_consommation' et 'delta_TMoy'
```

In [26]:
```
# Calcul de la différence de consommation cumulée par rapport à la ligne précédente pour chaque région
df_conso_temp['delta_consommation'] = df_conso_temp.groupby('region')['consommation'].diff()
```

```python
# Calcul de la différence de température réelle moyenne par rapport à la ligne précédente pour chaque région
df_conso_temp['delta_TMoy'] = df_conso_temp.groupby('region')['temperature_C'].diff()

# Vérification des résultats finaux
df_conso_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2103528 entries, 0 to 2103527
Data columns (total 37 columns):
 #   Column                  Dtype
---  ------                  -----
 0   region                  object
 1   date_heure              datetime64[ns]
 2   annee                   int64
 3   annee_mois              object
 4   mois                    int64
 5   semaine                 int64
 6   jour                    int64
 7   jour_nom                object
 8   jour_numero             int64
 9   heure                   object
 10  jour_fractionnel        float64
 11  consommation            float64
 12  thermique               float64
 13  nucleaire               float64
 14  eolien                  float64
 15  solaire                 float64
 16  hydraulique             float64
 17  bioenergies             float64
 18  pompage                 float64
 19  ech_physiques           float64
 20  export                  float64
 21  import                  float64
 22  stockage_batterie       float64
 23  destockage_batterie     float64
 24  tco_6f                  float64
 25  consommation_isol       float64
 26  tch_6f                  float64
 27  tch_consommation        float64
 28  tch_consommation_isol   float64
 29  reference               object
 30  mois_nom                object
 31  date                    object
 32  vent_10m_vitesse_moyen  float64
 33  temperature_C           float64
 34  temperature_ressentie   float64
 35  delta_consommation      float64
 36  delta_TMoy              float64
dtypes: datetime64[ns](1), float64(24), int64(5), object(7)
memory usage: 609.8+ MB
```

In [27]:
```
# Ajout d'une colonne des variations de consommations par variation de la température réelle moyenne
# =====> Sensibilité de Consommation Électrique à la Température (SCET)
```

In [28]:
```
df_conso_temp['SCET'] = np.where(
    df_conso_temp['delta_TMoy'] == 0, np.nan,
    df_conso_temp['delta_consommation'] / df_conso_temp['delta_TMoy'] * 1_000_000)
# Remplacement des NaN par les 2 voisins les plus proches
df_conso_temp['SCET'] = df_conso_temp['SCET'].interpolate()
```

In [29]:
```
# Ajout d'une colonne de l'élasticité de la consommations électriques par rapport  aux variation de la température réelle moyenne
# =====> Elasticité Consommation Électrique/ Température (Elasticite_C_T)
```

In [30]:
```
df_conso_temp['Elasticite_C_T'] = df_conso_temp['delta_consommation']*df_conso_temp['temperature_C'] /df_conso_temp['consommation'].replace(0, np.nan)
# Remplacement des NaN par les 2 voisins les plus proches
df_conso_temp['Elasticite_C_T'] = df_conso_temp['Elasticite_C_T'].interpolate()
df_conso_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2103528 entries, 0 to 2103527
Data columns (total 39 columns):
 #   Column                  Dtype
---  ------                  -----
 0   region                  object
 1   date_heure              datetime64[ns]
 2   annee                   int64
 3   annee_mois              object
 4   mois                    int64
 5   semaine                 int64
 6   jour                    int64
 7   jour_nom                object
 8   jour_numero             int64
 9   heure                   object
 10  jour_fractionnel        float64
 11  consommation            float64
 12  thermique               float64
 13  nucleaire               float64
 14  eolien                  float64
 15  solaire                 float64
 16  hydraulique             float64
 17  bioenergies             float64
 18  pompage                 float64
 19  ech_physiques           float64
 20  export                  float64
 21  import                  float64
 22  stockage_batterie       float64
 23  destockage_batterie     float64
 24  tco_6f                  float64
 25  consommation_isol       float64
 26  tch_6f                  float64
 27  tch_consommation        float64
 28  tch_consommation_isol   float64
 29  reference               object
 30  mois_nom                object
 31  date                    object
 32  vent_10m_vitesse_moyen  float64
 33  temperature_C           float64
 34  temperature_ressentie   float64
 35  delta_consommation      float64
 36  delta_TMoy              float64
 37  SCET                    float64
