

Projet ISSD 2024

February 2, 2025

1 Projet ISSD 2024 - Houel Nathan

Au cours de ces années le climat est un sujet qui est de plus en plus cité dans l'actualité dû au réchauffement climatique. Le climat est un phénomène très important pour l'Homme car il conditionne sa survie que ce soit au niveau de la température, des tempêtes ou encore d'autres phénomènes. C'est pour cela qu'il est donc très important de s'en intéresser et de l'étudier.

J'ai donc choisi de travailler sur la **base des données météorologiques** pour pouvoir étudier une partie du climat en France. En effet, je compte m'intéresser aux **tempêtes** car il s'agit d'un phénomène très important et qui crée souvent des désastres. Ma problématique est donc la suivante :

1.0.1 Problématique : Quelles sont les régions en France métropolitaine qui sont le plus touchées par des tempêtes ?

Afin de répondre à cette problématique je vais chercher globalement à **filtrer** parmi toutes mes données, celles qui correspondent à des mesures liées à une tempête. Puis, je vais ensuite regarder quelles régions sont le plus touchées par des tempêtes pour pouvoir réaliser un **Top 5** des régions. Pour finir, je souhaite développer l'étude en réalisant une **prédiction** sur les 6 prochaines années du nombre de tempête qui peut potentiellement se réaliser pour chaque région.

2 1. Analyse et présentation des données

Importation de l'ensemble des librairies nécessaires :

```
[1]: import numpy as np
import csv
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
sns.set(color_codes = True)
sns.set(font_scale=1.5)
from scipy import stats
from IPython.display import Image
from sklearn.linear_model import LinearRegression
```

Chargement du jeu de donnée original :

```
[2]: DATA_original = pd.read_csv('data/data_meteo2.csv', delimiter = ';')
print('Nombre de mesure dans la base :' + str(len(DATA_original)))
DATA_original.head()
```

Nombre de mesure dans la base :976597

```
[2]: Unnamed: 0      Date  Pression au niveau mer \
0      2  2010-01-05T13:00:00+01:00      100720.0
1      3  2010-01-05T13:00:00+01:00      100750.0
2      6  2010-01-05T13:00:00+01:00      101630.0
3      7  2010-01-05T16:00:00+01:00      100480.0
4      8  2010-01-05T16:00:00+01:00      100530.0

Variation de pression en 3 heures  Direction du vent moyen 10 mn \
0      -190.0      200.0
1      -230.0      210.0
2      180.0      30.0
3      -150.0      190.0
4      -210.0      10.0

Vitesse du vent moyen 10 mn  Température  Point de rosée  Humidité \
0      3.6      273.65      271.75      87.0
1      4.1      276.95      272.55      73.0
2      0.5      299.05      295.95      83.0
3      1.5      274.85      271.45      78.0
4      2.6      272.65      268.25      69.0

Visibilité horizontale ...  Température maximale sur 12 heures (°C) \
0      12000.0 ...      NaN
1      25000.0 ...      NaN
2      15000.0 ...      NaN
3      12000.0 ...      NaN
4      18000.0 ...      NaN

Température maximale sur 24 heures (°C) \
0      NaN
1      NaN
2      30.4
3      NaN
4      NaN

Température minimale du sol sur 12 heures (en °C)  Latitude  Longitude \
0      NaN  49.180000  -0.456167
1      NaN  48.444167  -4.412000
2      NaN  14.595333  -60.995667
3      NaN  50.570000   3.097500
4      NaN  48.716833   2.384333
```

	Altitude	department (name)	department (code)	region (name)	\
0	67	Calvados	14	Normandie	
1	94	Finistère	29	Bretagne	
2	3	Martinique	972	Martinique	
3	47	Nord	59	Hauts-de-France	
4	89	Essonne	91	Île-de-France	

	mois_de_l_annee
0	1
1	1
2	1
3	1
4	1

[5 rows x 53 columns]

Affichage de l'en-tête :

```
[3]: entete = list(DATA_original.columns)
print('Nombre de colonnes : ' + str(len(entete)))
print(entete)
```

Nombre de colonnes : 53

```
['Unnamed: 0', 'Date', 'Pression au niveau mer', 'Variation de pression en 3
heures', 'Direction du vent moyen 10 mn', 'Vitesse du vent moyen 10 mn',
'Température', 'Point de rosée', 'Humidité', 'Visibilité horizontale', 'Temps
présent', 'Temps passé 1', 'Temps passé 2', 'Nébulosité totale', 'Nébulosité
des nuages de l' étage inférieur', 'Hauteur de la base des nuages de l'étage
inférieur', 'Type des nuages de l'étage inférieur', 'Type des nuages de l'étage
moyen', 'Type des nuages de l'étage supérieur', 'Pression station', 'Niveau
barométrique', 'Variation de pression en 24 heures', 'Méthode de mesure
Température du thermomètre mouillé', 'Température du thermomètre mouillé',
'Rafale sur les 10 dernières minutes', 'Rafales sur une période', 'Periode de
mesure de la rafale', 'Etat du sol', 'Hauteur totale de la couche de neige,
glace, autre au sol', 'Hauteur de la neige fraîche', 'Periode de mesure de la
neige fraîche', 'Précipitations dans la dernière heure', 'Précipitations dans
les 3 dernières heures', 'Précipitations dans les 12 dernières heures',
'Précipitations dans les 24 dernières heures', 'Phénomène spécial 1', 'Phénomène
spécial 2', 'Phénomène spécial 3', 'Phénomène spécial 4', 'Nom', 'Température
(°C)', 'Température minimale sur 12 heures (°C)', 'Température minimale sur 24
heures (°C)', 'Température maximale sur 12 heures (°C)', 'Température maximale
sur 24 heures (°C)', 'Température minimale du sol sur 12 heures (en °C)',
'Latitude', 'Longitude', 'Altitude', 'department (name)', 'department (code)',
'region (name)', 'mois_de_l_annee']
```

Copie de la base pour pouvoir la modifier et la remettre en forme :

```
[4]: DATA = DATA_original.copy()
```

Affichage des différentes valeurs pour chaque colonnes :

```
[5]: mois = DATA.groupby(['mois_de_l_annee'])  
     mois.size()
```

```
[5]: mois_de_l_annee  
     1      88384  
     2      78170  
     3      81751  
     4      79074  
     5      81454  
     6      79185  
     7      81907  
     8      82497  
     9      80236  
    10      82087  
    11      79674  
    12      82178  
     dtype: int64
```

```
[6]: DATA["Température (°C)"].describe()
```

```
[6]: count      975597.000000  
     mean         15.618396  
     std          8.453559  
     min        -19.700000  
     25%          9.200000  
     50%         14.900000  
     75%         22.900000  
     max         67.700000  
     Name: Température (°C), dtype: float64
```

En réalisant cette manipulation un bon nombre de fois sur à peu près toutes les colonnes, j'ai pu **analyser et visualiser** les différentes types d'informations que j'avais dans ce jeu de données.

Ainsi, ce jeu de données représente des données météorologiques en **France métropolitaine** mais aussi celles de ses **DROM-COM**. Nous avons au total **976 597 mesures** et **53 données par mesure**. Nous retrouvons parmi ces données de nombreuses informations précieuses telles que la **date, la température, la pression, les précipitations** dans un certain intervalles de temps, etc..

En ce qui concerne la date, les données de ce jeu vont du **1er Janvier 2010** à 1h jusqu'au **19 Février 2024** à 19h. Entre chaque mesure nous avons un intervalle de temps de **3 heures**.

Pour la localisation, nous avons au total **14 régions** mélangeant des régions DROM-COM et des régions métropolitaines. Et parmi ces 14 régions, nous avons au total **25 stations**.

Cependant, on remarque que l'ordre des mesures n'est pas **chronologique**, il est donc compliqué de pouvoir mieux analyser les données dans ce cas-là. Je vais donc passer au pré-traitement des données pour pouvoir mieux les analyser.

3 2. Pré-traitement et nettoyage du jeu de données

L'objectif ici va être de **remettre en forme** l'ensemble du jeu de données, en remettant dans l'ordre chronologique les mesures, en **supprimant les doublons** mais aussi les colonnes qui ne nous intéressent pas pour répondre à la problématique. Une fois que toutes les données inutiles seront supprimées, je vais mettre en index du jeu de données la date pour que l'étude soit plus facile.

Modification afin d'afficher les mesures dans l'ordre chronologique :

```
[7]: df = pd.DataFrame(DATA)
df['Date'] = pd.to_datetime(df['Date'], utc=True)
df = df.sort_values(by='Date')
df.head(n=1000000)[["Date", "mois_de_l_annee"]]
```

```
[7]:
```

		Date	mois_de_l_annee
358652	2010-01-01 00:00:00+00:00		1
544620	2010-01-01 00:00:00+00:00		1
544619	2010-01-01 00:00:00+00:00		1
310595	2010-01-01 00:00:00+00:00		1
61487	2010-01-01 00:00:00+00:00		1
...
917658	2024-02-19 18:00:00+00:00		2
843837	2024-02-19 18:00:00+00:00		2
917657	2024-02-19 18:00:00+00:00		2
891851	2024-02-19 18:00:00+00:00		2
843840	2024-02-19 18:00:00+00:00		2

[976597 rows x 2 columns]

On remarque maintenant que les mesures sont affichées dans l'ordre chronologique.

Suppression des doublons :

```
[8]: df_clean = df.head(n=1000000).drop_duplicates()
print(df.head(n=1000000).Température.mean())
print(df_clean.Température.mean())
print(len(df))
print(len(df_clean))
df_clean.head()
```

288.7683957105239

288.7683957105239

976597

976597

```
[8]:
```

	Unnamed: 0	Date	Pression au niveau mer \
358652	877895	2010-01-01 00:00:00+00:00	99080.0
544620	1335100	2010-01-01 00:00:00+00:00	99210.0
544619	1335099	2010-01-01 00:00:00+00:00	99530.0

310595	759786	2010-01-01 00:00:00+00:00	99250.0
61487	150878	2010-01-01 00:00:00+00:00	99360.0

	Variation de pression en 3 heures	Direction du vent moyen 10 mn \
358652	-120.0	0.0
544620	20.0	240.0
544619	10.0	30.0
310595	-100.0	310.0
61487	90.0	30.0

	Vitesse du vent moyen 10 mn	Température	Point de rosée	Humidité \
358652	0.0	280.65	278.75	88.0
544620	2.6	280.85	280.45	97.0
544619	5.7	273.95	273.05	94.0
310595	3.6	283.65	280.85	83.0
61487	5.1	278.05	276.35	89.0

	Visibilité horizontale ...	Température maximale sur 12 heures (°C) \
358652	12000.0 ...	NaN
544620	NaN ...	NaN
544619	4500.0 ...	NaN
310595	40000.0 ...	NaN
61487	9000.0 ...	NaN

	Température maximale sur 24 heures (°C) \
358652	NaN
544620	NaN
544619	NaN
310595	NaN
61487	NaN

	Température minimale du sol sur 12 heures (en °C)	Latitude \
358652	NaN	45.726500
544620	NaN	44.745000
544619	NaN	48.716833
310595	NaN	43.648833
61487	NaN	47.150000

	Longitude	Altitude	department (name)	department (code) \
358652	5.077833	235	Rhône	69
544620	1.396667	260	Lot	46
544619	2.384333	89	Essonne	91
310595	7.209000	2	Alpes-Maritimes	06
61487	-1.608833	26	Loire-Atlantique	44

	region (name)	mois_de_l_annee
358652	Auvergne-Rhône-Alpes	1

544620	Occitanie	1
544619	Île-de-France	1
310595	Provence-Alpes-Côte d'Azur	1
61487	Pays de la Loire	1

[5 rows x 53 columns]

En réalisant cette manipulation nous pouvons remarquer que l'après cette manipulation n'a pas changé le nombre de mesure présentes dans notre jeu de données. On en conclut donc qu'il n'y avait pas de doublon.

Suppression de colonne :

Pour mon étude, les données qui vont être les plus intéressantes vont être celles liées à la **localisation**, **au vent**, **aux rafales**, **à la température**, **aux précipitations** et évidemment la **date**. Je décide donc de supprimer toutes les autres colonnes qui ne sont pas liées à ces paramètres pour pouvoir alléger mon jeu de données.

```
[9]: colonnes_supprimes = ["Point de rosée", "Humidité", "Visibilité horizontale",
    ↳ "Etat du sol", 'Niveau barométrique', 'Méthode de mesure Température du
    ↳ thermomètre mouillé', 'Température du thermomètre mouillé', "Hauteur de la
    ↳ base des nuages de l'étage inférieur", "Type des nuages de l'étage inférieur",
    ↳ "Type des nuages de l'étage moyen", "Type des nuages de l'étage supérieur",
    ↳ "Nébulosité des nuages de l' étage inférieur", 'Phénomène spécial 1',
    ↳ 'Phénomène spécial 2', 'Phénomène spécial 3', 'Phénomène spécial 4', 'Unnamed:
    ↳ 0', 'Hauteur totale de la couche de neige, glace, autre au sol', 'Hauteur de
    ↳ la neige fraîche', 'Periode de mesure de la neige fraiche', 'Temps présent',
    ↳ 'Temps passé 1', 'Temps passé 2', 'Nébulosité totale' ]
df = df_clean.drop(columns=colonnes_supprimes)
df.head()
```

```
[9]:
```

	Date	Pression au niveau mer \
358652	2010-01-01 00:00:00+00:00	99080.0
544620	2010-01-01 00:00:00+00:00	99210.0
544619	2010-01-01 00:00:00+00:00	99530.0
310595	2010-01-01 00:00:00+00:00	99250.0
61487	2010-01-01 00:00:00+00:00	99360.0

	Variation de pression en 3 heures	Direction du vent moyen 10 mn \
358652	-120.0	0.0
544620	20.0	240.0
544619	10.0	30.0
310595	-100.0	310.0
61487	90.0	30.0

	Vitesse du vent moyen 10 mn	Température	Pression station \
358652	0.0	280.65	96250.0
544620	2.6	280.85	96130.0
544619	5.7	273.95	98420.0

310595	3.6	283.65	98920.0
61487	5.1	278.05	99040.0

Variation de pression en 24 heures \	
358652	NaN
544620	NaN
544619	NaN
310595	NaN
61487	NaN

Rafale sur les 10 dernières minutes	Rafales sur une période ... \
358652	NaN 4.1 ...
544620	NaN 7.2 ...
544619	NaN 8.7 ...
310595	NaN 6.7 ...
61487	NaN 9.8 ...

Température maximale sur 12 heures (°C) \	
358652	NaN
544620	NaN
544619	NaN
310595	NaN
61487	NaN

Température maximale sur 24 heures (°C) \	
358652	NaN
544620	NaN
544619	NaN
310595	NaN
61487	NaN

Température minimale du sol sur 12 heures (en °C)	Latitude \
358652	NaN 45.726500
544620	NaN 44.745000
544619	NaN 48.716833
310595	NaN 43.648833
61487	NaN 47.150000

Longitude	Altitude	department (name)	department (code) \
358652	5.077833	235 Rhône	69
544620	1.396667	260 Lot	46
544619	2.384333	89 Essonne	91
310595	7.209000	2 Alpes-Maritimes	06
61487	-1.608833	26 Loire-Atlantique	44

region (name)	mois_de_l_annee
358652	Auvergne-Rhône-Alpes 1

544620	Occitanie	1
544619	Île-de-France	1
310595	Provence-Alpes-Côte d'Azur	1
61487	Pays de la Loire	1

[5 rows x 29 columns]

Affichage de l'en-tête :

```
[10]: entete = list(df.columns)
print('Nombre de colonnes : ' + str(len(entete)))
df.isnull().sum()
```

Nombre de colonnes : 29

```
[10]: Date 0
Pression au niveau mer 1293
Variation de pression en 3 heures 14559
Direction du vent moyen 10 mn 2261
Vitesse du vent moyen 10 mn 2124
Température 1000
Pression station 48981
Variation de pression en 24 heures 333346
Rafale sur les 10 dernières minutes 423701
Rafales sur une période 126228
Periode de mesure de la rafale 125496
Précipitations dans la dernière heure 60339
Précipitations dans les 3 dernières heures 62105
Précipitations dans les 12 dernières heures 293268
Précipitations dans les 24 dernières heures 322997
Nom 0
Température (°C) 1000
Température minimale sur 12 heures (°C) 759068
Température minimale sur 24 heures (°C) 933506
Température maximale sur 12 heures (°C) 759651
Température maximale sur 24 heures (°C) 931962
Température minimale du sol sur 12 heures (en °C) 514301
Latitude 0
Longitude 0
Altitude 0
department (name) 0
department (code) 0
region (name) 0
mois_de_l_annee 0
dtype: int64
```

Nous sommes donc passés de **53 données** par mesure à **29 données** par mesure.

Mise en place de la colonne 'date' en index du jeu de données :

```
[11]: df.set_index(df.columns[0], inplace=True)
df.head()
```

```
[11]:                                     Pression au niveau mer \
Date
2010-01-01 00:00:00+00:00                99080.0
2010-01-01 00:00:00+00:00                99210.0
2010-01-01 00:00:00+00:00                99530.0
2010-01-01 00:00:00+00:00                99250.0
2010-01-01 00:00:00+00:00                99360.0

                                     Variation de pression en 3 heures \
Date
2010-01-01 00:00:00+00:00                -120.0
2010-01-01 00:00:00+00:00                 20.0
2010-01-01 00:00:00+00:00                 10.0
2010-01-01 00:00:00+00:00                -100.0
2010-01-01 00:00:00+00:00                 90.0

                                     Direction du vent moyen 10 mn \
Date
2010-01-01 00:00:00+00:00                 0.0
2010-01-01 00:00:00+00:00                240.0
2010-01-01 00:00:00+00:00                 30.0
2010-01-01 00:00:00+00:00                310.0
2010-01-01 00:00:00+00:00                 30.0

                                     Vitesse du vent moyen 10 mn  Température \
Date
2010-01-01 00:00:00+00:00                 0.0      280.65
2010-01-01 00:00:00+00:00                 2.6      280.85
2010-01-01 00:00:00+00:00                 5.7      273.95
2010-01-01 00:00:00+00:00                 3.6      283.65
2010-01-01 00:00:00+00:00                 5.1      278.05

                                     Pression station \
Date
2010-01-01 00:00:00+00:00                96250.0
2010-01-01 00:00:00+00:00                96130.0
2010-01-01 00:00:00+00:00                98420.0
2010-01-01 00:00:00+00:00                98920.0
2010-01-01 00:00:00+00:00                99040.0

                                     Variation de pression en 24 heures \
Date
2010-01-01 00:00:00+00:00                NaN
2010-01-01 00:00:00+00:00                NaN
```

2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

Rafale sur les 10 dernières minutes \

Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

Rafales sur une période \

Date	
2010-01-01 00:00:00+00:00	4.1
2010-01-01 00:00:00+00:00	7.2
2010-01-01 00:00:00+00:00	8.7
2010-01-01 00:00:00+00:00	6.7
2010-01-01 00:00:00+00:00	9.8

Periode de mesure de la rafale ... \

Date		...
2010-01-01 00:00:00+00:00	-10.0	...
2010-01-01 00:00:00+00:00	-10.0	...
2010-01-01 00:00:00+00:00	-10.0	...
2010-01-01 00:00:00+00:00	-10.0	...
2010-01-01 00:00:00+00:00	-10.0	...

