

```
In [1]: """
Created on Wed Feb  5 18:27:48 2025
@author: Thierry ALLEM
"""
```

```
Out[1]: '\nCreated on Wed Feb  5 18:27:48 2025\n@author: Thierry ALLEM\n'
```

```
In [2]: # TRAITEMENT DES DONNEES SUR LE RAYONNEMENT SOLAIRE ET LA VITESSE DU VENTS A 100M (pour l'étude sur La production des éoliennes)
# Approximations sur la base d'une base de données de relevés tri-horaires
```

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

```
In [4]: # Importation du fichier des relevés de températures
# Lecture du fichier CSV
df_vent_100m_solaire_3h = pd.read_csv("solaire_vent_regions.csv", sep=';', encoding='utf-8')
df_vent_100m_solaire_3h.head()
```

```
Out[4]:
```

	date_heure	Code INSEE région	region	Vitesse du vent à 100m (m/s)	Rayonnement solaire global (W/m2)
0	2016-10-11T05:00:00+02:00	84	Auvergne-Rhône-Alpes	4.10	0.00
1	2016-10-11T05:00:00+02:00	93	Provence-Alpes-Côte d'Azur	7.56	0.00
2	2016-10-11T08:00:00+02:00	27	Bourgogne-Franche-Comté	4.52	0.11
3	2016-10-11T08:00:00+02:00	84	Auvergne-Rhône-Alpes	4.05	0.07
4	2016-10-11T08:00:00+02:00	76	Occitanie	7.42	0.00

```
In [5]: df_vent_100m_solaire_3h.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 339351 entries, 0 to 339350
Data columns (total 5 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   date_heure                            339351 non-null object
1   Code INSEE région                    339351 non-null int64
2   region                               339351 non-null object
3   Vitesse du vent à 100m (m/s)         339351 non-null float64
4   Rayonnement solaire global (W/m2)    339351 non-null float64
dtypes: float64(2), int64(1), object(2)
memory usage: 12.9+ MB
```

```
In [6]: # Conversion de la colonne 'date_heure' en datetime
df_vent_100m_solaire_3h['date_heure'] = pd.to_datetime(df_vent_100m_solaire_3h['date_heure'], errors='coerce', utc=True)
```

```
df_vent_100m_solaire_3h["date_heure"] = df_vent_100m_solaire_3h["date_heure"].dt.tz_localize(None)

# Extraction de L'année
df_vent_100m_solaire_3h['annee'] = df_vent_100m_solaire_3h['date_heure'].dt.year

# Ajout d'une colonne 'date' avec Le format 'yyyy-mm-dd'
df_vent_100m_solaire_3h['date'] = df_vent_100m_solaire_3h['date_heure'].dt.strftime('%Y-%m-%d')

# Ajout d'une colonne 'heure' à L'heure française
df_vent_100m_solaire_3h['heure'] = df_vent_100m_solaire_3h['date_heure'].dt.strftime('%H:%M')
df_vent_100m_solaire_3h.info()
df_vent_100m_solaire_3h['annee'].unique()
df_vent_100m_solaire_3h['heure'].unique()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 339351 entries, 0 to 339350
Data columns (total 8 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   date_heure                            339351 non-null  datetime64[ns]
1   Code INSEE région                    339351 non-null  int64
2   region                               339351 non-null  object
3   Vitesse du vent à 100m (m/s)         339351 non-null  float64
4   Rayonnement solaire global (W/m2)    339351 non-null  float64
5   annee                                339351 non-null  int32
6   date                                  339351 non-null  object
7   heure                                 339351 non-null  object
dtypes: datetime64[ns](1), float64(2), int32(1), int64(1), object(3)
memory usage: 19.4+ MB
```

```
Out[6]: array(['03:00', '06:00', '09:00', '12:00', '18:00', '21:00', '00:00',
              '15:00'], dtype=object)
```

```
In [7]: # Filtrage des années pour limiter le dataset de 2013 à 2022 inclus
df_vent_100m_solaire_3h = df_vent_100m_solaire_3h[(df_vent_100m_solaire_3h["annee"] >= 2013) & (df_vent_100m_solaire_3h["annee"] <= 2022)]
```

```
In [8]: # Liste des régions
df_vent_100m_solaire_3h['region'].unique()
```

```
Out[8]: array(['Auvergne-Rhône-Alpes', 'Provence-Alpes-Côte d'Azur',
              'Bourgogne-Franche-Comté', 'Occitanie', 'Grand Est',
              'Centre-Val de Loire', 'Île-de-France', 'Nouvelle-Aquitaine',
              'Normandie', 'Hauts-de-France', 'Pays de la Loire', 'Corse',
              'Bretagne'], dtype=object)
```

```
In [9]: # Suppression des lignes de régions non concernées par cette étude
regions_a_exclure = [np.nan, 'Corse']
df_vent_100m_solaire_3h = df_vent_100m_solaire_3h[~df_vent_100m_solaire_3h['region'].isin(regions_a_exclure)]
df_vent_100m_solaire_3h['region'].unique()