 38  Elasticite_C_T          float64
dtypes: datetime64[ns](1), float64(26), int64(5), object(7)
memory usage: 641.9+ MB
```

In [31]:
```
# AJOUT DES POPULATIONS
```

In [32]:
```
df_population = pd.read_csv("recensement.csv", sep=';', encoding='latin-1')
```

```
df_population.head()
```

Out[32]:

	region	2023	2022	2021	2020	2019	2018	2017	2016	2015	2014	2013
0	AUVERGNE RHONE ALPES	8197325	8155762	8114361	8078652	8042936	7994459	7948287	7916889	7877698	7820966	7757595
1	BOURGOGNE FRANCHE COMTE	2786296	2797366	2800194	2801695	2805580	2807807	2811423	2818338	2820940	2820623	2819783
2	BRETAGNE	3429882	3414637	3394567	3373835	3354854	3335414	3318904	3306529	3293850	3276543	3258707
3	CENTRE VAL DE LOIRE	2572278	2572916	2573303	2574863	2573180	2572853	2576252	2577866	2578592	2577435	2570548
4	GRAND EST	5562262	5563889	5561287	5562651	5556219	5550389	5549586	5555186	5559051	5554645	5552388

In [33]:
```
# Transformation de 'df_population' pour le passer en format long (melt)
df_population_long = df_population.melt(
    id_vars=['region'],
    var_name='annee',
    value_name='population'
)
```

In [34]:
```
# On s'assure que 'annee' dans les deux DataFrames est de type int pour la jointure
df_population_long['annee'] = df_population_long['annee'].astype(int)
df_conso_temp['annee'] = df_conso_temp['annee'].astype(int)

# Jointure sur 'region' et 'annee'
df_conso_temp = pd.merge(df_conso_temp, df_population_long, on=['region', 'annee'], how='left')
```

In [35]:
```
# Ajout d'une colonne de la SCETH par habitant
# =====> Sensibilité de Consommation Électrique à la Température (SCETH)
```

In [36]:
```
# Calcul de SCETH avec gestion des zéros
df_conso_temp['SCETH'] = np.where((df_conso_temp['delta_TMoy'] == 0) | (df_conso_temp['population'] == 0),np.nan,
    df_conso_temp['delta_consommation'] / df_conso_temp['delta_TMoy'] / df_conso_temp['population'] * 1_000_000)
```

In [37]:
```
# Remplacement des NaN par les 2 voisins les plus proches
df_conso_temp['SCETH'] = df_conso_temp['SCETH'].interpolate()
```

In [38]:
```
# Reccherche de valeurs manquantes
df_conso_temp.isna().sum()
```

Out[38]:
```
region                     0
date_heure                 0
annee                      0
annee_mois                 0
mois                       0
semaine                    0
jour                       0
jour_nom                   0
jour_numero                0
heure                      0
jour_fractionnel           0
consommation               0
thermique                  0
nucleaire                  0
eolien                     0
solaire                    0
hydraulique                0
bioenergies                0
pompage                    0
ech_physiques              0
export                     0
import                     0
stockage_batterie          0
destockage_batterie        0
tco_6f                     0
consommation_isol          0
tch_6f                     0
tch_consommation           0
tch_consommation_isol      0
reference                  0
mois_nom                   0
date                       0
vent_10m_vitesse_moyen     0
temperature_C              0
temperature_ressentie      0
delta_consommation        12
delta_TMoy                12
SCET                      24
Elasticite_C_T            12
population                 0
SCETH                     24
dtype: int64
```

In [39]:
```
# Suppression des lignes où les températures sont des NaN (extrémités des données)
df_conso_temp = df_conso_temp.dropna(subset=['SCET','Elasticite_C_T', 'SCETH'])
df_conso_temp.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2103504 entries, 24 to 2103527
Data columns (total 41 columns):
 #   Column                  Dtype
---  ------                  -----
 0   region                  object
 1   date_heure              datetime64[ns]
 2   annee                   int32
 3   annee_mois              object
 4   mois                    int64
 5   semaine                 int64
 6   jour                    int64
 7   jour_nom                object
 8   jour_numero             int64
 9   heure                   object
 10  jour_fractionnel        float64
 11  consommation            float64
 12  thermique               float64
 13  nucleaire               float64
 14  eolien                  float64
 15  solaire                 float64
 16  hydraulique             float64
 17  bioenergies             float64
 18  pompage                 float64
 19  ech_physiques           float64
 20  export                  float64
 21  import                  float64
 22  stockage_batterie       float64
 23  destockage_batterie     float64
 24  tco_6f                  float64
 25  consommation_isol       float64
 26  tch_6f                  float64