Température maximale sur 12 heures (°C) \

Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

Température maximale sur 24 heures (°C) \

Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

Température minimale du sol sur 12 heures (en °C) \

Date	
2010-01-01 00:00:00+00:00	NaN

2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

	Latitude	Longitude	Altitude	department (name) \
Date				
2010-01-01 00:00:00+00:00	45.726500	5.077833	235	Rhône
2010-01-01 00:00:00+00:00	44.745000	1.396667	260	Lot
2010-01-01 00:00:00+00:00	48.716833	2.384333	89	Essonne
2010-01-01 00:00:00+00:00	43.648833	7.209000	2	Alpes-Maritimes
2010-01-01 00:00:00+00:00	47.150000	-1.608833	26	Loire-Atlantique

	department (code)	region (name) \
Date		
2010-01-01 00:00:00+00:00	69	Auvergne-Rhône-Alpes
2010-01-01 00:00:00+00:00	46	Occitanie
2010-01-01 00:00:00+00:00	91	Île-de-France
2010-01-01 00:00:00+00:00	06	Provence-Alpes-Côte d'Azur
2010-01-01 00:00:00+00:00	44	Pays de la Loire

	mois_de_l_annee
Date	
2010-01-01 00:00:00+00:00	1
2010-01-01 00:00:00+00:00	1
2010-01-01 00:00:00+00:00	1
2010-01-01 00:00:00+00:00	1
2010-01-01 00:00:00+00:00	1

[5 rows x 28 columns]

Réarrangement de l'ordre des colonnes:

```
[12]: nouvel_ordre_colonne = ['mois_de_l_annee', 'Nom', 'region (name)', 'department_
↳(name)', 'department (code)', 'Latitude', 'Longitude', 'Altitude',
↳'Température', 'Température (°C)', 'Température minimale sur 12 heures (°C)',
↳'Température minimale sur 24 heures (°C)', 'Température maximale sur 12 heures_
↳(°C)', 'Température maximale sur 24 heures (°C)', 'Température minimale du sol_
↳sur 12 heures (en °C)', 'Pression au niveau mer', 'Variation de pression en 3_
↳heures', 'Pression station', 'Variation de pression en 24 heures', 'Direction_
↳du vent moyen 10 mn', 'Vitesse du vent moyen 10 mn', 'Rafale sur les 10_
↳dernières minutes', 'Rafales sur une période', 'Periode de mesure de la_
↳rafale', 'Précipitations dans la dernière heure', 'Précipitations dans les 3_
↳dernières heures', 'Précipitations dans les 12 dernières heures',
↳'Précipitations dans les 24 dernières heures']
df = df.reindex(columns=nouvel_ordre_colonne)
df.head()
```

[12]:

	mois_de_l_annee	Nom \
Date		
2010-01-01 00:00:00+00:00	1	LYON-ST EXUPERY
2010-01-01 00:00:00+00:00	1	GOURDON
2010-01-01 00:00:00+00:00	1	ORLY
2010-01-01 00:00:00+00:00	1	NICE
2010-01-01 00:00:00+00:00	1	NANTES-BOUGUENNAIS

	region (name)	department (name) \
Date		
2010-01-01 00:00:00+00:00	Auvergne-Rhône-Alpes	Rhône
2010-01-01 00:00:00+00:00	Occitanie	Lot
2010-01-01 00:00:00+00:00	Île-de-France	Essonne
2010-01-01 00:00:00+00:00	Provence-Alpes-Côte d'Azur	Alpes-Maritimes
2010-01-01 00:00:00+00:00	Pays de la Loire	Loire-Atlantique

	department (code)	Latitude	Longitude	Altitude \
Date				
2010-01-01 00:00:00+00:00	69	45.726500	5.077833	235
2010-01-01 00:00:00+00:00	46	44.745000	1.396667	260
2010-01-01 00:00:00+00:00	91	48.716833	2.384333	89
2010-01-01 00:00:00+00:00	06	43.648833	7.209000	2
2010-01-01 00:00:00+00:00	44	47.150000	-1.608833	26

	Température	Température (°C)	... \
Date			...
2010-01-01 00:00:00+00:00	280.65	7.5	...
2010-01-01 00:00:00+00:00	280.85	7.7	...
2010-01-01 00:00:00+00:00	273.95	0.8	...
2010-01-01 00:00:00+00:00	283.65	10.5	...
2010-01-01 00:00:00+00:00	278.05	4.9	...

	Variation de pression en 24 heures \
Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

	Direction du vent moyen 10 mn \
Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	240.0
2010-01-01 00:00:00+00:00	30.0
2010-01-01 00:00:00+00:00	310.0
2010-01-01 00:00:00+00:00	30.0

Date	Vitesse du vent moyen 10 mn \
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	2.6
2010-01-01 00:00:00+00:00	5.7
2010-01-01 00:00:00+00:00	3.6
2010-01-01 00:00:00+00:00	5.1

Date	Rafale sur les 10 dernières minutes \
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

Date	Rafales sur une période \
2010-01-01 00:00:00+00:00	4.1
2010-01-01 00:00:00+00:00	7.2
2010-01-01 00:00:00+00:00	8.7
2010-01-01 00:00:00+00:00	6.7
2010-01-01 00:00:00+00:00	9.8

Date	Periode de mesure de la rafale \
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0

Date	Précipitations dans la dernière heure \
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

Date	Précipitations dans les 3 dernières heures \
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	3.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

2010-01-01 00:00:00+00:00	0.0
---------------------------	-----

	Précipitations dans les 12 dernières heures \
Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

	Précipitations dans les 24 dernières heures
Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

[5 rows x 28 columns]

Modification des valeurs de la colonne 'mois_de_l_annee' :

```
[13]: dico = {1: 'Janvier', 2: 'Février', 3: 'Mars', 4: 'Avril', 5: 'Mai', 6: 'Juin',
    ↪7: 'Juillet', 8: 'Août', 9: 'Septembre', 10: 'Octobre', 11: 'Novembre', 12:
    ↪'Décembre'}

df_perfect = df.copy()
df_perfect["mois_de_l_annee"] = df["mois_de_l_annee"].map(dico)
df_perfect.head()
```

	mois_de_l_annee	Nom \
Date		
2010-01-01 00:00:00+00:00	Janvier	LYON-ST EXUPERY
2010-01-01 00:00:00+00:00	Janvier	GOURDON
2010-01-01 00:00:00+00:00	Janvier	ORLY
2010-01-01 00:00:00+00:00	Janvier	NICE
2010-01-01 00:00:00+00:00	Janvier	NANTES-BOUGUENNAIS

	region (name)	department (name) \
Date		
2010-01-01 00:00:00+00:00	Auvergne-Rhône-Alpes	Rhône
2010-01-01 00:00:00+00:00	Occitanie	Lot
2010-01-01 00:00:00+00:00	Île-de-France	Essonne
2010-01-01 00:00:00+00:00	Provence-Alpes-Côte d'Azur	Alpes-Maritimes
2010-01-01 00:00:00+00:00	Pays de la Loire	Loire-Atlantique

	department (code)	Latitude	Longitude	Altitude \
Date				

2010-01-01 00:00:00+00:00	69	45.726500	5.077833	235
2010-01-01 00:00:00+00:00	46	44.745000	1.396667	260
2010-01-01 00:00:00+00:00	91	48.716833	2.384333	89
2010-01-01 00:00:00+00:00	06	43.648833	7.209000	2
2010-01-01 00:00:00+00:00	44	47.150000	-1.608833	26

Date	Température	Température (°C)	...	\
2010-01-01 00:00:00+00:00	280.65	7.5	...	
2010-01-01 00:00:00+00:00	280.85	7.7	...	
2010-01-01 00:00:00+00:00	273.95	0.8	...	
2010-01-01 00:00:00+00:00	283.65	10.5	...	
2010-01-01 00:00:00+00:00	278.05	4.9	...	

Date	Variation de pression en 24 heures	\
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	

Date	Direction du vent moyen 10 mn	\
2010-01-01 00:00:00+00:00	0.0	
2010-01-01 00:00:00+00:00	240.0	
2010-01-01 00:00:00+00:00	30.0	
2010-01-01 00:00:00+00:00	310.0	
2010-01-01 00:00:00+00:00	30.0	

Date	Vitesse du vent moyen 10 mn	\
2010-01-01 00:00:00+00:00	0.0	
2010-01-01 00:00:00+00:00	2.6	
2010-01-01 00:00:00+00:00	5.7	
2010-01-01 00:00:00+00:00	3.6	
2010-01-01 00:00:00+00:00	5.1	

Date	Rafale sur les 10 dernières minutes	\
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	
2010-01-01 00:00:00+00:00	NaN	

Rafales sur une période \

Date	
2010-01-01 00:00:00+00:00	4.1
2010-01-01 00:00:00+00:00	7.2
2010-01-01 00:00:00+00:00	8.7
2010-01-01 00:00:00+00:00	6.7
2010-01-01 00:00:00+00:00	9.8

Periode de mesure de la rafale \

Date	
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0

Précipitations dans la dernière heure \

Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

Précipitations dans les 3 dernières heures \

Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	3.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

Précipitations dans les 12 dernières heures \

Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

Précipitations dans les 24 dernières heures

Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

[5 rows x 28 columns]

Grâce à cette manipulation les valeurs numérique de la colonne 'mois_de_l_annee' sont devenues le vrai nom du mois, ce qui facilite l'étude pour la suite.

Modification des valeurs NaN par des 0 :

Maintenant je souhaite remplacer les NaN par des 0. Cependant, il faut faire très **attention** car il n'est pas possible de modifier les NaN pour toutes les colonnes. En effet, remplacer une température en Kelvin par 0 n'a pas de sens. Alors que pour le cas de des précipitations par exemple, le remplacement par 0 reste correct, il est possible qu'il y ait eu 0 mm de précipitation. Je décide donc de faire le remplacement sur des **colonnes bien choisies**.

```
[14]: df_perfect.head()[["Rafale sur les 10 dernières minutes", "Rafales sur une_
    ↳période", "Periode de mesure de la rafale", "Précipitations dans la dernière_
    ↳heure", "Précipitations dans les 3 dernières heures", "Précipitations dans les_
    ↳12 dernières heures", "Précipitations dans les 24 dernières heures"]]
```

```
[14]:                                     Rafale sur les 10 dernières minutes \
```

Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

```
                                     Rafales sur une période \
```

Date	
2010-01-01 00:00:00+00:00	4.1
2010-01-01 00:00:00+00:00	7.2
2010-01-01 00:00:00+00:00	8.7
2010-01-01 00:00:00+00:00	6.7
2010-01-01 00:00:00+00:00	9.8

```
                                     Periode de mesure de la rafale \
```

Date	
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0

```
                                     Précipitations dans la dernière heure \
```

Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

	Précipitations dans les 3 dernières heures \
Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	3.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

	Précipitations dans les 12 dernières heures \
Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

	Précipitations dans les 24 dernières heures
Date	
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN
2010-01-01 00:00:00+00:00	NaN

```
[15]: df_perfect["Rafale sur les 10 dernières minutes"].fillna(0, inplace = True)
df_perfect["Rafales sur une période"].fillna(0, inplace = True)
df_perfect["Periode de mesure de la rafale"].fillna(0, inplace = True)
df_perfect["Précipitations dans la dernière heure"].fillna(0, inplace=True)
df_perfect["Précipitations dans les 3 dernières heures"].fillna(0, inplace=True)
df_perfect["Précipitations dans les 12 dernières heures"].fillna(0, inplace=True)
df_perfect["Précipitations dans les 24 dernières heures"].fillna(0, inplace=True)
df_perfect.head()[["Rafale sur les 10 dernières minutes", "Rafales sur une_
↳période", "Periode de mesure de la rafale","Précipitations dans la dernière_
↳heure", "Précipitations dans les 3 dernières heures", "Précipitations dans les_
↳12 dernières heures", "Précipitations dans les 24 dernières heures"]]
```

	Rafale sur les 10 dernières minutes \
Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

	Rafales sur une période \
Date	

2010-01-01 00:00:00+00:00	4.1
2010-01-01 00:00:00+00:00	7.2
2010-01-01 00:00:00+00:00	8.7
2010-01-01 00:00:00+00:00	6.7
2010-01-01 00:00:00+00:00	9.8

Periode de mesure de la rafale \

Date	
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0
2010-01-01 00:00:00+00:00	-10.0

Précipitations dans la dernière heure \

Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

Précipitations dans les 3 dernières heures \

Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	3.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

Précipitations dans les 12 dernières heures \

Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

Précipitations dans les 24 dernières heures

Date	
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0
2010-01-01 00:00:00+00:00	0.0

On remarque bien ici que nous n'avons plus de valeurs NaN mais des 0 à la place.

Recherche et suppression de valeurs aberrantes :

Nous allons maintenant vérifier que nous n'avons pas de valeurs **aberrantes** dans notre jeu de données. Dans le cas qui suit, je réalise la manipulation avec seulement la **température** et les **précipitations** pour montrer comme exemple, mais il est important de noter que l'analyse de données aberrantes a été réalisée sur l'ensemble de données.

Affichage des températures :

```
[16]: df_perfect["Température (°C)"].describe()
```

```
[16]: count      975597.000000
      mean        15.618396
      std         8.453559
      min        -19.700000
      25%         9.200000
      50%        14.900000
      75%        22.900000
      max         67.700000
      Name: Température (°C), dtype: float64
```

On retrouve une valeur max de **67,7 °C** ce qui semble aberrant.

On va donc chercher à vérifier cette valeur pour savoir si elle est cohérente ou pas. Pour cela, j'ai cherché sur internet la valeur maximale relevée en France. Source : <https://meteofrance.com/magazine/meteo-questions/quelle-est-la-temperature-la-plus-elevee-enregistree-en-france>

```
[17]: max_temp = df_perfect[df_perfect["Température (°C)"] > 46.0] #46 est la
      ↪ température max relevée en France
      max_temp.head()[["Température (°C)", "region (name)"]]
```

```
[17]:
```

	Température (°C)	region (name)
Date		
2010-10-24 06:00:00+00:00	67.7	Martinique
2012-12-16 03:00:00+00:00	66.2	Martinique

On remarque ici la présence de **2 valeurs** de températures un trop élevées. D'après Météo France, la température maximale en Martinique est de **34.6°C**. On en déduit donc que ces 2 températures sont incohérentes et qu'il s'agit d'erreurs des capteurs. On va donc supprimer ces 2 valeurs :

```
[18]: df = df_perfect.copy()
      indexNames = df[df["Température (°C)"] > 46].index
      df.drop(indexNames, inplace = True)
      df["Température (°C)"].describe()
```

```
[18]: count      975550.000000
      mean        15.618520
      std         8.453268
      min        -19.700000
```

```

25%          9.200000
50%         14.900000
75%         22.900000
max          42.000000
Name: Température (°C), dtype: float64

```

Nous avons maintenant une valeur maximale qui est **42°C** ce qui est plus cohérent.

Affichage des précipitations dans l'heure :

```
[19]: df["Précipitations dans la dernière heure"].describe()
```

```

[19]: count      976550.000000
      mean         0.083197
      std         0.665146
      min        -0.100000
      25%         0.000000
      50%         0.000000
      75%         0.000000
      max         93.000000
      Name: Précipitations dans la dernière heure, dtype: float64

```

On remarque que le maximum est à **93 mm**, ce qui paraît aberrant. Regardons combien de fois apparaît cette valeur :

```
[20]: preci = df.groupby("Précipitations dans la dernière heure")
      preci.size()
```

```

[20]: Précipitations dans la dernière heure
-0.1      36284
 0.0     863278
 0.2     27889
 0.3         44
 0.4     11673
      ...
59.5         1
60.0         1
61.5         1
64.9         1
93.0         1
Length: 285, dtype: int64

```

On remarque qu'il y a un grand saut pour la valeur 93. Je vais donc regarder à quelle date elle correspond :

```

[21]: max_preci = df_perfect[df_perfect["Précipitations dans la dernière heure"] > 90 ]
      max_preci.head()[["mois_de_l_annee", "Nom", "region (name)", "Précipitations dans_
      ↪ la dernière heure"]]

```

```
[21]:
```

	mois_de_l_annee	Nom region (name)	\
Date			
2014-09-29 15:00:00+00:00	Septembre	MONTPELLIER	Occitanie


```

Précipitations dans la dernière heure

Date
2014-09-29 15:00:00+00:00          93.0

```

On voit donc ici que la valeur de 93 mm a été mesurée le 29 septembre 2014 à la station de Montpellier. En vérifiant sur internet, d'après un reportage de franceinfo (<https://france3-regions.francetvinfo.fr/occitanie/2014/09/30/un-record-historique-de-pluie-montpellier-et-des-millions-de-degats-dans-l-herault-561820.html>), cette valeur paraît **cohérente** ! Il n'y a donc aucune raison de supprimer cette valeur.

Une fois que tout le jeu de données a été traité et nettoyé il va donc être possible d'afficher les différentes données et de les interpréter.