```

```
Out[9]: array(['Auvergne-Rhône-Alpes', 'Provence-Alpes-Côte d'Azur',  
             'Bourgogne-Franche-Comté', 'Occitanie', 'Grand Est',  
             'Centre-Val de Loire', 'Île-de-France', 'Nouvelle-Aquitaine',  
             'Normandie', 'Hauts-de-France', 'Pays de la Loire', 'Bretagne'],  
            dtype=object)
```

```
In [10]: # Renommage des valeurs de 'region'  
renommage_regions = {  
    'Occitanie': 'OCCITANIE',  
    'Normandie': 'NORMANDIE',  
    'Bretagne': 'BRETAGNE',  
    'Provence-Alpes-Côte d'Azur': 'PROVENCE ALPES COTE D AZUR',  
    'Hauts-de-France': 'HAUTS DE FRANCE',  
    'Île-de-France': 'ILE DE FRANCE',  
    'Grand Est': 'GRAND EST',  
    'Nouvelle-Aquitaine': 'NOUVELLE AQUITAINE',  
    'Bourgogne-Franche-Comté': 'BOURGOGNE FRANCHE COMTE',  
    'Pays de la Loire': 'PAYS DE LA LOIRE',  
    'Auvergne-Rhône-Alpes': 'AUVERGNE RHONE ALPES',  
    'Centre-Val de Loire': 'CENTRE VAL DE LOIRE'  
}  
df_vent_100m_solaire_3h['region'] = df_vent_100m_solaire_3h['region'].replace(renommage_regions)  
df_vent_100m_solaire_3h['region'].unique()
```

```
Out[10]: array(['AUVERGNE RHONE ALPES', 'PROVENCE ALPES COTE D AZUR',  
              'BOURGOGNE FRANCHE COMTE', 'OCCITANIE', 'GRAND EST',  
              'CENTRE VAL DE LOIRE', 'ILE DE FRANCE', 'NOUVELLE AQUITAINE',  
              'NORMANDIE', 'HAUTS DE FRANCE', 'PAYS DE LA LOIRE', 'BRETAGNE'],  
             dtype=object)
```

```
In [11]: # INTERPOLATION DES DONNEES SUR LA PERIODE 2013 -2022 PAR PAS DE 30 MINUTES  
  
# Création de la plage temporelle de 2013-01-01 à 2022-12-31 avec un pas de 30 minutes  
date_range = pd.date_range("2013-01-01 00:00", "2022-12-31 23:30", freq="30min")  
  
# Liste des régions  
regions = df_vent_100m_solaire_3h['region'].unique()  
  
# Construction du DataFrame initial  
df_base = pd.DataFrame({"date_heure": np.tile(date_range, len(regions)),  
                        "region": np.repeat(regions, len(date_range))})  
  
# Fusion des données initiales  
merged_df = pd.merge(df_base, df_vent_100m_solaire_3h,  
                     on=["date_heure", "region"],  
                     how="left")  
  
# Traitement des années 2013-2015  
merged_df['annee'] = merged_df['date_heure'].dt.year  
mask_zeros = merged_df['annee'].isin([2013, 2014, 2015])  
merged_df.loc[mask_zeros, ['Vitesse du vent à 100m (m/s)', 'Rayonnement solaire global (W/m2)']] = 0.0
```

```
# Interpolation des valeurs par région
merged_df.sort_values(by=['region', 'date_heure'], inplace=True)
merged_df[['Vitesse du vent à 100m (m/s)', 'Rayonnement solaire global (W/m2)']] = (
    merged_df.groupby('region', group_keys=False)[['Vitesse du vent à 100m (m/s)', 'Rayonnement solaire global (W/m2)']]
    .apply(lambda group: group.interpolate(method='linear'))
)
```

```
In [12]: # Nettoyage des colonnes inutiles
df_vent_100m_solaire_30min = merged_df[['date_heure', 'region', 'Vitesse du vent à 100m (m/s)', 'Rayonnement solaire global (W/m2)']]
df_vent_100m_solaire_30min.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2104044 entries, 0 to 350673
Data columns (total 4 columns):
#   Column                                Dtype
---  -
0   date_heure                           datetime64[ns]
1   region                                object
2   Vitesse du vent à 100m (m/s)         float64
3   Rayonnement solaire global (W/m2)    float64
dtypes: datetime64[ns](1), float64(2), object(1)
memory usage: 80.3+ MB
```

```
In [13]: # Renommage de noms de colonnes

colonnes_a_renommer = {'Vitesse du vent à 100m (m/s)': 'vitesse_vent_100m_m_par_s',
                       'Rayonnement solaire global (W/m2)': 'rayonnement_solaire_global_W_par_m2'}
df_vent_100m_solaire_30min = df_vent_100m_solaire_30min.rename(columns=colonnes_a_renommer)
df_vent_100m_solaire_30min.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2104044 entries, 0 to 350673
Data columns (total 4 columns):
#   Column                                Dtype
---  -
0   date_heure                           datetime64[ns]
1   region                                object
2   vitesse_vent_100m_m_par_s           float64
3   rayonnement_solaire_global_W_par_m2 float64
dtypes: datetime64[ns](1), float64(2), object(1)
memory usage: 80.3+ MB
```

```
In [14]: df_vent_100m_solaire_30min.head(20)
```