 27  tch_consommation        float64
 28  tch_consommation_isol   float64
 29  reference               object
 30  mois_nom                object
 31  date                    object
 32  vent_10m_vitesse_moyen  float64
 33  temperature_C           float64
 34  temperature_ressentie   float64
 35  delta_consommation      float64
 36  delta_TMoy              float64
 37  SCET                    float64
 38  Elasticite_C_T          float64
 39  population              int64
 40  SCETH                   float64
dtypes: datetime64[ns](1), float64(27), int32(1), int64(5), object(7)
memory usage: 666.0+ MB
```

In [40]:
```
df_conso_temp.isna().sum()
```

Out[40]:
```
region                   0
date_heure               0
annee                    0
annee_mois               0
mois                     0
semaine                  0
jour                     0
jour_nom                 0
jour_numero              0
heure                    0
jour_fractionnel         0
consommation             0
thermique                0
nucleaire                0
eolien                   0
solaire                  0
hydraulique              0
bioenergies              0
pompage                  0
ech_physiques            0
export                   0
import                   0
stockage_batterie        0
destockage_batterie      0
tco_6f                   0
consommation_isol        0
tch_6f                   0
tch_consommation         0
tch_consommation_isol    0
reference                0
mois_nom                 0
date                     0
vent_10m_vitesse_moyen   0
temperature_C            0
temperature_ressentie    0
delta_consommation       0
delta_TMoy               0
SCET                     0
Elasticite_C_T           0
population               0
SCETH                    0
dtype: int64
```

In [41]:
```
# AJOUT DES DONNEES SUR LA VITESSE DU VENT A 100M ET DES RAYONNEMENTS SOLAIRES
```

In [42]:
```
df_vent_100m_solaire_30min = pd.read_csv("df_vent_100m_solaire_30min.csv", sep=';', encoding='latin-1')
df_vent_100m_solaire_30min.head()
```

Out[42]:

	date_heure	region	vitesse_vent_100m_m_par_s	rayonnement_solaire_global_W_par_m2
0	2013-01-01 00:00:00	AUVERGNE RHONE ALPES	0.0	0.0
1	2013-01-01 00:30:00	AUVERGNE RHONE ALPES	0.0	0.0
2	2013-01-01 01:00:00	AUVERGNE RHONE ALPES	0.0	0.0
3	2013-01-01 01:30:00	AUVERGNE RHONE ALPES	0.0	0.0
4	2013-01-01 02:00:00	AUVERGNE RHONE ALPES	0.0	0.0

In [43]:
```
# Conversion des dates
df_vent_100m_solaire_30min['date_heure'] = pd.to_datetime(df_vent_100m_solaire_30min['date_heure'])
```

In [55]:
```
# Fusion
df_conso_meteo = pd.merge(
    df_conso_temp,
    df_vent_100m_solaire_30min[['date_heure','region','vitesse_vent_100m_m_par_s', 'rayonnement_solaire_global_W_par_m2']],
    on=['region', 'date_heure'],
    how='left')
df_conso_meteo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2103996 entries, 0 to 2103995
Data columns (total 43 columns):
 #   Column                              Dtype
---  ------                              -----
 0   region                              object
 1   date_heure                          datetime64[ns]
 2   annee                               int32
 3   annee_mois                          object
 4   mois                                int64
 5   semaine                             int64
 6   jour                                int64
 7   jour_nom                            object
 8   jour_numero                         int64
 9   heure                               object
 10  jour_fractionnel                    float64
 11  consommation                        float64
 12  thermique                           float64
 13  nucleaire                           float64
 14  eolien                              float64
 15  solaire                             float64
 16  hydraulique                         float64
 17  bioenergies                         float64
 18  pompage                             float64
 19  ech_physiques                       float64
 20  export                              float64
 21  import                              float64
 22  stockage_batterie                   float64
 23  destockage_batterie                 float64
 24  tco_6f                              float64
 25  consommation_isol                   float64
 26  tch_6f                              float64
 27  tch_consommation                    float64
 28  tch_consommation_isol               float64
 29  reference                           object
 30  mois_nom                            object
 31  date                                object
 32  vent_10m_vitesse_moyen              float64
 33  temperature_C                       float64
 34  temperature_ressentie               float64
 35  delta_consommation                  float64