4 3. Représentation des données

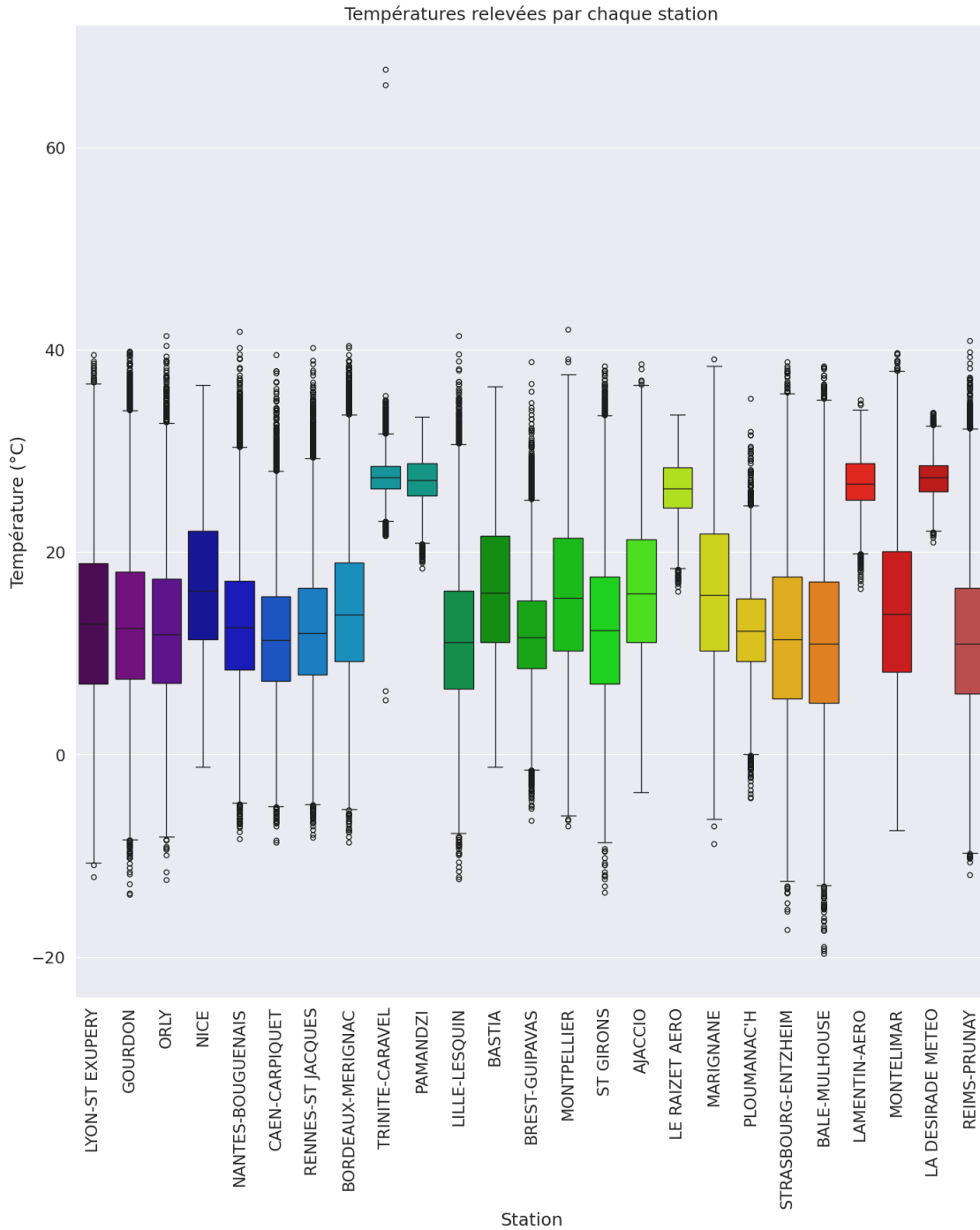
```
[22]: Data = df_perfect.copy()
```

L'objectif ici est d'afficher les données les plus intéressantes pour pouvoir les interpréter et en tirer des conclusions qui peuvent refléter ce que l'on sait déjà et ainsi valider la cohérence de nos données.

Température :

```
[23]: Data = Data.reset_index(drop=True)
plt.figure(figsize = (20,20))
sns.catplot(x="Nom", y="Température (°C)", kind = "box", height = 15, data = Data,
↪ palette="nipy_spectral", legend = False, hue="Nom")
plt.title("Températures relevées par chaque station")
plt.xlabel("Station")
plt.ylabel("Température (°C)")
plt.xticks(rotation=90)
plt.show()
```

<Figure size 2000x2000 with 0 Axes>



```
[24]: print("Moyenne de la température à partir de chaque station : " +
        str(Data["Température (°C)"].mean()) + "°C")
```

Moyenne de la température à partir de chaque station :15.618395710523941°C

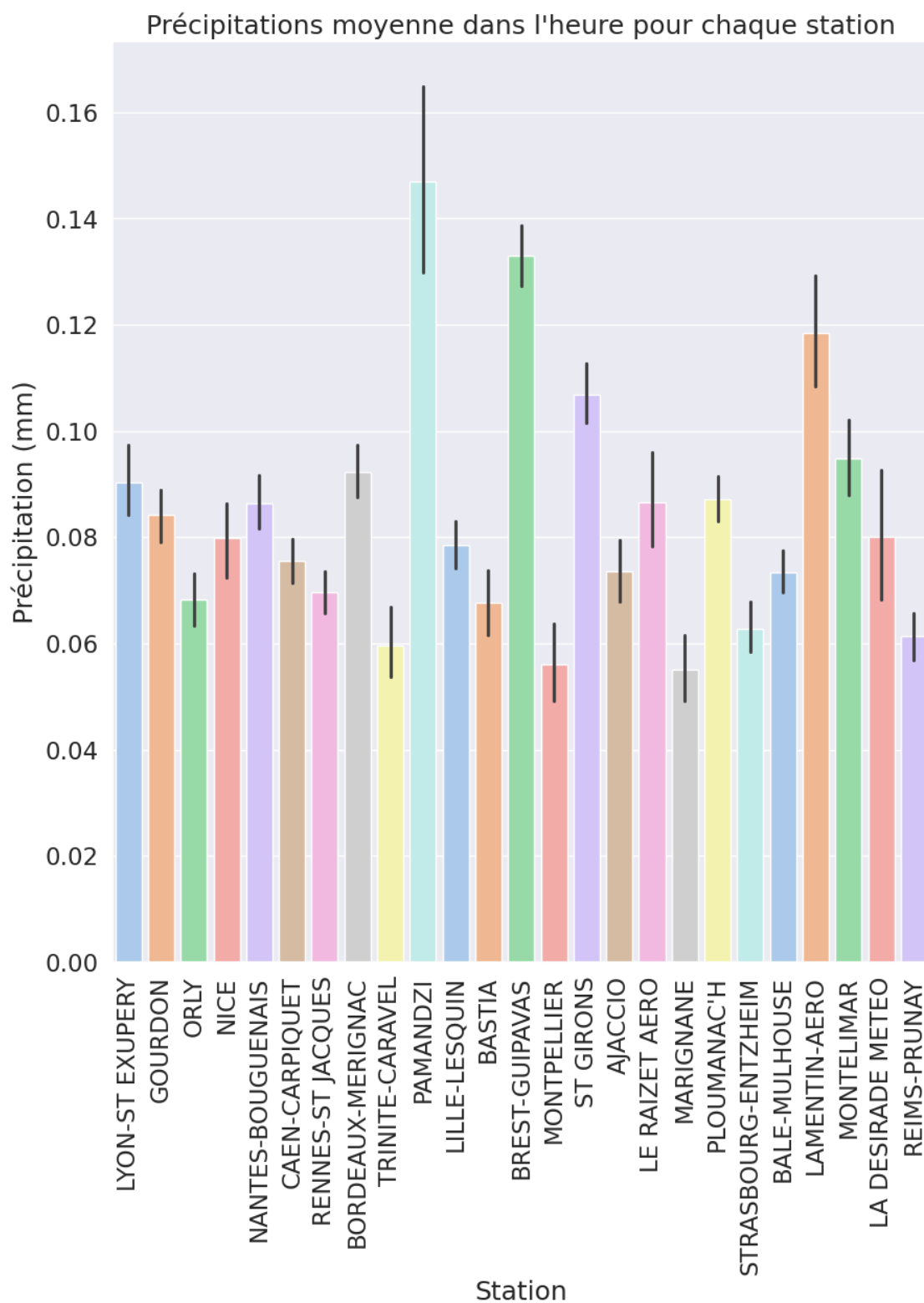
On remarque que pour chaque station nous avons des valeurs cohérentes selon leur localisation. De

plus leurs extremums sont de plus cohérents. La moyenne de température obtenue est elle aussi correcte car elle est de **15.6°C**, or la température moyenne en France métropolitaine est de **14.6°C**. Cette différence de 1°C peut être expliquée par la présence des DROM-COM qui vont avoir tendance à remonter cette moyenne. Source : <https://www.ecologie.gouv.fr/impacts-du-changement-climatique-atmosphere-temperatures-et-precipitations>

Précipitations :

```
[25]: plt.figure(figsize = (20,20))
      sns.catplot(x="Nom", y="Précipitations dans la dernière heure",kind =_
      ↪"bar",hue="Nom",height = 10, data = Data, palette="pastel")
      plt.title("Précipitations moyenne dans l'heure pour chaque station")
      plt.xlabel("Station")
      plt.ylabel("Précipitation (mm)")
      plt.xticks(rotation=90)
      plt.show()
```

<Figure size 2000x2000 with 0 Axes>



```
[26]: print("Moyenne des précipitations dans 1'heure :" + str(Data["Précipitations dans la dernière heure"].mean())+" mm")
```

Moyenne des précipitations dans 1'heure :0.0832019758405975 mm

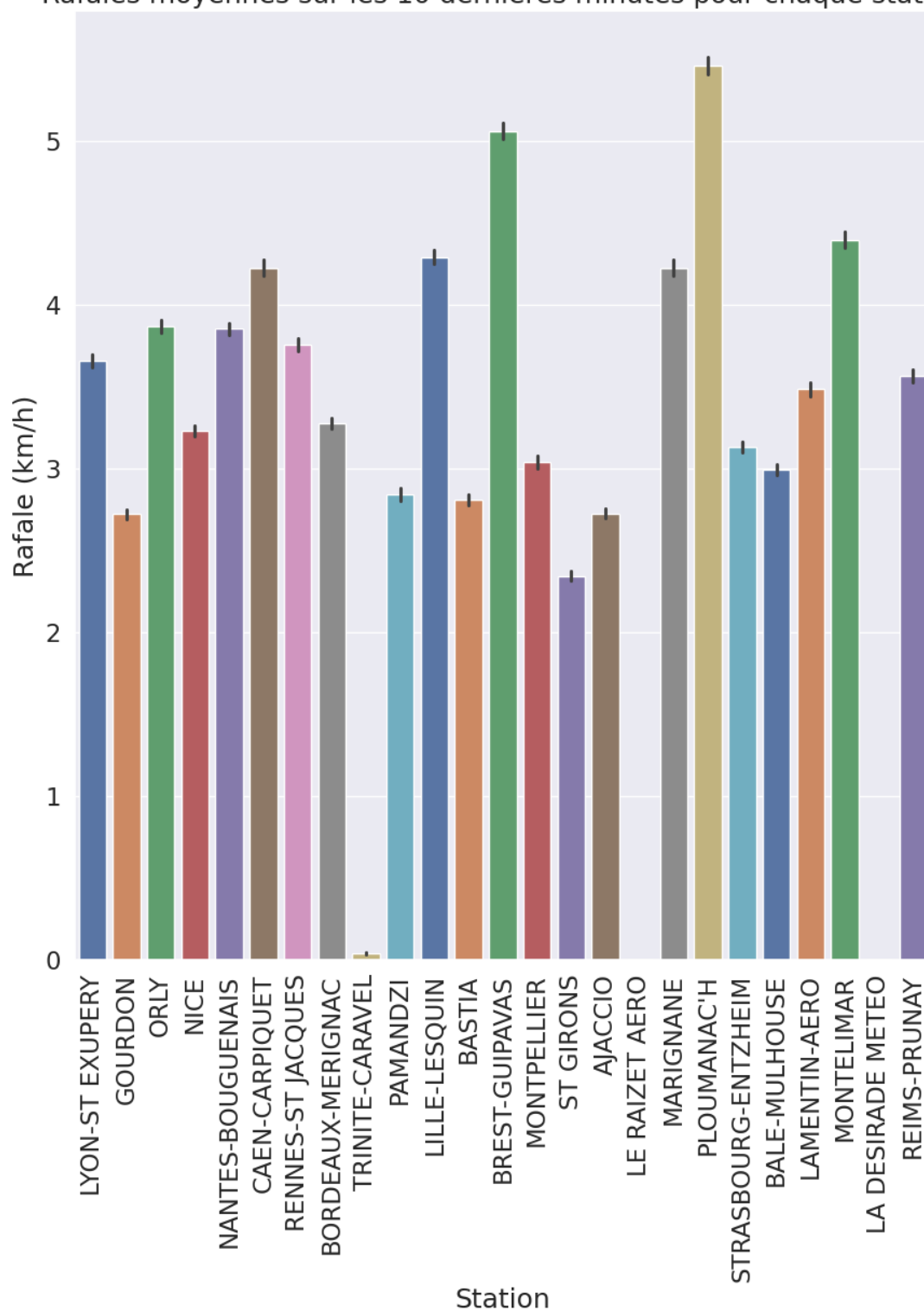
On remarque ici que en moyenne les précipitations sont de à peu près 0.1 mm. On remarque tout de même 4 stations pour lesquelles leur moyenne est au-dessus de la normale.

Rafales:

```
[27]: plt.figure(figsize = (20,20))
sns.catplot(x="Nom", y="Rafale sur les 10 dernières minutes",hue = "Nom",kind = "bar",height = 10, data = Data, palette="deep")
plt.title("Rafales moyennes sur les 10 dernières minutes pour chaque station")
plt.xlabel("Station")
plt.ylabel("Rafale (km/h)")
plt.xticks(rotation=90)
plt.show()
```

<Figure size 2000x2000 with 0 Axes>

Rafales moyennes sur les 10 dernières minutes pour chaque station



```
[28]: print("Moyenne des rafales dans les 10 dernières minutes : " + str(Data["Rafale_
      ↳sur les 10 dernières minutes"].mean())+" km/h")
```

Moyenne des rafales dans les 10 dernières minutes : 3.2614680364572077 km/h

Dans l'ensemble, les rafales moyennes de chaque station reste dans la même plage de valeur sauf pour **Trinite-Caravel** et **Le Raziet Aero**, mais cette différence est sûrement liée au fait qu'il n'y a pas de capteur ou alors qu'il y a des problèmes. De plus, la valeur pour la station **Ploumanac'h** semble cohérente étant donné que cette station est sur le littoral.

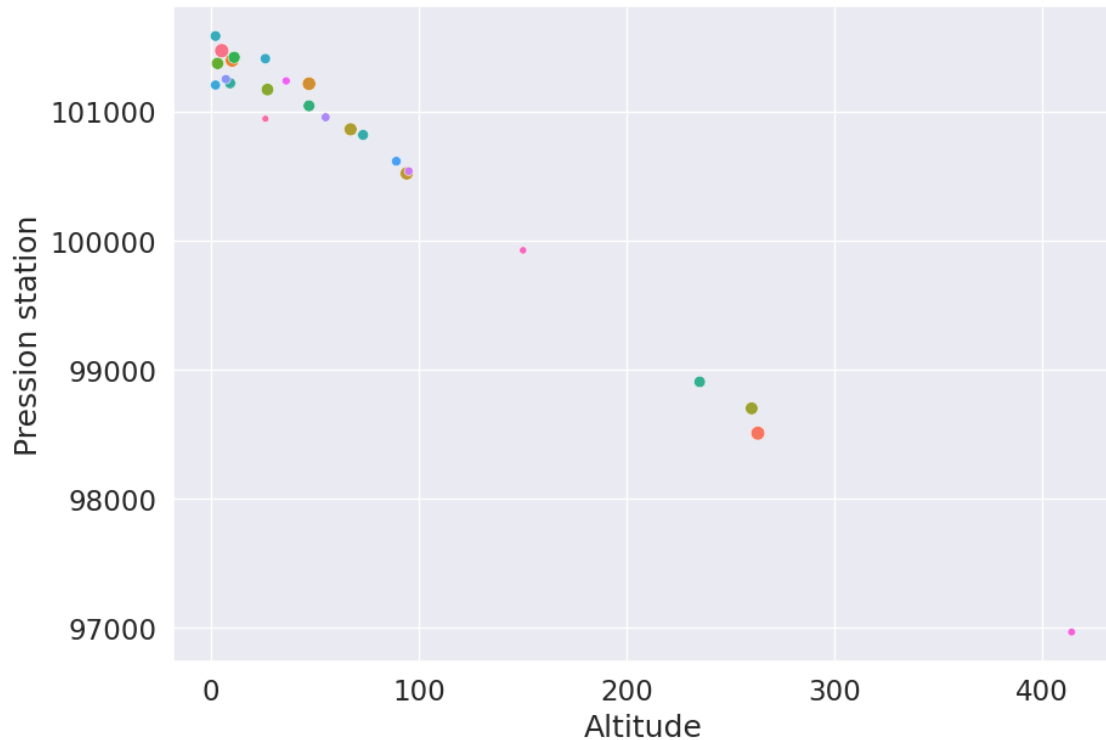
Affichage de la pression moyenne de chaque station en fonction de l'altitude :

```
[29]: Data["Pression station"] = pd.to_numeric(Data["Pression station"],
      ↳errors="coerce")
Data["Altitude"] = pd.to_numeric(Data["Altitude"], errors="coerce")
moy = Data.groupby("Nom")[["Pression station", "Altitude"]].mean()
new_data = pd.DataFrame(moy)
new_data.head(n=10)
```

```
[29]:
```

	Pression station	Altitude
Nom		
AJACCIO	101470.668512	5.0
BALE-MULHOUSE	98505.669433	263.0
BASTIA	101396.228428	10.0
BORDEAUX-MERIGNAC	101214.337013	47.0
BREST-GUIPAVAS	100519.278679	94.0
CAEN-CARPIQUET	100860.959265	67.0
GOURDON	98697.992651	260.0
LA DESIRADE METEO	101169.190608	27.0
LAMENTIN-AERO	101372.077201	3.0
LE RAIZET AERO	101419.258614	11.0

```
[30]: plt.figure(figsize = (10,7))
sns.scatterplot(x="Altitude", y="Pression station", hue="Nom", size="Nom",
      ↳legend=False, data= new_data)
plt.show()
```



Ce que l'on remarque grâce à ce graphe c'est que plus nous sommes en altitude plus la pression atmosphérique est faible. Cette analyse est cohérente avec la réalité des choses.

Affichage de la vitesse moyenne des rafales de chaque station en fonction de l'altitude :

```
[31]: Data["Rafales sur une période"] = pd.to_numeric(Data["Rafales sur une période"],
→errors="coerce")
Data["Altitude"] = pd.to_numeric(Data["Altitude"], errors="coerce")
raf_moy = Data.groupby("Nom")[["Rafales sur une période", "Altitude"]].mean()
df_raf = pd.DataFrame(raf_moy)
df_raf.head(10)
```

```
[31]:
```

	Rafales sur une période	Altitude
Nom		
AJACCIO	5.098792	5.0
BALE-MULHOUSE	5.494648	263.0
BASTIA	5.150491	10.0
BORDEAUX-MERIGNAC	5.953738	47.0
BREST-GUIPAVAS	8.302939	94.0
CAEN-CARPIQUET	7.350596	67.0
GOURDON	5.077951	260.0
LA DESIRADE METEO	0.000000	27.0
LAMENTIN-AERO	4.082125	3.0
LE RAIZET AERO	0.001412	11.0

Ici, on remarque qu'il n'y a pas de lien entre la vitesse du vent et l'altitude. Ce que l'on pense n'est pas vérifié ici.