Out[14]:

	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
5	2013-01-01 02:30:00	AUVERGNE RHONE ALPES	0.0	0.0
6	2013-01-01 03:00:00	AUVERGNE RHONE ALPES	0.0	0.0
7	2013-01-01 03:30:00	AUVERGNE RHONE ALPES	0.0	0.0
8	2013-01-01 04:00:00	AUVERGNE RHONE ALPES	0.0	0.0
9	2013-01-01 04:30:00	AUVERGNE RHONE ALPES	0.0	0.0
10	2013-01-01 05:00:00	AUVERGNE RHONE ALPES	0.0	0.0
11	2013-01-01 05:30:00	AUVERGNE RHONE ALPES	0.0	0.0
12	2013-01-01 06:00:00	AUVERGNE RHONE ALPES	0.0	0.0
13	2013-01-01 06:30:00	AUVERGNE RHONE ALPES	0.0	0.0
14	2013-01-01 07:00:00	AUVERGNE RHONE ALPES	0.0	0.0
15	2013-01-01 07:30:00	AUVERGNE RHONE ALPES	0.0	0.0
16	2013-01-01 08:00:00	AUVERGNE RHONE ALPES	0.0	0.0
17	2013-01-01 08:30:00	AUVERGNE RHONE ALPES	0.0	0.0
18	2013-01-01 09:00:00	AUVERGNE RHONE ALPES	0.0	0.0
19	2013-01-01 09:30:00	AUVERGNE RHONE ALPES	0.0	0.0

In [15]:

```
# Tri et sauvegarde des résultats
df_vent_100m_solaire_30min = df_vent_100m_solaire_30min.sort_values(['region', 'date_heure'])
df_vent_100m_solaire_30min.to_csv("df_vent_100m_solaire_30min.csv", sep=';', index=False)
```