 36  delta_TMoy                          float64
 37  SCET                                float64
 38  Elasticite_C_T                      float64
 39  population                          int64
 40  SCETH                               float64
 41  vitesse_vent_100m_m_par_s           float64
 42  rayonnement_solaire_global_W_par_m2  float64
dtypes: datetime64[ns](1), float64(29), int32(1), int64(5), object(7)
memory usage: 682.2+ MB
```

In [56]:
```
df_conso_meteo.isna().sum()
```

Out[56]:
```
region                                0
date_heure                            0
annee                                 0
annee_mois                            0
mois                                  0
semaine                               0
jour                                  0
jour_nom                              0
jour_numero                           0
heure                                 0
jour_fractionnel                      0
consommation                          0
thermique                             0
nucleaire                             0
eolien                                0
solaire                               0
hydraulique                           0
bioenergies                           0
pompage                               0
ech_physiques                         0
export                                0
import                                0
stockage_batterie                     0
destockage_batterie                   0
tco_6f                                0
consommation_isol                     0
tch_6f                                0
tch_consommation                      0
tch_consommation_isol                 0
reference                             0
mois_nom                              0
date                                  0
vent_10m_vitesse_moyen                0
temperature_C                         0
temperature_ressentie                 0
delta_consommation                    0
delta_TMoy                            0
SCET                                  0
Elasticite_C_T                        0
population                            0
SCETH                                 0
vitesse_vent_100m_m_par_s             0
rayonnement_solaire_global_W_par_m2   0
dtype: int64
```

In [57]:
```
# AJOUT D'UNE COLONNE DU CUBE DE LA VITESSE DU VENT ET CALCUL DU COEFFICIENT EOLIEN REGIONAL POUR ANALYSE ULTERIEURE DE L'ENERGIE EOLIENNE
```

In [58]:
```
df_conso_meteo['vitesse_vent_100m_cube'] = df_conso_meteo['vitesse_vent_100m_m_par_s'] ** 3

df_conso_meteo['K_eolien_regional'] = np.where(
    df_conso_meteo['vitesse_vent_100m_cube'] != 0,
    (df_conso_meteo['eolien'] / df_conso_meteo['vitesse_vent_100m_cube']) * 1_000_000, 0)
```

In [59]:
```
# AJOUT D'UNE COLONNE DE CALCUL DU COEFFICIENT SOLAIRE REGIONAL POUR ANALYSE ULTERIEURE DE L'ENERGIE SOLAIRE
```

In [60]:
```
df_conso_meteo['K_solaire_regional'] = np.where(
    df_conso_meteo['rayonnement_solaire_global_W_par_m2'] != 0,
    (df_conso_meteo['solaire'] / df_conso_meteo['rayonnement_solaire_global_W_par_m2']) * 1_000_000,0)
```

In [61]:
```
print(df_conso_meteo.columns)
```

```
Index(['region', 'date_heure', 'annee', 'annee_mois', 'mois', 'semaine',
       'jour', 'jour_nom', 'jour_numero', 'heure', 'jour_fractionnel',
       'consommation', 'thermique', 'nucleaire', 'eolien', 'solaire',
       'hydraulique', 'bioenergies', 'pompage', 'ech_physiques', 'export',
       'import', 'stockage_batterie', 'destockage_batterie', 'tco_6f',
       'consommation_isol', 'tch_6f', 'tch_consommation',
       'tch_consommation_isol', 'reference', 'mois_nom', 'date',
       'vent_10m_vitesse_moyen', 'temperature_C', 'temperature_ressentie',
       'delta_consommation', 'delta_TMoy', 'SCET', 'Elasticite_C_T',
       'population', 'SCETH', 'vitesse_vent_100m_m_par_s',
       'rayonnement_solaire_global_W_par_m2', 'vitesse_vent_100m_cube',
       'K_eolien_regional', 'K_solaire_regional'],
      dtype='object')
```

In [62]:
```
# Organisation des colonnes

ordre_desire =['reference','region', 'population','date_heure', 'date','annee', 'annee_mois', 'mois', 'mois_nom',
       'semaine', 'jour', 'jour_nom', 'jour_numero', 'heure',
       'jour_fractionnel','consommation', 'thermique',
       'nucleaire', 'eolien', 'solaire', 'hydraulique', 'bioenergies',
       'pompage', 'ech_physiques', 'export', 'import', 'stockage_batterie',
       'destockage_batterie', 'tco_6f', 'consommation_isol', 'tch_6f',
       'tch_consommation', 'tch_consommation_isol', 'delta_consommation','temperature_C', 'temperature_ressentie', 'delta_TMoy', 'SCET',
      'Elasticite_C_T', 'SCETH',
       'vent_10m_vitesse_moyen', 'vitesse_vent_100m_m_par_s','vitesse_vent_100m_cube','K_eolien_regional',