Affichage de la température moyenne de chaque station en fonction de l'altitude :

```
[32]: Data["Température (°C)"] = pd.to_numeric(Data["Température (°C)"],
↳errors="coerce")
Data["Altitude"] = pd.to_numeric(Data["Altitude"], errors="coerce")
temp_moy = Data.groupby("Nom")[["Température (°C)", "Altitude"]].mean()
df_temp = pd.DataFrame(temp_moy)
df_temp.head(10)
```

```
[32]:
```

	Température (°C)	Altitude
Nom		
AJACCIO	16.228559	5.0
BALE-MULHOUSE	11.336853	263.0
BASTIA	16.441455	10.0
BORDEAUX-MERIGNAC	14.185175	47.0
BREST-GUIPAVAS	11.774124	94.0
CAEN-CARPIQUET	11.470118	67.0
GOURDON	13.000840	260.0
LA DESIRADE METEO	27.410710	27.0
LAMENTIN-AERO	26.948552	3.0
LE RAIZET AERO	26.313581	11.0

Ce graphique nous montre encore des valeurs qui sont cohérentes. En effet, on remarque dans l'ensemble que lorsque la station est proche de la mer, la température moyenne sera plus élevée. Cependant, on ne peut pas dire que plus nous sommes en altitude plus la température moyenne sera basse. En effet, on remarque que pour des stations à moins de 100 mètres d'altitude nous avons des températures moyennes inférieures à 12°C. De plus, notre intervalle d'étude pour l'altitude est assez faible, un peu plus de 400m, pour pouvoir prouver cette hypothèse.

Pour conclure, la représentation de données nous a permis de prouver la cohérence de nos résultats et donc de nos valeurs. De plus, ces représentations ont permis de valider ou non des principes liés à la météorologie.

5 4. Filtrage des mesures

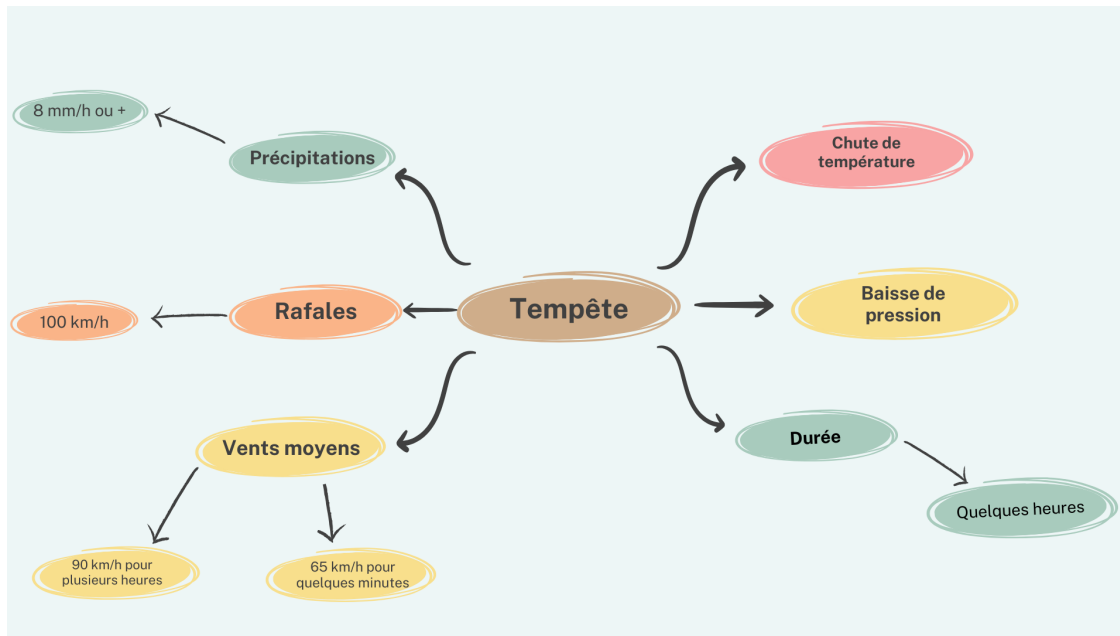
Maintenant que nous avons vérifié l'ensemble des mesures et que notre jeu de données est bien mis en place, nous pouvons donc nous attaquer à la résolution de la **problématique**.

L'objectif ici est de supprimer toutes les mesures qui ne correspondent pas à des mesures de tempêtes. Pour cela je vais utiliser les différentes **caractéristiques des tempêtes** pour pouvoir **filtrer** mes mesures.

D'après les sources qui sont disponibles dans la rubrique **références**, j'ai pu réaliser cette **map mind** qui regroupe les différentes **caractéristiques d'une tempête**.

```
[33]: Image(filename='Mind map tempete.png')
```

```
[33]:
```



[34]: `Data = df_perfect.copy()`

D'après cette map mind, nous avons **6 caractéristiques** propres à la tempête, or elles n'ont pas besoin d'être toutes réunies pour pouvoir définir un phénomène comme étant une tempête. En effet, les rafales, les vents moyens ou alors les précipitations peuvent définir à elles seules un phénomène comme une tempête. Alors que pour les **3 données** de droite, il s'agit plus des **conséquences** d'une tempête.

Filtrage pour les seuils de rafale et des vents moyens :

Dans mon cas, je décide d'étudier les **rafales et les vents moyens** en même temps car ce sont deux caractéristiques qui sont **liées et indépendantes**. Pour cela, je vais tout d'abord garder les mesures où les rafales sont supérieures au seuil et de même pour les vents moyens. Ensuite, je vais **regrouper** les mesures qui sont supérieures au 2 seuils afin de pouvoir déterminer les dates où il y a eu des tempêtes.

En ce qui concerne notre caractéristique sur les rafales, nous avons un seuil à **100 km/h**, ce qui revient à environ **27,8 m/s**.

[35]: `raf_sup_seuil = Data[Data["Rafale sur les 10 dernières minutes"] >= 27.8]`
`raf_sup_seuil.head(n=100)[["Rafale sur les 10 dernières minutes"]]`

[35]:

Date	Rafale sur les 10 dernières minutes
2010-02-28 00:00:00+00:00	29.3
2010-02-28 06:00:00+00:00	29.3
2010-02-28 09:00:00+00:00	28.3
2010-10-25 09:00:00+00:00	28.8


```

2010-10-25 12:00:00+00:00      28.8
...
2021-07-06 00:00:00+00:00      27.9
2021-10-21 00:00:00+00:00      27.9
2022-01-05 12:00:00+00:00      27.9
2022-01-05 15:00:00+00:00      32.5
2022-02-18 15:00:00+00:00      36.2

```

[100 rows x 1 columns]

```
[36]: raf_sup_seuil["Rafale sur les 10 dernières minutes"].describe()
```

```

[36]: count      117.000000
      mean       29.979487
      std        2.607170
      min       27.800000
      25%       28.100000
      50%       29.200000
      75%       30.600000
      max       43.200000
      Name: Rafale sur les 10 dernières minutes, dtype: float64

```

De plus, une tempête est définie à partir du moment où les vents moyens atteignent au minimum **65 km/h** soit une vitesse de **18 m/s**.

```

[37]: vent_sup_seuil = Data[Data["Vitesse du vent moyen 10 mn"] >= 18]
      vent_sup_seuil.head()[["Vitesse du vent moyen 10 mn"]]

```

```

[37]:                                Vitesse du vent moyen 10 mn
Date
2010-01-28 09:00:00+00:00      18.0
2010-02-28 00:00:00+00:00      18.5
2010-05-15 18:00:00+00:00      18.0
2010-05-16 12:00:00+00:00      19.0
2010-10-25 09:00:00+00:00      20.1

```

```
[38]: vent_sup_seuil["Vitesse du vent moyen 10 mn"].describe()
```

```

[38]: count      189.000000
      mean       19.255026
      std        1.396401
      min       18.000000
      25%       18.300000
      50%       19.000000
      75%       19.600000
      max       28.300000
      Name: Vitesse du vent moyen 10 mn, dtype: float64

```

Je vais donc créer un nouveau **DataFrame** dans lequel il y a l'ensemble des mesures pour lesquelles la date est en commun entre les mesures de rafale et vents moyens supérieur à leur seuil respectif.

```
[39]: regroup = pd.merge(vent_sup_seuil, raf_sup_seuil, on="Date", how='inner')
regroup.head(n=65)
```

```
[39]:
```

	Date	mois_de_l_annee_x	Nom_x \
	2010-02-28 00:00:00+00:00	Février	BORDEAUX-MERIGNAC
	2010-10-25 09:00:00+00:00	Octobre	MARIGNANE
	2010-10-25 12:00:00+00:00	Octobre	MARIGNANE
	2011-12-16 00:00:00+00:00	Décembre	BREST-GUIPAVAS
	2011-12-24 15:00:00+00:00	Décembre	MARIGNANE

	2023-11-24 15:00:00+00:00	Novembre	MARIGNANE
	2023-11-24 18:00:00+00:00	Novembre	MARIGNANE
	2023-12-20 15:00:00+00:00	Décembre	MARIGNANE
	2023-12-21 00:00:00+00:00	Décembre	MARIGNANE
	2024-01-07 09:00:00+00:00	Janvier	MARIGNANE

	Date	region (name)_x	department (name)_x \
	2010-02-28 00:00:00+00:00	Nouvelle-Aquitaine	Gironde
	2010-10-25 09:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
	2010-10-25 12:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
	2011-12-16 00:00:00+00:00	Bretagne	Finistère
	2011-12-24 15:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône

	2023-11-24 15:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
	2023-11-24 18:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
	2023-12-20 15:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
	2023-12-21 00:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
	2024-01-07 09:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône

	Date	department (code)_x	Latitude_x	Longitude_x \
	2010-02-28 00:00:00+00:00	33	44.830667	-0.691333
	2010-10-25 09:00:00+00:00	13	43.437667	5.216000
	2010-10-25 12:00:00+00:00	13	43.437667	5.216000
	2011-12-16 00:00:00+00:00	29	48.444167	-4.412000
	2011-12-24 15:00:00+00:00	13	43.437667	5.216000

	2023-11-24 15:00:00+00:00	13	43.437667	5.216000
	2023-11-24 18:00:00+00:00	13	43.437667	5.216000
	2023-12-20 15:00:00+00:00	13	43.437667	5.216000
	2023-12-21 00:00:00+00:00	13	43.437667	5.216000
	2024-01-07 09:00:00+00:00	13	43.437667	5.216000

Date	Altitude_x	Température_x	Température (°C)_x	...	\
2010-02-28 00:00:00+00:00	47	292.25	19.1	...	
2010-10-25 09:00:00+00:00	9	284.05	10.9	...	
2010-10-25 12:00:00+00:00	9	283.45	10.3	...	
2011-12-16 00:00:00+00:00	94	286.65	13.5	...	
2011-12-24 15:00:00+00:00	9	282.35	9.2	...	
...	
2023-11-24 15:00:00+00:00	9	287.05	13.9	...	
2023-11-24 18:00:00+00:00	9	286.15	13.0	...	
2023-12-20 15:00:00+00:00	9	284.35	11.2	...	
2023-12-21 00:00:00+00:00	9	281.85	8.7	...	
2024-01-07 09:00:00+00:00	9	280.35	7.2	...	

Date	Variation de pression en 24 heures_y	\
2010-02-28 00:00:00+00:00	NaN	
2010-10-25 09:00:00+00:00	NaN	
2010-10-25 12:00:00+00:00	NaN	
2011-12-16 00:00:00+00:00	NaN	
2011-12-24 15:00:00+00:00	NaN	
...	...	
2023-11-24 15:00:00+00:00	-1250.0	
2023-11-24 18:00:00+00:00	-1100.0	
2023-12-20 15:00:00+00:00	-870.0	
2023-12-21 00:00:00+00:00	-300.0	
2024-01-07 09:00:00+00:00	140.0	

Date	Direction du vent moyen 10 mn_y	\
2010-02-28 00:00:00+00:00	210.0	
2010-10-25 09:00:00+00:00	320.0	
2010-10-25 12:00:00+00:00	10.0	
2011-12-16 00:00:00+00:00	240.0	
2011-12-24 15:00:00+00:00	310.0	
...	...	
2023-11-24 15:00:00+00:00	330.0	
2023-11-24 18:00:00+00:00	320.0	
2023-12-20 15:00:00+00:00	330.0	
2023-12-21 00:00:00+00:00	320.0	
2024-01-07 09:00:00+00:00	320.0	

Date	Vitesse du vent moyen 10 mn_y	\
2010-02-28 00:00:00+00:00	18.5	
2010-10-25 09:00:00+00:00	20.1	

2010-10-25 12:00:00+00:00	13.4
2011-12-16 00:00:00+00:00	19.5
2011-12-24 15:00:00+00:00	20.1
...	...
2023-11-24 15:00:00+00:00	21.4
2023-11-24 18:00:00+00:00	20.9
2023-12-20 15:00:00+00:00	20.1
2023-12-21 00:00:00+00:00	19.4
2024-01-07 09:00:00+00:00	19.2

Rafale sur les 10 dernières minutes_y \

Date	
2010-02-28 00:00:00+00:00	29.3
2010-10-25 09:00:00+00:00	28.8
2010-10-25 12:00:00+00:00	28.8
2011-12-16 00:00:00+00:00	30.4
2011-12-24 15:00:00+00:00	27.8
...	...
2023-11-24 15:00:00+00:00	27.9
2023-11-24 18:00:00+00:00	30.1
2023-12-20 15:00:00+00:00	28.9
2023-12-21 00:00:00+00:00	27.9
2024-01-07 09:00:00+00:00	28.6

Rafales sur une période_y \

Date	
2010-02-28 00:00:00+00:00	29.3
2010-10-25 09:00:00+00:00	28.8
2010-10-25 12:00:00+00:00	28.8
2011-12-16 00:00:00+00:00	32.4
2011-12-24 15:00:00+00:00	29.8
...	...
2023-11-24 15:00:00+00:00	28.4
2023-11-24 18:00:00+00:00	30.1
2023-12-20 15:00:00+00:00	28.9
2023-12-21 00:00:00+00:00	27.9
2024-01-07 09:00:00+00:00	28.6

Periode de mesure de la rafale_y \

Date	
2010-02-28 00:00:00+00:00	-10.0
2010-10-25 09:00:00+00:00	-10.0
2010-10-25 12:00:00+00:00	-10.0
2011-12-16 00:00:00+00:00	-10.0
2011-12-24 15:00:00+00:00	-10.0
...	...
2023-11-24 15:00:00+00:00	-10.0

2023-11-24 18:00:00+00:00	-10.0
2023-12-20 15:00:00+00:00	-10.0
2023-12-21 00:00:00+00:00	-10.0
2024-01-07 09:00:00+00:00	-10.0

Précipitations dans la dernière heure_y \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.2
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	0.6
2011-12-24 15:00:00+00:00	0.0
...	...
2023-11-24 15:00:00+00:00	0.0
2023-11-24 18:00:00+00:00	0.0
2023-12-20 15:00:00+00:00	0.0
2023-12-21 00:00:00+00:00	0.0
2024-01-07 09:00:00+00:00	0.0

Précipitations dans les 3 dernières heures_y \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.2
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	7.0
2011-12-24 15:00:00+00:00	0.0
...	...
2023-11-24 15:00:00+00:00	0.0
2023-11-24 18:00:00+00:00	0.0
2023-12-20 15:00:00+00:00	0.0
2023-12-21 00:00:00+00:00	0.0
2024-01-07 09:00:00+00:00	0.0

Précipitations dans les 12 dernières heures_y \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.0
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	0.0
2011-12-24 15:00:00+00:00	0.0
...	...
2023-11-24 15:00:00+00:00	0.0
2023-11-24 18:00:00+00:00	0.0
2023-12-20 15:00:00+00:00	0.0
2023-12-21 00:00:00+00:00	0.0
2024-01-07 09:00:00+00:00	0.0

Précipitations dans les 24 dernières heures_y

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.0
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	0.0
2011-12-24 15:00:00+00:00	0.0
...	...
2023-11-24 15:00:00+00:00	0.0
2023-11-24 18:00:00+00:00	0.0
2023-12-20 15:00:00+00:00	0.0
2023-12-21 00:00:00+00:00	0.0
2024-01-07 09:00:00+00:00	0.0

[65 rows x 56 columns]

```
[40]: e = list(regroup.columns)
      print(e)
```

```
['mois_de_l_annee_x', 'Nom_x', 'region (name)_x', 'department (name)_x',
'department (code)_x', 'Latitude_x', 'Longitude_x', 'Altitude_x',
'Température_x', 'Température (°C)_x', 'Température minimale sur 12 heures
(°C)_x', 'Température minimale sur 24 heures (°C)_x', 'Température maximale sur
12 heures (°C)_x', 'Température maximale sur 24 heures (°C)_x', 'Température
minimale du sol sur 12 heures (en °C)_x', 'Pression au niveau mer_x', 'Variation
de pression en 3 heures_x', 'Pression station_x', 'Variation de pression en 24
heures_x', 'Direction du vent moyen 10 mn_x', 'Vitesse du vent moyen 10 mn_x',
'Rafale sur les 10 dernières minutes_x', 'Rafales sur une période_x', 'Periode
de mesure de la rafale_x', 'Précipitations dans la dernière heure_x',
'Précipitations dans les 3 dernières heures_x', 'Précipitations dans les 12
dernières heures_x', 'Précipitations dans les 24 dernières heures_x',
'mois_de_l_annee_y', 'Nom_y', 'region (name)_y', 'department (name)_y',
'department (code)_y', 'Latitude_y', 'Longitude_y', 'Altitude_y',
'Température_y', 'Température (°C)_y', 'Température minimale sur 12 heures
(°C)_y', 'Température minimale sur 24 heures (°C)_y', 'Température maximale sur
12 heures (°C)_y', 'Température maximale sur 24 heures (°C)_y', 'Température
minimale du sol sur 12 heures (en °C)_y', 'Pression au niveau mer_y', 'Variation
de pression en 3 heures_y', 'Pression station_y', 'Variation de pression en 24
heures_y', 'Direction du vent moyen 10 mn_y', 'Vitesse du vent moyen 10 mn_y',
'Rafale sur les 10 dernières minutes_y', 'Rafales sur une période_y', 'Periode
de mesure de la rafale_y', 'Précipitations dans la dernière heure_y',
'Précipitations dans les 3 dernières heures_y', 'Précipitations dans les 12
dernières heures_y', 'Précipitations dans les 24 dernières heures_y']
```

Nous avons ici un dataframe avec les colonnes en doublons, je vais donc les supprimer pour avoir un dataframe dans lequel les valeurs de vents moyens et de rafales sont supérieurs aux différents seuils.

```
[41]: colon_suppr = ['mois_de_l_annee_y', 'Nom_y', 'region (name)_y', 'department_
↳(name)_y', 'department (code)_y', 'Latitude_y', 'Longitude_y', 'Altitude_y',
↳'Température_y', 'Température (°C)_y', 'Température minimale sur 12 heures_
↳(°C)_y', 'Température minimale sur 24 heures (°C)_y', 'Température maximale_
↳sur 12 heures (°C)_y', 'Température maximale sur 24 heures (°C)_y',
↳'Température minimale du sol sur 12 heures (en °C)_y', 'Pression au niveau_
↳mer_y', 'Variation de pression en 3 heures_y', 'Pression station_y',
↳'Variation de pression en 24 heures_y', 'Direction du vent moyen 10 mn_y',
↳'Vitesse du vent moyen 10 mn_y', 'Rafale sur les 10 dernières minutes_y',
↳'Rafales sur une période_y', 'Periode de mesure de la rafale_y',
↳'Précipitations dans la dernière heure_y', 'Précipitations dans les 3_
↳dernières heures_y', 'Précipitations dans les 12 dernières heures_y',
↳'Précipitations dans les 24 dernières heures_y']
regroup_clean = regroup.drop(columns=colon_suppr)
print(len(regroup_clean))
regroup_clean.head()
```

65

```
[41]:
```

	mois_de_l_annee_x	Nom_x \
Date		
2010-02-28 00:00:00+00:00	Février	BORDEAUX-MERIGNAC
2010-10-25 09:00:00+00:00	Octobre	MARIGNANE
2010-10-25 12:00:00+00:00	Octobre	MARIGNANE
2011-12-16 00:00:00+00:00	Décembre	BREST-GUIPAVAS
2011-12-24 15:00:00+00:00	Décembre	MARIGNANE

	region (name)_x	department (name)_x \
Date		
2010-02-28 00:00:00+00:00	Nouvelle-Aquitaine	Gironde
2010-10-25 09:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
2010-10-25 12:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône
2011-12-16 00:00:00+00:00	Bretagne	Finistère
2011-12-24 15:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône

	department (code)_x	Latitude_x	Longitude_x \
Date			
2010-02-28 00:00:00+00:00	33	44.830667	-0.691333
2010-10-25 09:00:00+00:00	13	43.437667	5.216000
2010-10-25 12:00:00+00:00	13	43.437667	5.216000
2011-12-16 00:00:00+00:00	29	48.444167	-4.412000
2011-12-24 15:00:00+00:00	13	43.437667	5.216000

	Altitude_x	Température_x	Température (°C)_x	...	\
Date					
2010-02-28 00:00:00+00:00	47	292.25	19.1	...	
2010-10-25 09:00:00+00:00	9	284.05	10.9	...	