       'rayonnement_solaire_global_W_par_m2','K_solaire_regional']
df_conso_meteo = df_conso_meteo[ordre_desire]
df_conso_meteo.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2103996 entries, 0 to 2103995
Data columns (total 46 columns):
 #   Column                                Dtype
---  ------                                -----
 0   reference                             object
 1   region                                object
 2   population                            int64
 3   date_heure                            datetime64[ns]
 4   date                                  object
 5   annee                                 int32
 6   annee_mois                            object
 7   mois                                  int64
 8   mois_nom                              object
 9   semaine                               int64
 10  jour                                  int64
 11  jour_nom                              object
 12  jour_numero                           int64
 13  heure                                 object
 14  jour_fractionnel                      float64
 15  consommation                          float64
 16  thermique                             float64
 17  nucleaire                             float64
 18  eolien                                float64
 19  solaire                               float64
 20  hydraulique                           float64
 21  bioenergies                           float64
 22  pompage                               float64
 23  ech_physiques                         float64
 24  export                                float64
 25  import                                float64
 26  stockage_batterie                     float64
 27  destockage_batterie                   float64
 28  tco_6f                                float64
 29  consommation_isol                     float64
 30  tch_6f                                float64
 31  tch_consommation                      float64
 32  tch_consommation_isol                 float64
 33  delta_consommation                    float64
 34  temperature_C                         float64
 35  temperature_ressentie                 float64
 36  delta_TMoy                            float64
 37  SCET                                  float64
 38  Elasticite_C_T                        float64
 39  SCETH                                 float64
 40  vent_10m_vitesse_moyen                float64
 41  vitesse_vent_100m_m_par_s             float64
 42  vitesse_vent_100m_cube                float64
 43  K_eolien_regional                     float64
 44  rayonnement_solaire_global_W_par_m2   float64
 45  K_solaire_regional                    float64
```

```
dtypes: datetime64[ns](1), float64(32), int32(1), int64(5), object(7)
memory usage: 730.4+ MB
```

In [63]:
```
df_conso_meteo.isna().sum()
```

Out[63]:
```
reference                           0
region                              0
population                          0
date_heure                          0
date                                0
annee                               0
annee_mois                          0
mois                                0
mois_nom                            0
semaine                             0
jour                                0
jour_nom                            0
jour_numero                         0
heure                               0
jour_fractionnel                    0
consommation                        0
thermique                           0
nucleaire                           0
eolien                              0
solaire                             0
hydraulique                         0
bioenergies                         0
pompage                             0
ech_physiques                       0
export                              0
import                              0
stockage_batterie                   0
destockage_batterie                 0
tco_6f                              0
consommation_isol                   0
tch_6f                              0
tch_consommation                    0
tch_consommation_isol               0
delta_consommation                  0
temperature_C                       0
temperature_ressentie               0
delta_TMoy                          0
SCET                                0
Elasticite_C_T                      0
SCETH                               0
vent_10m_vitesse_moyen              0
vitesse_vent_100m_m_par_s           0
vitesse_vent_100m_cube              0
K_eolien_regional                   0
rayonnement_solaire_global_W_par_m2 0
K_solaire_regional                  0
dtype: int64
```

In [64]:
```
# Sauvegarde des résultats
df_conso_meteo.to_csv("df_conso_meteo.csv", sep=';',index=False)
```