2010-10-25 12:00:00+00:00	9	283.45	10.3	...
2011-12-16 00:00:00+00:00	94	286.65	13.5	...
2011-12-24 15:00:00+00:00	9	282.35	9.2	...

Variation de pression en 24 heures_x \

Date	
2010-02-28 00:00:00+00:00	NaN
2010-10-25 09:00:00+00:00	NaN
2010-10-25 12:00:00+00:00	NaN
2011-12-16 00:00:00+00:00	NaN
2011-12-24 15:00:00+00:00	NaN

Direction du vent moyen 10 mn_x \

Date	
2010-02-28 00:00:00+00:00	210.0
2010-10-25 09:00:00+00:00	320.0
2010-10-25 12:00:00+00:00	330.0
2011-12-16 00:00:00+00:00	240.0
2011-12-24 15:00:00+00:00	310.0

Vitesse du vent moyen 10 mn_x \

Date	
2010-02-28 00:00:00+00:00	18.5
2010-10-25 09:00:00+00:00	20.1
2010-10-25 12:00:00+00:00	18.0
2011-12-16 00:00:00+00:00	19.5
2011-12-24 15:00:00+00:00	20.1

Rafale sur les 10 dernières minutes_x \

Date	
2010-02-28 00:00:00+00:00	29.3
2010-10-25 09:00:00+00:00	28.8
2010-10-25 12:00:00+00:00	24.2
2011-12-16 00:00:00+00:00	30.4
2011-12-24 15:00:00+00:00	27.8

Rafales sur une période_x \

Date	
2010-02-28 00:00:00+00:00	29.3
2010-10-25 09:00:00+00:00	28.8
2010-10-25 12:00:00+00:00	25.2
2011-12-16 00:00:00+00:00	32.4
2011-12-24 15:00:00+00:00	29.8

Periode de mesure de la rafale_x \

Date	
2010-02-28 00:00:00+00:00	-10.0

2010-10-25 09:00:00+00:00	-10.0
2010-10-25 12:00:00+00:00	-10.0
2011-12-16 00:00:00+00:00	-10.0
2011-12-24 15:00:00+00:00	-10.0

Précipitations dans la dernière heure_x \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.2
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	0.6
2011-12-24 15:00:00+00:00	0.0

Précipitations dans les 3 dernières heures_x \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.2
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	7.0
2011-12-24 15:00:00+00:00	0.0

Précipitations dans les 12 dernières heures_x \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.0
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	0.0
2011-12-24 15:00:00+00:00	0.0

Précipitations dans les 24 dernières heures_x

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.0
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	0.0
2011-12-24 15:00:00+00:00	0.0

[5 rows x 28 columns]

```
[42]: nouveaux_noms = {col: col.replace('_x', '') for col in regroup_clean.columns}
regroup_clean.rename(columns= nouveaux_noms, inplace=True)
regroup_clean.head()
```

```
[42]:
```

	mois_de_l_annee	Nom \
Date		
2010-02-28 00:00:00+00:00	Février	BORDEAUX-MERIGNAC
2010-10-25 09:00:00+00:00	Octobre	MARIGNANE

2010-10-25 12:00:00+00:00	Octobre	MARIGNANE
2011-12-16 00:00:00+00:00	Décembre	BREST-GUIPAVAS
2011-12-24 15:00:00+00:00	Décembre	MARIGNANE

Date	region (name)	department (name)	\
2010-02-28 00:00:00+00:00	Nouvelle-Aquitaine	Gironde	
2010-10-25 09:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône	
2010-10-25 12:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône	
2011-12-16 00:00:00+00:00	Bretagne	Finistère	
2011-12-24 15:00:00+00:00	Provence-Alpes-Côte d'Azur	Bouches-du-Rhône	

Date	department (code)	Latitude	Longitude	Altitude	\
2010-02-28 00:00:00+00:00	33	44.830667	-0.691333	47	
2010-10-25 09:00:00+00:00	13	43.437667	5.216000	9	
2010-10-25 12:00:00+00:00	13	43.437667	5.216000	9	
2011-12-16 00:00:00+00:00	29	48.444167	-4.412000	94	
2011-12-24 15:00:00+00:00	13	43.437667	5.216000	9	

Date	Température	Température (°C)	...	\
2010-02-28 00:00:00+00:00	292.25	19.1	...	
2010-10-25 09:00:00+00:00	284.05	10.9	...	
2010-10-25 12:00:00+00:00	283.45	10.3	...	
2011-12-16 00:00:00+00:00	286.65	13.5	...	
2011-12-24 15:00:00+00:00	282.35	9.2	...	

Date	Variation de pression en 24 heures	\
2010-02-28 00:00:00+00:00	NaN	
2010-10-25 09:00:00+00:00	NaN	
2010-10-25 12:00:00+00:00	NaN	
2011-12-16 00:00:00+00:00	NaN	
2011-12-24 15:00:00+00:00	NaN	

Date	Direction du vent moyen 10 mn	\
2010-02-28 00:00:00+00:00	210.0	
2010-10-25 09:00:00+00:00	320.0	
2010-10-25 12:00:00+00:00	330.0	
2011-12-16 00:00:00+00:00	240.0	
2011-12-24 15:00:00+00:00	310.0	

Date	Vitesse du vent moyen 10 mn	\
2010-02-28 00:00:00+00:00	18.5	

2010-10-25 09:00:00+00:00	20.1
2010-10-25 12:00:00+00:00	18.0
2011-12-16 00:00:00+00:00	19.5
2011-12-24 15:00:00+00:00	20.1

Rafale sur les 10 dernières minutes \

Date	
2010-02-28 00:00:00+00:00	29.3
2010-10-25 09:00:00+00:00	28.8
2010-10-25 12:00:00+00:00	24.2
2011-12-16 00:00:00+00:00	30.4
2011-12-24 15:00:00+00:00	27.8

Rafales sur une période \

Date	
2010-02-28 00:00:00+00:00	29.3
2010-10-25 09:00:00+00:00	28.8
2010-10-25 12:00:00+00:00	25.2
2011-12-16 00:00:00+00:00	32.4
2011-12-24 15:00:00+00:00	29.8

Periode de mesure de la rafale \

Date	
2010-02-28 00:00:00+00:00	-10.0
2010-10-25 09:00:00+00:00	-10.0
2010-10-25 12:00:00+00:00	-10.0
2011-12-16 00:00:00+00:00	-10.0
2011-12-24 15:00:00+00:00	-10.0

Précipitations dans la dernière heure \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.2
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	0.6
2011-12-24 15:00:00+00:00	0.0

Précipitations dans les 3 dernières heures \

Date	
2010-02-28 00:00:00+00:00	0.0
2010-10-25 09:00:00+00:00	0.2
2010-10-25 12:00:00+00:00	0.0
2011-12-16 00:00:00+00:00	7.0
2011-12-24 15:00:00+00:00	0.0

Précipitations dans les 12 dernières heures \

Date

```

2010-02-28 00:00:00+00:00      0.0
2010-10-25 09:00:00+00:00      0.0
2010-10-25 12:00:00+00:00      0.0
2011-12-16 00:00:00+00:00      0.0
2011-12-24 15:00:00+00:00      0.0

```

Précipitations dans les 24 dernières heures

```

Date
2010-02-28 00:00:00+00:00      0.0
2010-10-25 09:00:00+00:00      0.0
2010-10-25 12:00:00+00:00      0.0
2011-12-16 00:00:00+00:00      0.0
2011-12-24 15:00:00+00:00      0.0

```

[5 rows x 28 columns]

La manipulation précédente a permis de renommer l'ensemble des colonnes en supprimant le 'x'.

```

[43]: vent_raf = regroup_clean.copy()
      region = vent_raf.groupby("region (name)")
      region.size()

```

```

[43]: region (name)
      Auvergne-Rhône-Alpes      1
      Bretagne                 14
      Corse                     7
      Grand Est                 1
      Hauts-de-France           4
      Normandie                 4
      Nouvelle-Aquitaine        3
      Pays de la Loire           1
      Provence-Alpes-Côte d'Azur 30
      dtype: int64

```

```

[44]: PACA = vent_raf[vent_raf["region (name)"] == "Provence-Alpes-Côte d'Azur"]
      PACA.head(n=30)[["mois_de_l_annee", "Nom", "Rafale sur les 10 dernières minutes",
      ↪ "Vitesse du vent moyen 10 mn"]]

```

```

[44]:      mois_de_l_annee      Nom \
      Date
      2010-10-25 09:00:00+00:00      Octobre  MARIGNANE
      2010-10-25 12:00:00+00:00      Octobre  MARIGNANE
      2011-12-24 15:00:00+00:00      Décembre  MARIGNANE
      2011-12-29 15:00:00+00:00      Décembre  MARIGNANE
      2012-10-28 06:00:00+00:00      Octobre  MARIGNANE
      2013-02-02 12:00:00+00:00      Février  MARIGNANE
      2013-03-06 09:00:00+00:00      Mars    MARIGNANE
      2013-03-14 09:00:00+00:00      Mars    MARIGNANE

```

2013-03-14 12:00:00+00:00	Mars	MARIGNANE
2013-03-14 15:00:00+00:00	Mars	MARIGNANE
2017-02-06 12:00:00+00:00	Février	MARIGNANE
2017-03-06 18:00:00+00:00	Mars	MARIGNANE
2017-11-13 12:00:00+00:00	Novembre	MARIGNANE
2018-10-01 15:00:00+00:00	Octobre	MARIGNANE
2018-11-27 09:00:00+00:00	Novembre	MARIGNANE
2019-01-02 12:00:00+00:00	Janvier	MARIGNANE
2019-03-25 18:00:00+00:00	Mars	MARIGNANE
2019-05-05 03:00:00+00:00	Mai	MARIGNANE
2019-05-05 09:00:00+00:00	Mai	MARIGNANE
2020-11-20 00:00:00+00:00	Novembre	MARIGNANE
2022-01-05 12:00:00+00:00	Janvier	MARIGNANE
2022-01-05 15:00:00+00:00	Janvier	MARIGNANE
2022-04-02 12:00:00+00:00	Avril	MARIGNANE
2023-08-28 03:00:00+00:00	Août	MARIGNANE
2023-08-28 06:00:00+00:00	Août	MARIGNANE
2023-11-24 15:00:00+00:00	Novembre	MARIGNANE
2023-11-24 18:00:00+00:00	Novembre	MARIGNANE
2023-12-20 15:00:00+00:00	Décembre	MARIGNANE
2023-12-21 00:00:00+00:00	Décembre	MARIGNANE
2024-01-07 09:00:00+00:00	Janvier	MARIGNANE

Rafale sur les 10 dernières minutes \

Date	
2010-10-25 09:00:00+00:00	28.8
2010-10-25 12:00:00+00:00	24.2
2011-12-24 15:00:00+00:00	27.8
2011-12-29 15:00:00+00:00	28.3
2012-10-28 06:00:00+00:00	31.4
2013-02-02 12:00:00+00:00	32.9
2013-03-06 09:00:00+00:00	27.8
2013-03-14 09:00:00+00:00	27.8
2013-03-14 12:00:00+00:00	30.9
2013-03-14 15:00:00+00:00	27.8
2017-02-06 12:00:00+00:00	35.1
2017-03-06 18:00:00+00:00	28.1
2017-11-13 12:00:00+00:00	30.6
2018-10-01 15:00:00+00:00	29.0
2018-11-27 09:00:00+00:00	27.9
2019-01-02 12:00:00+00:00	28.3
2019-03-25 18:00:00+00:00	29.3
2019-05-05 03:00:00+00:00	28.3
2019-05-05 09:00:00+00:00	31.5
2020-11-20 00:00:00+00:00	28.0
2022-01-05 12:00:00+00:00	27.9
2022-01-05 15:00:00+00:00	32.5

2022-04-02 12:00:00+00:00	29.7
2023-08-28 03:00:00+00:00	30.4
2023-08-28 06:00:00+00:00	29.5
2023-11-24 15:00:00+00:00	27.9
2023-11-24 18:00:00+00:00	30.1
2023-12-20 15:00:00+00:00	28.9
2023-12-21 00:00:00+00:00	27.9
2024-01-07 09:00:00+00:00	28.6

Date	Vitesse du vent moyen 10 mn
2010-10-25 09:00:00+00:00	20.1
2010-10-25 12:00:00+00:00	18.0
2011-12-24 15:00:00+00:00	20.1
2011-12-29 15:00:00+00:00	20.6
2012-10-28 06:00:00+00:00	20.6
2013-02-02 12:00:00+00:00	22.1
2013-03-06 09:00:00+00:00	18.0
2013-03-14 09:00:00+00:00	19.0
2013-03-14 12:00:00+00:00	23.1
2013-03-14 15:00:00+00:00	19.5
2017-02-06 12:00:00+00:00	23.5
2017-03-06 18:00:00+00:00	19.5
2017-11-13 12:00:00+00:00	21.4
2018-10-01 15:00:00+00:00	19.5
2018-11-27 09:00:00+00:00	20.2
2019-01-02 12:00:00+00:00	20.5
2019-03-25 18:00:00+00:00	21.5
2019-05-05 03:00:00+00:00	18.9
2019-05-05 09:00:00+00:00	20.7
2020-11-20 00:00:00+00:00	18.3
2022-01-05 12:00:00+00:00	19.7
2022-01-05 15:00:00+00:00	20.8
2022-04-02 12:00:00+00:00	18.7
2023-08-28 03:00:00+00:00	21.0
2023-08-28 06:00:00+00:00	20.4
2023-11-24 15:00:00+00:00	21.4
2023-11-24 18:00:00+00:00	20.9
2023-12-20 15:00:00+00:00	20.1
2023-12-21 00:00:00+00:00	19.4
2024-01-07 09:00:00+00:00	19.2

Le résultat est surprenant étant donné que la région **PACA** n'est pas forcément une région très connue pour subir beaucoup de tempête, cependant ce résultat ne semble **pas aberrant** car certaines caractéristiques peuvent prouver ce résultat. En effet, la station **Marignane** se situe proche de la **mer Méditerranée** qui est bien connue pour ses **mistral**s et ses **vents de l'Est**. Ainsi, on en déduit que ces résultats sont causés par la proximité de la station avec la mer. Cependant, on peut en déduire que ces mesures sont des mesures liées à des tempêtes mais qui sont moins importantes

qu'à la normale.

```
[45]: Breizh = vent_raf[vent_raf["region (name)"] == "Bretagne"]
Breizh.head()[["mois_de_l_annee", "Nom", "Rafale sur les 10 dernières minutes",
↪ "Vitesse du vent moyen 10 mn", "Altitude"]]
```

```
[45]:
```

	mois_de_l_annee	Nom \
Date		
2011-12-16 00:00:00+00:00	Décembre	BREST-GUIPAVAS
2013-03-11 15:00:00+00:00	Mars	PLOUMANAC'H
2013-12-23 12:00:00+00:00	Décembre	BREST-GUIPAVAS
2013-12-23 21:00:00+00:00	Décembre	BREST-GUIPAVAS
2013-12-24 00:00:00+00:00	Décembre	BREST-GUIPAVAS

	Rafale sur les 10 dernières minutes \
Date	
2011-12-16 00:00:00+00:00	30.4
2013-03-11 15:00:00+00:00	27.8
2013-12-23 12:00:00+00:00	27.8
2013-12-23 21:00:00+00:00	29.3
2013-12-24 00:00:00+00:00	36.5

	Vitesse du vent moyen 10 mn	Altitude
Date		
2011-12-16 00:00:00+00:00	19.5	94
2013-03-11 15:00:00+00:00	21.1	55
2013-12-23 12:00:00+00:00	18.0	94
2013-12-23 21:00:00+00:00	20.1	94
2013-12-24 00:00:00+00:00	19.5	94

On remarque aussi que la **Bretagne** dépasse souvent le seuil mais cette fois-ci ce résultat n'est pas surprenant étant donné que cette région est connue pour avoir beaucoup de tempêtes dû à sa proximité avec l'océan Atlantique.

```
[46]: region.size()
```

```
[46]: region (name)
```

Auvergne-Rhône-Alpes	1
Bretagne	14
Corse	7
Grand Est	1
Hauts-de-France	4
Normandie	4
Nouvelle-Aquitaine	3
Pays de la Loire	1
Provence-Alpes-Côte d'Azur	30

```
dtype: int64
```

D'après les résultats précédents, on remarque que les régions qui sont le plus touchées par des vents

moyens et des rafales supérieures aux seuils sont : **Provence-Alpes-Côte d'Azur, Bretagne, Corse, Hauts-de-France, Normandie et Nouvelle-Aquitaine.**

Filtrage pour le seuil de précipitation :

Dans ce cas-là, je décide d'étudier en même temps les précipitations ainsi que la durée d'une tempête qui est de quelques heures, soit environ **3 heures**. Étudier seulement les précipitations me permettra d'identifier la présence de tempête ou pas étant donné que les précipitations font parties des caractéristiques d'une tempête.

Ainsi, je vais pouvoir étudier **les précipitations qui se sont passées dans les 3 dernières heures** afin d'identifier de potentielles tempêtes.

D'après la carte mentale, une tempête est définie si il y a **8 mm ou plus par heure**. Dans notre cas, nous voulons étudier les précipitations sur 3 heures. Pour cela notre seuil va donc être de **24 mm pour 3 heures**.

```
[47]: precipitation = Data[Data["Précipitations dans les 3 dernières heures"] >= 24]
      print(len(precipitation))
      precipitation.head()[["Précipitations dans les 3 dernières heures", "region_",
                           ↪(name)", "Nom"]]
```

536

```
[47]:
```

Précipitations dans les 3 dernières heures \		
Date		
2010-01-04 03:00:00+00:00		34.0
2010-01-07 09:00:00+00:00		47.0
2010-02-16 21:00:00+00:00		24.0
2010-06-09 21:00:00+00:00		28.0
2010-06-10 21:00:00+00:00		27.0

	region (name)	Nom
Date		
2010-01-04 03:00:00+00:00	Mayotte	PAMANDZI
2010-01-07 09:00:00+00:00	Mayotte	PAMANDZI
2010-02-16 21:00:00+00:00	Corse	BASTIA
2010-06-09 21:00:00+00:00	Grand Est	STRASBOURG-ENTZHEIM
2010-06-10 21:00:00+00:00	Normandie	CAEN-CARPIQUET

```
[48]: precipitation["Précipitations dans les 3 dernières heures"].describe()
```

```
[48]: count    536.000000
      mean     36.784328
      std     16.634316
      min     24.000000
      25%     26.700000
      50%     31.900000
      75%     40.600000
      max    201.000000
```


Name: Précipitations dans les 3 dernières heures, dtype: float64

```
[49]: region_precip = precipitation.groupby("region (name)")
      region_precip.size()
```

```
[49]: region (name)
      Auvergne-Rhône-Alpes      59
      Bretagne                  9
      Corse                     41
      Grand Est                 15
      Guadeloupe                73
      Hauts-de-France           7
      Martinique                91
      Mayotte                   101
      Normandie                 4
      Nouvelle-Aquitaine        10
      Occitanie                 56
      Pays de la Loire          4
      Provence-Alpes-Côte d'Azur 55
      Île-de-France             11
      dtype: int64
```

Finalement, nous avons **536 mesures** supérieures au seuil de précipitation. Ce résultat ne paraît pas aberrant étant donné que les phénomènes météorologiques liés à la pluie sont bien plus récurrents que ceux liés à des vents forts.

Regroupement des mesures présentant des tempêtes :

L'objectif ici est juste de créer un nouveau **DataFrame** qui **regroupe** l'ensemble des mesures supérieures aux seuils de vents moyens, rafales et aussi de précipitation.

```
[50]: regroup_seuil = pd.concat([vent_raf, precipitation]).sort_index()
      print(len(regroup_seuil))
      regroup_seuil.head(n=1000)
```

601

```
[50]:
```

	mois_de_l_annee	Nom \
Date		
2010-01-04 03:00:00+00:00	Janvier	PAMANDZI
2010-01-07 09:00:00+00:00	Janvier	PAMANDZI
2010-02-16 21:00:00+00:00	Février	BASTIA
2010-02-28 00:00:00+00:00	Février	BORDEAUX-MERIGNAC
2010-06-09 21:00:00+00:00	Juin	STRASBOURG-ENTZHEIM
...
2024-01-18 09:00:00+00:00	Janvier	PAMANDZI
2024-01-25 03:00:00+00:00	Janvier	PAMANDZI
2024-01-30 06:00:00+00:00	Janvier	TRINITE-CARAVEL
2024-02-09 00:00:00+00:00	Février	LE RAIZET AERO

2024-02-11 12:00:00+00:00

Février

PAMANDZI

region (name) department (name) \

Date

2010-01-04 03:00:00+00:00	Mayotte	Mayotte
2010-01-07 09:00:00+00:00	Mayotte	Mayotte
2010-02-16 21:00:00+00:00	Corse	Haute-Corse
2010-02-28 00:00:00+00:00	Nouvelle-Aquitaine	Gironde
2010-06-09 21:00:00+00:00	Grand Est	Bas-Rhin
...
2024-01-18 09:00:00+00:00	Mayotte	Mayotte
2024-01-25 03:00:00+00:00	Mayotte	Mayotte
2024-01-30 06:00:00+00:00	Martinique	Martinique
2024-02-09 00:00:00+00:00	Guadeloupe	Guadeloupe
2024-02-11 12:00:00+00:00	Mayotte	Mayotte

department (code) Latitude Longitude Altitude \

Date

2010-01-04 03:00:00+00:00	976	-12.805500	45.282833	7
2010-01-07 09:00:00+00:00	976	-12.805500	45.282833	7
2010-02-16 21:00:00+00:00	2b	42.540667	9.485167	10
2010-02-28 00:00:00+00:00	33	44.830667	-0.691333	47
2010-06-09 21:00:00+00:00	67	48.549500	7.640333	150
...
2024-01-18 09:00:00+00:00	976	-12.805500	45.282833	7
2024-01-25 03:00:00+00:00	976	-12.805500	45.282833	7
2024-01-30 06:00:00+00:00	972	14.774500	-60.875333	26
2024-02-09 00:00:00+00:00	971	16.264000	-61.516333	11
2024-02-11 12:00:00+00:00	976	-12.805500	45.282833	7

Température Température (°C) ... \

Date

2010-01-04 03:00:00+00:00	298.95	25.8	...
2010-01-07 09:00:00+00:00	300.25	27.1	...
2010-02-16 21:00:00+00:00	282.35	9.2	...
2010-02-28 00:00:00+00:00	292.25	19.1	...
2010-06-09 21:00:00+00:00	290.85	17.7	...
...
2024-01-18 09:00:00+00:00	298.85	25.7	...
2024-01-25 03:00:00+00:00	299.35	26.2	...
2024-01-30 06:00:00+00:00	296.75	23.6	...
2024-02-09 00:00:00+00:00	293.75	20.6	...
2024-02-11 12:00:00+00:00	298.45	25.3	...

Variation de pression en 24 heures \

Date

2010-01-04 03:00:00+00:00	0.0
---------------------------	-----

2010-01-07 09:00:00+00:00	10.0
2010-02-16 21:00:00+00:00	NaN
2010-02-28 00:00:00+00:00	NaN
2010-06-09 21:00:00+00:00	NaN
...	...
2024-01-18 09:00:00+00:00	-100.0
2024-01-25 03:00:00+00:00	-60.0
2024-01-30 06:00:00+00:00	-80.0
2024-02-09 00:00:00+00:00	480.0
2024-02-11 12:00:00+00:00	110.0

Direction du vent moyen 10 mn \

Date	
2010-01-04 03:00:00+00:00	330.0
2010-01-07 09:00:00+00:00	20.0
2010-02-16 21:00:00+00:00	10.0
2010-02-28 00:00:00+00:00	210.0
2010-06-09 21:00:00+00:00	10.0
...	...
2024-01-18 09:00:00+00:00	310.0
2024-01-25 03:00:00+00:00	250.0
2024-01-30 06:00:00+00:00	250.0
2024-02-09 00:00:00+00:00	320.0
2024-02-11 12:00:00+00:00	260.0

Vitesse du vent moyen 10 mn \

Date	
2010-01-04 03:00:00+00:00	3.1
2010-01-07 09:00:00+00:00	1.0
2010-02-16 21:00:00+00:00	2.6
2010-02-28 00:00:00+00:00	18.5
2010-06-09 21:00:00+00:00	5.7
...	...
2024-01-18 09:00:00+00:00	7.5
2024-01-25 03:00:00+00:00	9.1
2024-01-30 06:00:00+00:00	4.0
2024-02-09 00:00:00+00:00	7.3
2024-02-11 12:00:00+00:00	4.4

Rafale sur les 10 dernières minutes \

Date	
2010-01-04 03:00:00+00:00	0.0
2010-01-07 09:00:00+00:00	0.0
2010-02-16 21:00:00+00:00	0.0
2010-02-28 00:00:00+00:00	29.3
2010-06-09 21:00:00+00:00	0.0
...	...

2024-01-18 09:00:00+00:00	11.8
2024-01-25 03:00:00+00:00	11.9
2024-01-30 06:00:00+00:00	9.2
2024-02-09 00:00:00+00:00	0.0
2024-02-11 12:00:00+00:00	6.0

Rafales sur une période \

Date	
2010-01-04 03:00:00+00:00	8.2
2010-01-07 09:00:00+00:00	7.2
2010-02-16 21:00:00+00:00	5.1
2010-02-28 00:00:00+00:00	29.3
2010-06-09 21:00:00+00:00	13.4
...	...
2024-01-18 09:00:00+00:00	11.8
2024-01-25 03:00:00+00:00	11.9
2024-01-30 06:00:00+00:00	9.9
2024-02-09 00:00:00+00:00	0.0
2024-02-11 12:00:00+00:00	7.7

Periode de mesure de la rafale \

Date	
2010-01-04 03:00:00+00:00	-10.0
2010-01-07 09:00:00+00:00	-10.0
2010-02-16 21:00:00+00:00	-10.0
2010-02-28 00:00:00+00:00	-10.0
2010-06-09 21:00:00+00:00	-10.0
...	...
2024-01-18 09:00:00+00:00	-10.0
2024-01-25 03:00:00+00:00	-10.0
2024-01-30 06:00:00+00:00	-10.0
2024-02-09 00:00:00+00:00	0.0
2024-02-11 12:00:00+00:00	-10.0

Précipitations dans la dernière heure \

Date	
2010-01-04 03:00:00+00:00	12.0
2010-01-07 09:00:00+00:00	6.0
2010-02-16 21:00:00+00:00	6.0
2010-02-28 00:00:00+00:00	0.0
2010-06-09 21:00:00+00:00	0.8
...	...
2024-01-18 09:00:00+00:00	21.5
2024-01-25 03:00:00+00:00	5.1
2024-01-30 06:00:00+00:00	2.4
2024-02-09 00:00:00+00:00	2.6
2024-02-11 12:00:00+00:00	2.6

Date	Précipitations dans les 3 dernières heures \
2010-01-04 03:00:00+00:00	34.0
2010-01-07 09:00:00+00:00	47.0
2010-02-16 21:00:00+00:00	24.0
2010-02-28 00:00:00+00:00	0.0
2010-06-09 21:00:00+00:00	28.0
...	...
2024-01-18 09:00:00+00:00	27.6
2024-01-25 03:00:00+00:00	40.4
2024-01-30 06:00:00+00:00	25.6
2024-02-09 00:00:00+00:00	35.9
2024-02-11 12:00:00+00:00	38.0

Date	Précipitations dans les 12 dernières heures \
2010-01-04 03:00:00+00:00	0.0
2010-01-07 09:00:00+00:00	0.0
2010-02-16 21:00:00+00:00	0.0
2010-02-28 00:00:00+00:00	0.0
2010-06-09 21:00:00+00:00	0.0
...	...
2024-01-18 09:00:00+00:00	62.0
2024-01-25 03:00:00+00:00	46.6
2024-01-30 06:00:00+00:00	33.7
2024-02-09 00:00:00+00:00	35.9
2024-02-11 12:00:00+00:00	38.8

Date	Précipitations dans les 24 dernières heures
2010-01-04 03:00:00+00:00	0.0
2010-01-07 09:00:00+00:00	0.0
2010-02-16 21:00:00+00:00	0.0
2010-02-28 00:00:00+00:00	0.0
2010-06-09 21:00:00+00:00	0.0
...	...
2024-01-18 09:00:00+00:00	62.0
2024-01-25 03:00:00+00:00	0.0
2024-01-30 06:00:00+00:00	34.7
2024-02-09 00:00:00+00:00	35.9
2024-02-11 12:00:00+00:00	39.2

[601 rows x 28 columns]

Nous avons ici l'ensemble des mesures qui correspondent à des tempêtes d'après le seuil des différentes caractéristiques.

```
[51]: region_total = regroup_seuil.groupby(["region (name)"])
      region_total.size()
```

```
[51]: region (name)
      Auvergne-Rhône-Alpes      60
      Bretagne                 23
      Corse                    48
      Grand Est                 16
      Guadeloupe               73
      Hauts-de-France          11
      Martinique               91
      Mayotte                  101
      Normandie                 8
      Nouvelle-Aquitaine       13
      Occitanie                 56
      Pays de la Loire          5
      Provence-Alpes-Côte d'Azur 85
      Île-de-France            11
      dtype: int64
```

Afin de filtrer encore plus nos mesures, on va utiliser une conséquence des tempêtes qui est la chute de pression. Je vais donc utiliser la donnée “**Variation de pression en 3 heures**” pour pouvoir réduire le nombre de mesures correspondant à des tempêtes.

```
[52]: chute_pression = regroup_seuil[regroup_seuil["Variation de pression en 3_
      ↪heures"] < 0]
      print(len(chute_pression))
      chute_pression.head()[["Variation de pression en 3 heures"]]
```

226

```
[52]: Variation de pression en 3 heures
      Date
      2010-06-15 18:00:00+00:00      -190.0
      2010-08-14 15:00:00+00:00      -20.0
      2010-09-06 09:00:00+00:00      -80.0
      2010-09-07 00:00:00+00:00      -80.0
      2010-10-02 18:00:00+00:00     -170.0
```

Nous sommes passé de **536 mesures** à **226 mesures**.

Afin de filtrer encore plus il faut étudier la caractéristique de **baisse de température**. Cependant nous n’avons pas de données de variation de température disponible.

Je pensais pouvoir calculer cette variation de température en faisant la température maximale sur **12 heures moins la température minimale**. Le problème est que cette variation est sur une période de temps trop longues par rapport à 3 heures, mais aussi que je ne peux pas savoir s’ il s’agit d’une **baisse de température ou une augmentation**. Il m’est donc **impossible** de filtrer encore plus les mesures avec les données qu’ils me restent.

6 5. Traitement des mesures indiquant des tempêtes

```
[53]: tempetes = chute_pression.copy()
```

Affichage du nombre de tempête par station :

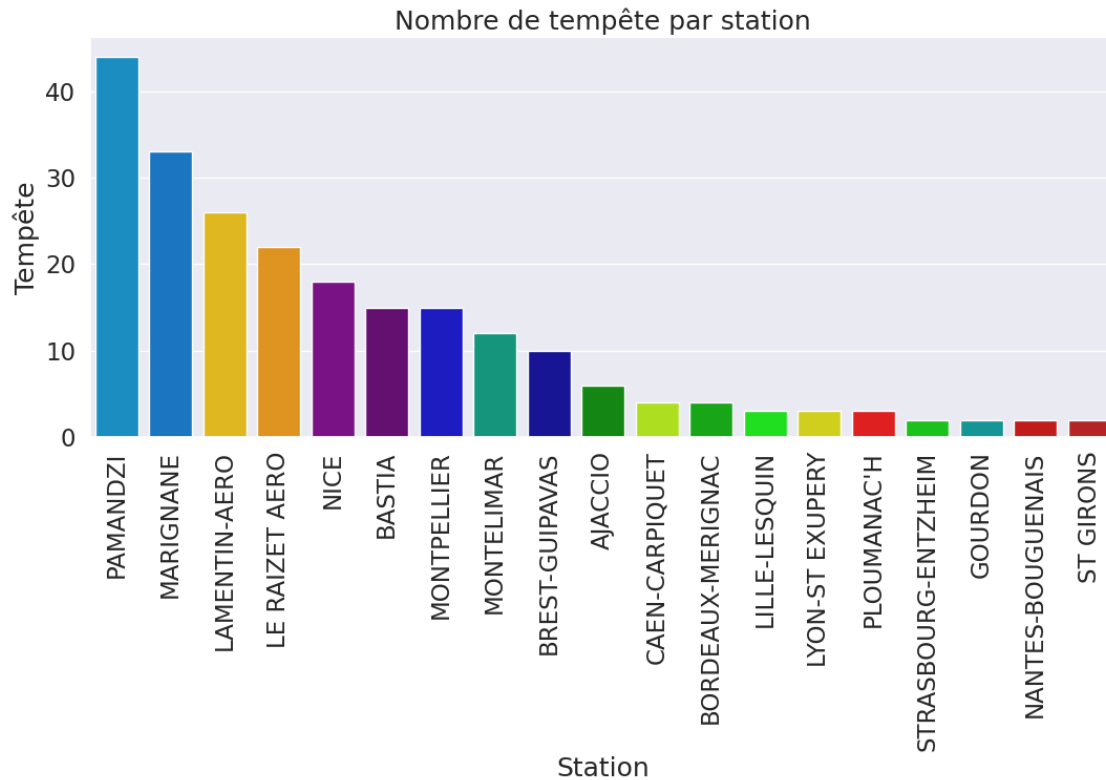
```
[54]: ordered_genres= tempetes["Nom"].value_counts().index.tolist()
      tempetes.groupby("Nom").size()
```

```
[54]: Nom
      AJACCIO                6
      BASTIA                 15
      BORDEAUX-MERIGNAC      4
      BREST-GUIPAVAS        10
      CAEN-CARPIQUET         4
      GOURDON                 2
      LAMENTIN-AERO          26
      LE RAIZET AERO         22
      LILLE-LESQUIN          3
      LYON-ST EXUPERY        3
      MARIGNANE              33
      MONTELMAR             12
      MONTPELLIER           15
      NANTES-BOUGUENAIS      2
      NICE                   18
      PAMANDZI               44
      PLOUMANAC'H           3
      ST GIRONS              2
      STRASBOURG-ENTZHEIM     2
      dtype: int64
```

```
[55]: plt.figure(figsize=(10,10))
      sns.catplot(x="Nom", kind="count", data=
      ↪tempetes,hue="Nom",legend=False,order=ordered_genres, height = 5, aspect = 2.
      ↪3,palette = 'nipy_spectral')
      plt.xlabel('Station')
      plt.ylabel('Tempête')
      plt.xticks(rotation = 90)
      plt.title("Nombre de tempête par station")
```

```
[55]: Text(0.5, 1.0, 'Nombre de tempête par station')
```

<Figure size 1000x1000 with 0 Axes>



Affichage du nombre de tempête par région :

```
[56]: ordered_genres1= tempetes["region (name)"].value_counts().index.tolist()
      tempetes.groupby("region (name)").size()
```

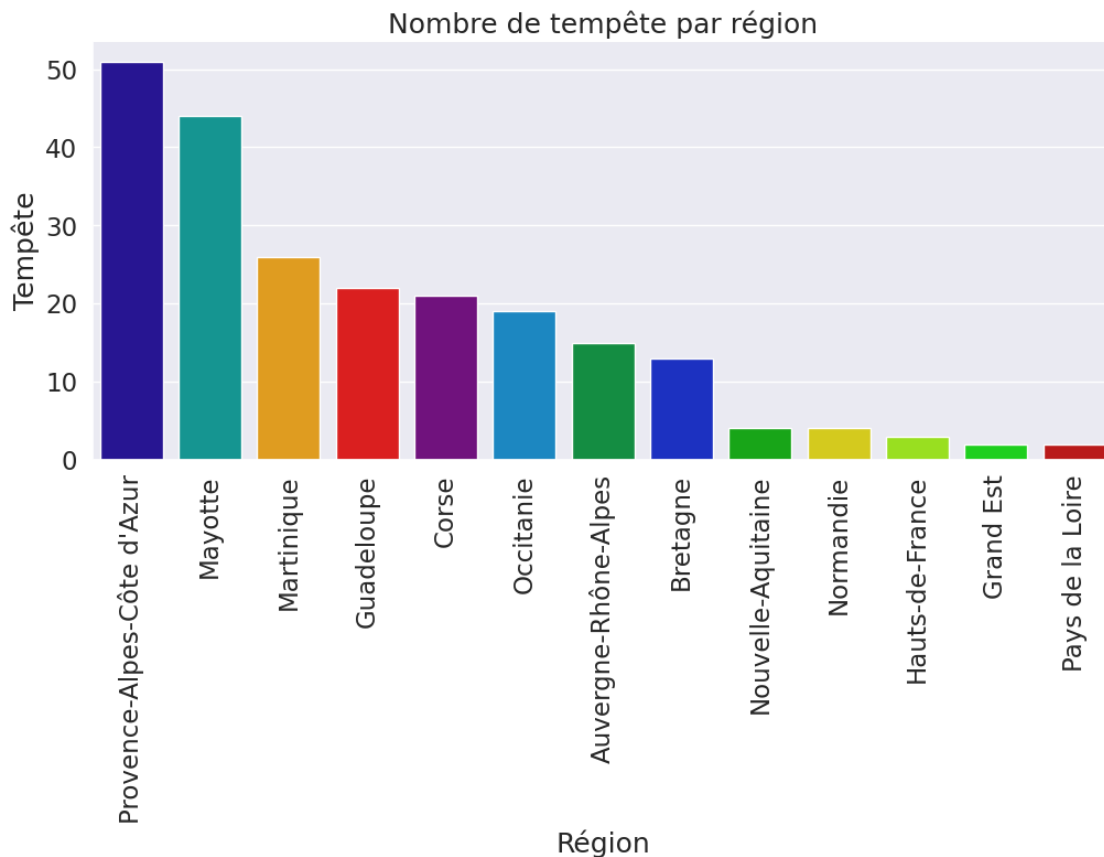
```
[56]: region (name)
      Auvergne-Rhône-Alpes      15
      Bretagne                  13
      Corse                     21
      Grand Est                  2
      Guadeloupe                22
      Hauts-de-France            3
      Martinique                 26
      Mayotte                    44
      Normandie                  4
      Nouvelle-Aquitaine         4
      Occitanie                  19
      Pays de la Loire            2
      Provence-Alpes-Côte d'Azur  51
      dtype: int64
```



```
[57]: plt.figure(figsize=(10,10))
sns.catplot(x="region (name)", kind="count", data= tempetes,hue="region (name)",
↳ legend=False, order=ordered_genres1, height = 5, aspect = 2.3,palette =
↳ 'nipy_spectral')
plt.xlabel('Région')
plt.ylabel('Tempête')
plt.xticks(rotation = 90)
plt.title("Nombre de tempête par région")
```

```
[57]: Text(0.5, 1.0, 'Nombre de tempête par région')
```

<Figure size 1000x1000 with 0 Axes>



Ainsi nous pouvons dire que les 5 régions les plus touchées en France par des tempêtes depuis 2010 sont les suivantes : **Provence-Alpes-Côte d'Azur, Mayotte, Martinique, Guadeloupe et la Corse.**

Ce qui est surprenant dans ce résultat est le fait que la région Provence-Alpes-Côte d'Azur soit la première région ayant le plus de tempête depuis 2010 alors que cette région est plutôt connue pour son climat méditerranéen, c'est-à-dire chaud avec beaucoup de soleil.

De plus, il est important de noter que dans cette étude nous impliquons les **DROM-COM**, or ces

régions là ne se situent pas au même endroit que la France géographiquement. Ainsi, ces régions sont plus sujettes à des **climats tropicaux** menant à des **tempêtes plus sévères** qu'en France. C'est donc pour cela que ces régions sont dans les 5 premières. Il serait donc intéressant de regarder les régions de **France métropolitaine uniquement**.

Affichage nombre de tempête par région en France métropolitaine :

Pour réaliser cette étude il faut enlever des données les mesures liées aux régions suivantes : **Mayotte, Martinique et Guadeloupe**.

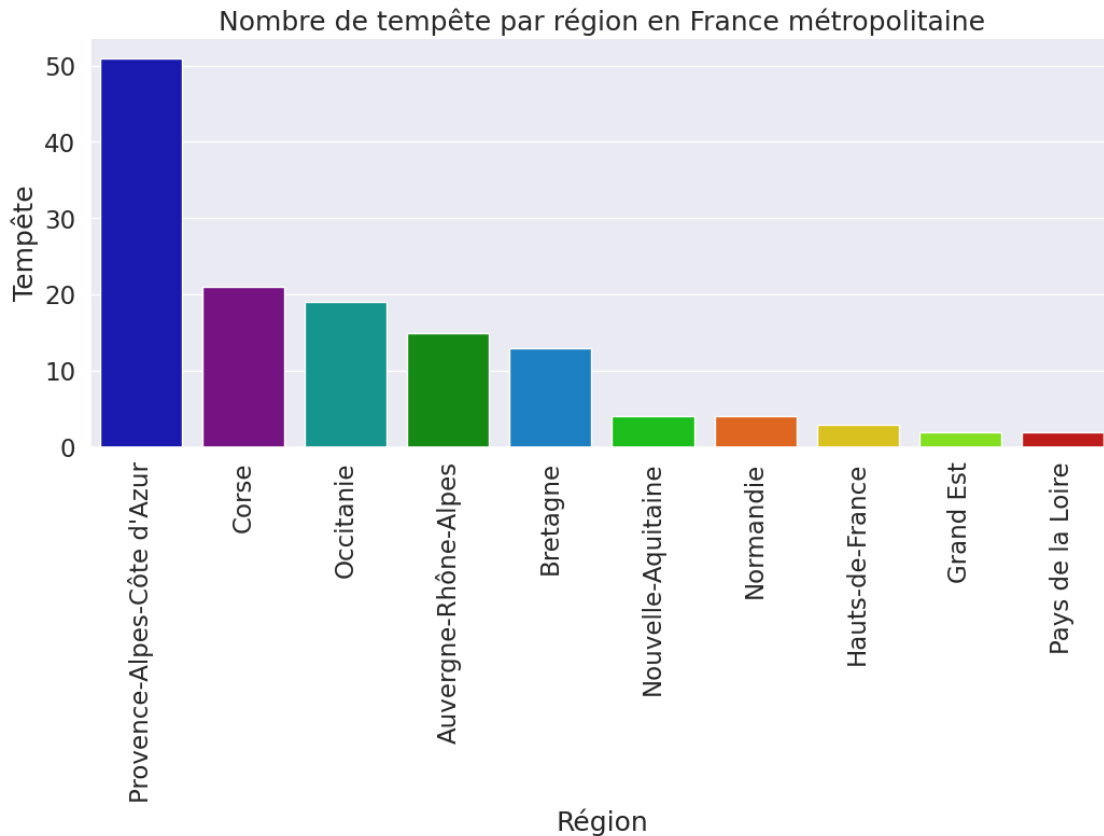
```
[58]: cond1 = tempetes["region (name)"] == "Mayotte"
      cond2 = tempetes["region (name)"] == "Martinique"
      cond3 = tempetes["region (name)"] == "Guadeloupe"
      France_metro = tempetes[~(cond1|cond2|cond3)]
      ordered_genres2= France_metro["region (name)"].value_counts().index.tolist()
      France_metro.groupby("region (name)").size()
```

```
[58]: region (name)
      Auvergne-Rhône-Alpes      15
      Bretagne                  13
      Corse                     21
      Grand Est                  2
      Hauts-de-France           3
      Normandie                 4
      Nouvelle-Aquitaine        4
      Occitanie                 19
      Pays de la Loire          2
      Provence-Alpes-Côte d'Azur 51
      dtype: int64
```

```
[59]: plt.figure(figsize=(10,10))
      sns.catplot(x="region (name)", kind="count", data= France_metro,hue="region_
      ↪(name)", legend=False, order=ordered_genres2, height = 5, aspect = 2.3,palette_
      ↪= 'nipy_spectral')
      plt.xlabel('Région')
      plt.ylabel('Tempête')
      plt.xticks(rotation = 90)
      plt.title("Nombre de tempête par région en France métropolitaine")
```

```
[59]: Text(0.5, 1.0, 'Nombre de tempête par région en France métropolitaine')
```

<Figure size 1000x1000 with 0 Axes>



Ainsi, on remarque cette fois-ci que les 5 régions qui subissent le plus de tempêtes en France métropolitaine depuis 2010 sont : **Provence-Alpes-Côte d'Azur, Corse, Occitanie, Auvergne-Rhône-Alpes et Bretagne**. Le résultat sur ce top 5 semblent cohérent mise à part pour la région **PACA**.

Ainsi, nous pouvons répondre à la **problématique**. En effet, d'après ce jeu de données nous pouvons affirmer que en France métropolitaine les **5 régions** ayant eu le plus de tempêtes depuis 2010 sont les suivantes : **Provence-Alpes-Côte d'Azur, Corse, Occitanie, Auvergne-Rhône-Alpes et Bretagne**.

Maintenant que nous avons répondu à la problématique il serait intéressant de développer en se demandant **si ce top 5 des régions restera le même d'ici 2030 ou va-t-il changer ?**

7 6. Prédiction du Top 5 en 2030

L'objectif maintenant est d'étudier plus en profondeur l'apparition de ces tempêtes pour pouvoir potentiellement prédire si le nombre de tempêtes augmentent ou pas.

Nous allons tout d'abord réduire le DataFrame en **supprimant toutes les mesures qui ne sont pas liées à ces 5 régions**. Je décide de ne pas étudier les **régions outre-mer** étant donné que le **climat est différent** et donc que les caractéristiques de tempête sont différentes.

```
[60]: condition1 = France_metro["region (name)"] == "Nouvelle-Aquitaine"
condition2 = France_metro["region (name)"] == "Normandie"
condition3 = France_metro["region (name)"] == "Hauts-de-France"
condition4 = France_metro["region (name)"] == "Grand Est"
condition5 = France_metro["region (name)"] == "Pays de la Loire"
top5_regions =
↳ France_metro[~(condition1|condition2|condition3|condition4|condition5)]
top5_regions.groupby("region (name)").size()
```

```
[60]: region (name)
Auvergne-Rhône-Alpes      15
Bretagne                  13
Corse                     21
Occitanie                 19
Provence-Alpes-Côte d'Azur 51
dtype: int64
```

Nous avons maintenant que les mesures liées aux **5 régions** où les tempêtes sont les plus récurrentes.

Je vais tout d'abord vérifier que chaque mesure correspond bien à une tempête, c'est-à-dire que nous n'avons pas **2 mesures pour la même région où l'intervalle de temps est de 3 heures**.

```
[61]: time = top5_regions[top5_regions["region (name)"] == "Corse"]
time.head(n=21)[["mois_de_l_annee", "region (name)"]]
```

```
[61]:
```

	mois_de_l_annee	region (name)
Date		
2010-06-15 18:00:00+00:00	Juin	Corse
2011-11-05 03:00:00+00:00	Novembre	Corse
2012-09-04 18:00:00+00:00	Septembre	Corse
2012-10-01 12:00:00+00:00	Octobre	Corse
2013-11-04 18:00:00+00:00	Novembre	Corse
2014-07-26 15:00:00+00:00	Juillet	Corse
2015-03-16 21:00:00+00:00	Mars	Corse
2015-10-01 18:00:00+00:00	Octobre	Corse
2015-10-02 03:00:00+00:00	Octobre	Corse
2016-02-10 09:00:00+00:00	Février	Corse
2017-03-06 21:00:00+00:00	Mars	Corse
2018-01-17 06:00:00+00:00	Janvier	Corse
2018-10-16 12:00:00+00:00	Octobre	Corse
2018-11-01 00:00:00+00:00	Novembre	Corse
2018-11-01 03:00:00+00:00	Novembre	Corse
2019-10-24 12:00:00+00:00	Octobre	Corse
2019-11-02 15:00:00+00:00	Novembre	Corse
2019-11-23 06:00:00+00:00	Novembre	Corse
2020-08-31 12:00:00+00:00	Août	Corse
2022-11-22 03:00:00+00:00	Novembre	Corse
2023-06-01 12:00:00+00:00	Juin	Corse

On remarque pour la **Corse** qu'il y a deux mesures prise avec un intervalle de temps de **3 heures**, je suppose donc qu'il s'agit de la même tempête. Il s'agit des mesures datant du **1er Novembre 2018 à 1 heure et 4 heures**.

Je vais donc supprimer une de ces 2 mesures. Je choisis de supprimer la mesure prise à 4 heures.

```
[62]: index_a_supprimer = pd.to_datetime('2018-11-01 04:00:00+01:00')
top5_regions = top5_regions.drop(index_a_supprimer)
top5_regions.head(n=21)[["mois_de_l_annee", "region (name)"]]
```

```
[62]:
```

	mois_de_l_annee	region (name)
Date		
2010-06-15 18:00:00+00:00	Juin	Corse
2010-08-14 15:00:00+00:00	Août	Provence-Alpes-Côte d'Azur
2010-09-06 09:00:00+00:00	Septembre	Bretagne
2010-09-07 00:00:00+00:00	Septembre	Occitanie
2010-10-02 18:00:00+00:00	Octobre	Bretagne
2010-10-25 09:00:00+00:00	Octobre	Provence-Alpes-Côte d'Azur
2010-10-25 12:00:00+00:00	Octobre	Provence-Alpes-Côte d'Azur
2010-10-30 12:00:00+00:00	Octobre	Occitanie
2010-10-31 12:00:00+00:00	Octobre	Provence-Alpes-Côte d'Azur
2010-10-31 15:00:00+00:00	Octobre	Provence-Alpes-Côte d'Azur
2010-11-20 18:00:00+00:00	Novembre	Provence-Alpes-Côte d'Azur
2011-01-10 12:00:00+00:00	Janvier	Provence-Alpes-Côte d'Azur
2011-06-04 21:00:00+00:00	Juin	Occitanie
2011-10-24 09:00:00+00:00	Octobre	Bretagne
2011-10-25 03:00:00+00:00	Octobre	Provence-Alpes-Côte d'Azur
2011-11-04 06:00:00+00:00	Novembre	Auvergne-Rhône-Alpes
2011-11-04 18:00:00+00:00	Novembre	Occitanie
2011-11-05 03:00:00+00:00	Novembre	Corse
2011-12-16 00:00:00+00:00	Décembre	Bretagne
2011-12-24 15:00:00+00:00	Décembre	Provence-Alpes-Côte d'Azur
2011-12-29 15:00:00+00:00	Décembre	Provence-Alpes-Côte d'Azur

J'ai donc réalisé cette manipulation pour toutes les autres régions afin de supprimer tous les doublons. Des doublons sont apparus uniquement dans la région **Provence-Alpes-Côte d'Azur**, cependant il y avait **7 doublons**, ce qui réduit le nombre de tempête de cette région. Après réflexion, il aurait été bien de faire cette manipulation avant de classer et de définir le top 5 des régions. Or ce problème ne modifie pas le résultat final, mais dans le cas d'une étude plus poussée il faudrait réaliser cette manipulation bien avant.

```
[63]: print(len(top5_regions))
top5_regions.groupby("region (name)").size()
```

118

```
[63]: region (name)
Auvergne-Rhône-Alpes    15
Bretagne                 13
```

```
Corse                20
Occitanie            19
Provence-Alpes-Côte d'Azur  51
dtype: int64
```

Nous avons maintenant **110 mesures** qui correspondent aux nombres de tempêtes qu'il s'est passé dans les 5 régions depuis 2010.

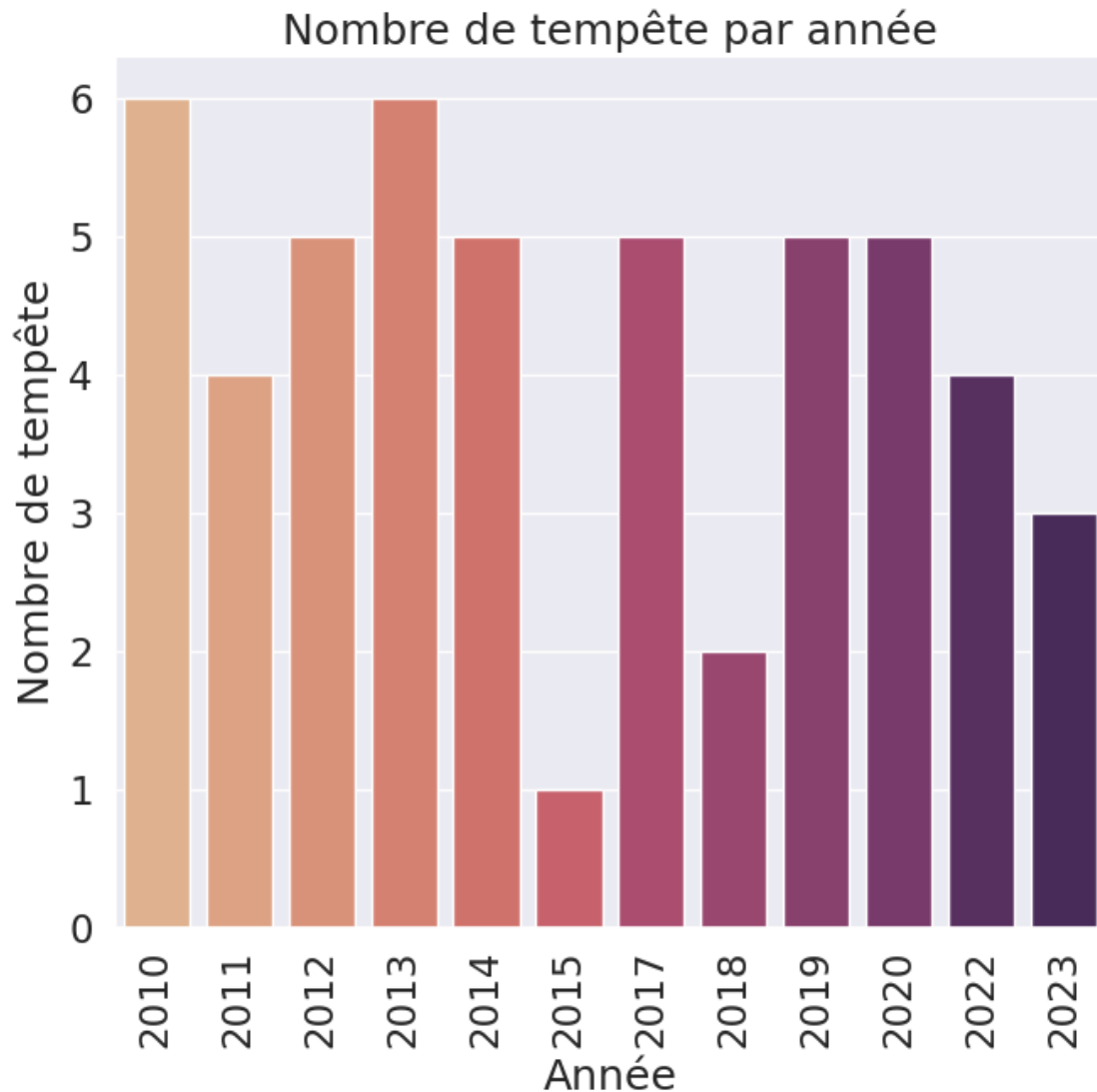
L'**objectif** maintenant est d'afficher pour chaque région l'**évolution temporelle des tempêtes** et de voir si il est possible de réaliser une **prédiction**.

Je commence avec la région **Provence-Alpes-Côte d'Azur** :

Provence-Alpes-Côte d'Azur :

```
[64]: PACA = top5_regions[top5_regions["region (name)"] == "Provence-Alpes-Côte d'Azur"]
mesures = pd.DataFrame(PACA)
mesures.index = pd.to_datetime(mesures.index, utc=True)
mesures['Année'] = mesures.index.year
result = mesures.groupby('Année').size()
```

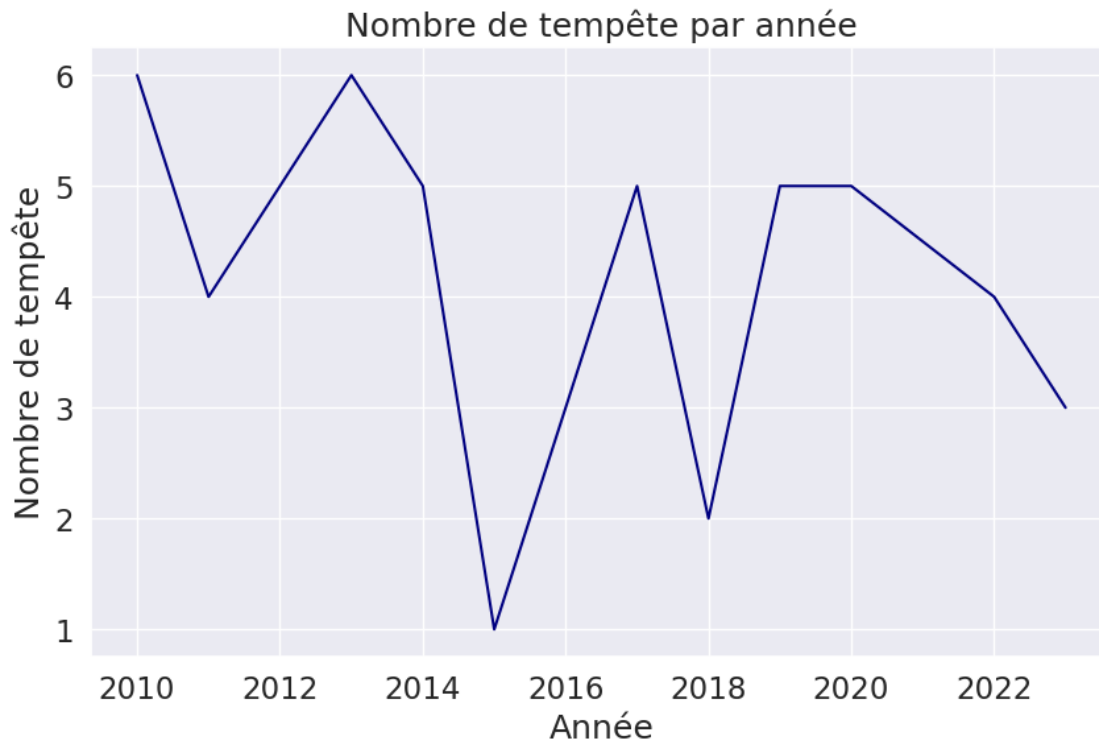
```
[65]: sns.catplot( data=mesures, x='Année',
    kind="count", hue="Année", legend=False, height=7, palette="flare")
plt.xlabel("Année")
plt.ylabel("Nombre de tempête")
plt.title("Nombre de tempête par année")
plt.xticks(rotation = 90)
plt.tight_layout()
plt.show()
```



Ici nous affichons le nombre de tempête qui se sont réalisées pour chaque année dans la région **Provence-Alpes-Côte d’Azur**.

L’affichage des mêmes valeurs à l’aide d’une courbe serait plus facile, nous avons donc la courbe suivante :

```
[66]: plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Nombre de tempête par année')
plt.show()
```



On réalise maintenant une **régression linéaire** sur ce graphe.

Tout d'abord, je crée un DataFrame de ma série **result** qui comporte uniquement le nombre de tempête par année. J'utilise pour ça la fonction suivante :

```
[67]: def create_dataframe(series, column_name):
      df = pd.DataFrame({column_name: series})
      return df
```

```
[68]: df = create_dataframe(result, 'Valeur')
      X = df.index.values.reshape(-1, 1)
      Y = df['Valeur'].values

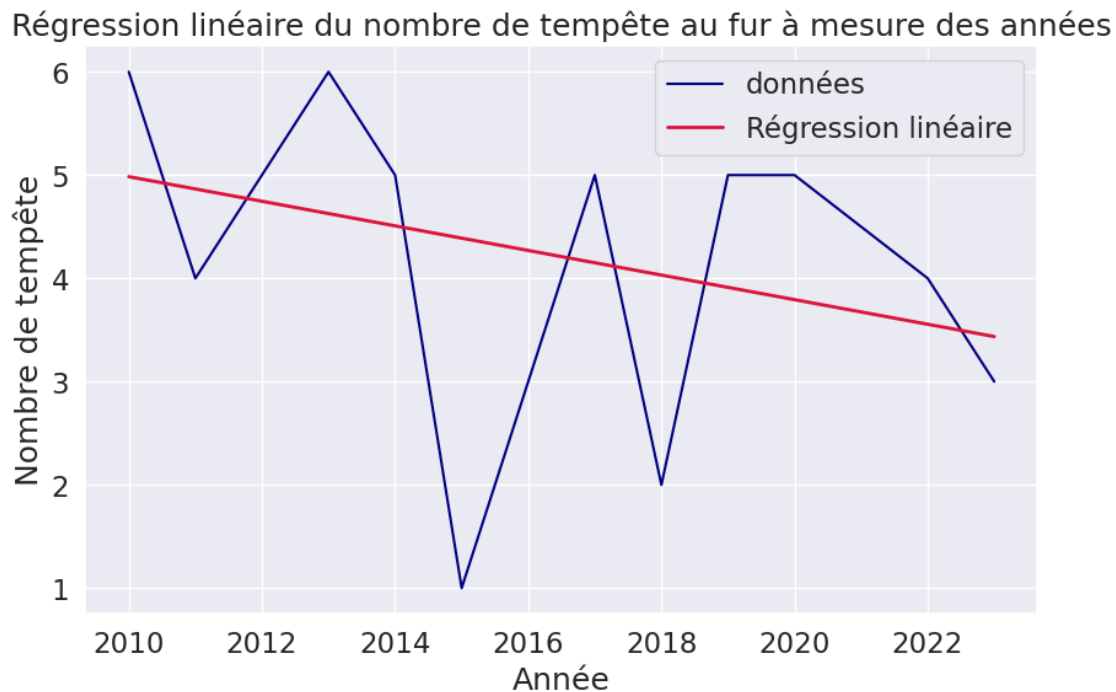
      model = LinearRegression()
      fct = model.fit(X, Y)
      y_pred = model.predict(X)
      print("Estimation : a = %.1f" %fct.coef_[0] + " et b = %.1f" %fct.intercept_)

      plt.figure(figsize=(10, 6))
      result.plot(color = "navy", label = "données")
      plt.plot(X, y_pred, color='crimson', linewidth=2, label='Régression linéaire')
      plt.xlabel('Année')
      plt.ylabel('Nombre de tempête')
```



```
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()
```

Estimation : $a = -0.1$ et $b = 244.4$



Nous avons ici une **régression linéaire** du nombre de tempête au fur à mesure des années pour la région **Provence-Alpes-Côte d’Azur**. On remarque ici que notre régression linéaire est **décroissante**.

Nous allons maintenant chercher à réaliser une **prédiction** du nombre de tempête qu’il y aura dans les prochaines années.

Pour cela je choisis de réaliser une prédiction sur plusieurs années, je vais donc prolonger mon étude jusqu’à **2030**.

```
[69]: X_PACA = [[2030]]
      PACAX = model.predict(X_PACA)
      print(PACAX)
```

```
[2.60210697]
```

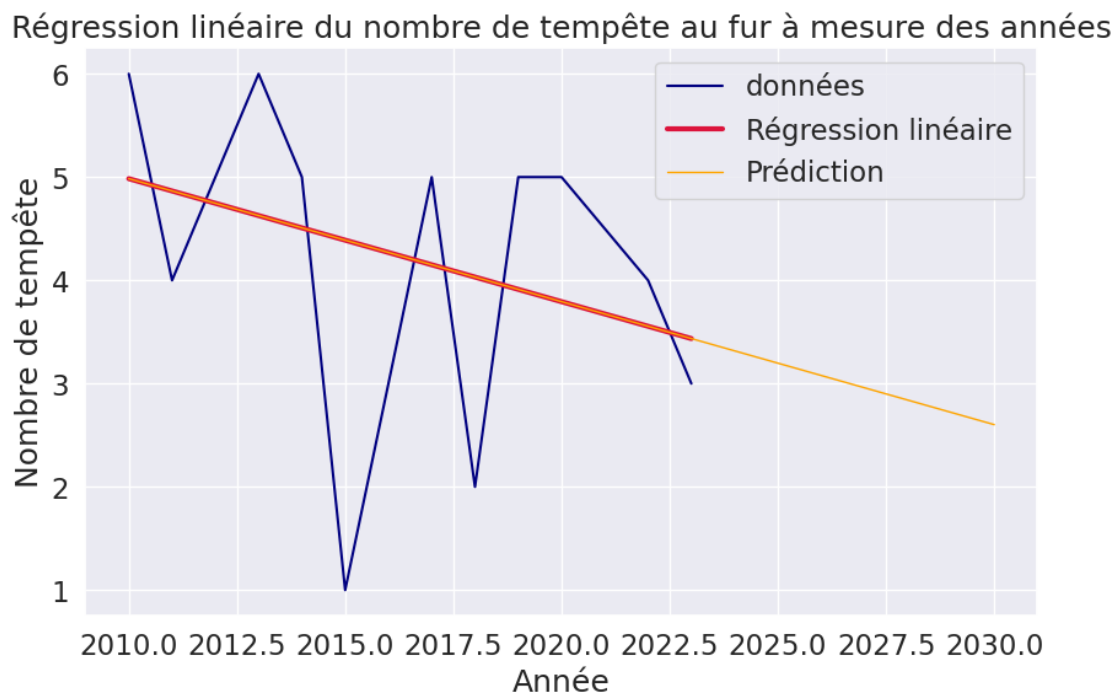
```
[70]: nouvelles_années = np.array([[2024], [2025], [2026], [2027], [2028], [2029],
      ↪ [2030]])
      X2 = np.concatenate((X, nouvelles_années))
```

```

linear_regressor = LinearRegression(copy_X=True, fit_intercept=True)
fct = linear_regressor.fit(X, Y)
y_pred = fct.intercept_ + X*fct.coef_[0]

plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.plot(X, y_pred, color='crimson', linewidth=3, label='Régression linéaire')
plt.plot([np.min(X2), np.max(X2)], [fct.intercept_ + np.min(X2)*fct.coef_[0],
↪fct.intercept_ + np.max(X2)*fct.coef_[0]], linewidth = 1, color='orange',
↪label="Prédiction")
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()

```



```

[71]: r_sq1 = fct.score(X, Y)
print('Coefficient de détermination:', r_sq1)

```

Coefficient de détermination: 0.11118314424635589

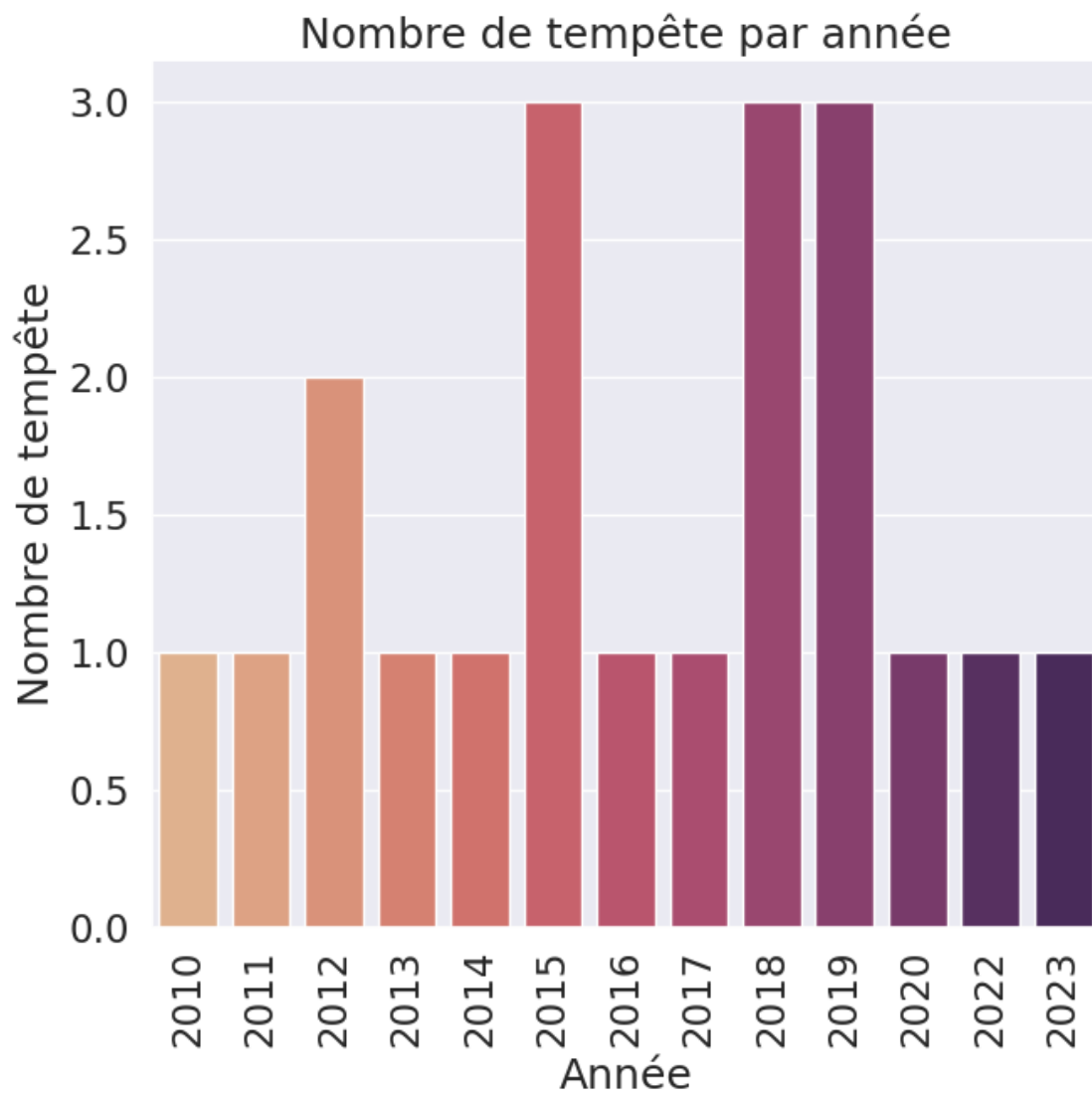
Nous avons ici un coefficient de détermination **plus proche de 0** que de 1. On peut donc supposer que notre prédiction n'est **pas vraiment très représentative** et cela n'est **pas étonnant** lorsque l'on regarde la variation du nombre de tempêtes au fur à mesure des années.

7.0.1 En ce qui concerne les autres régions, il s'agit exactement du même procédé, les seules changements sont les courbes, les régressions linéaires ainsi que les prédictions. Je ne vais donc pas détailler le tout.

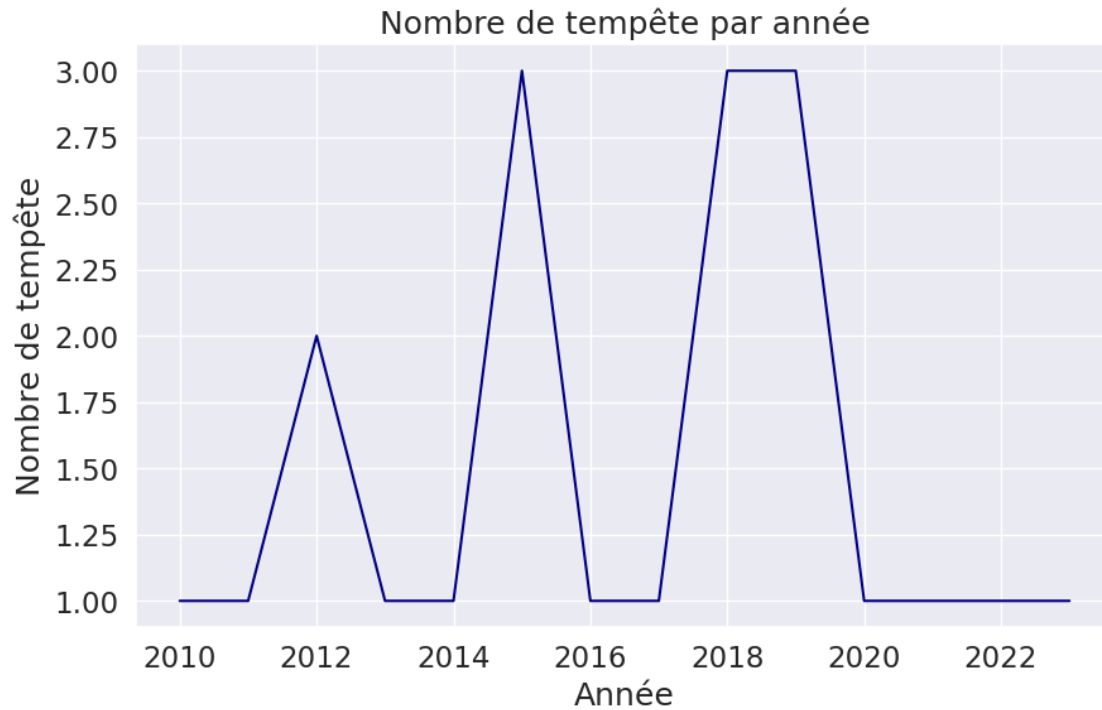
Corse :

```
[72]: Corse = top5_regions[top5_regions["region (name)"] == "Corse"]
mesures = pd.DataFrame(Corse)
mesures.index = pd.to_datetime(mesures.index, utc=True)
mesures['Année'] = mesures.index.year
result = mesures.groupby('Année').size()

[73]: sns.catplot( data=mesures, x='Année',
    ↪kind="count",height=7,hue="Année",legend=False,palette="flare")
plt.xlabel("Année")
plt.ylabel("Nombre de tempête")
plt.title("Nombre de tempête par année")
plt.xticks(rotation = 90)
plt.tight_layout()
plt.show()
```



```
[74]: plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Nombre de tempête par année')
plt.show()
```

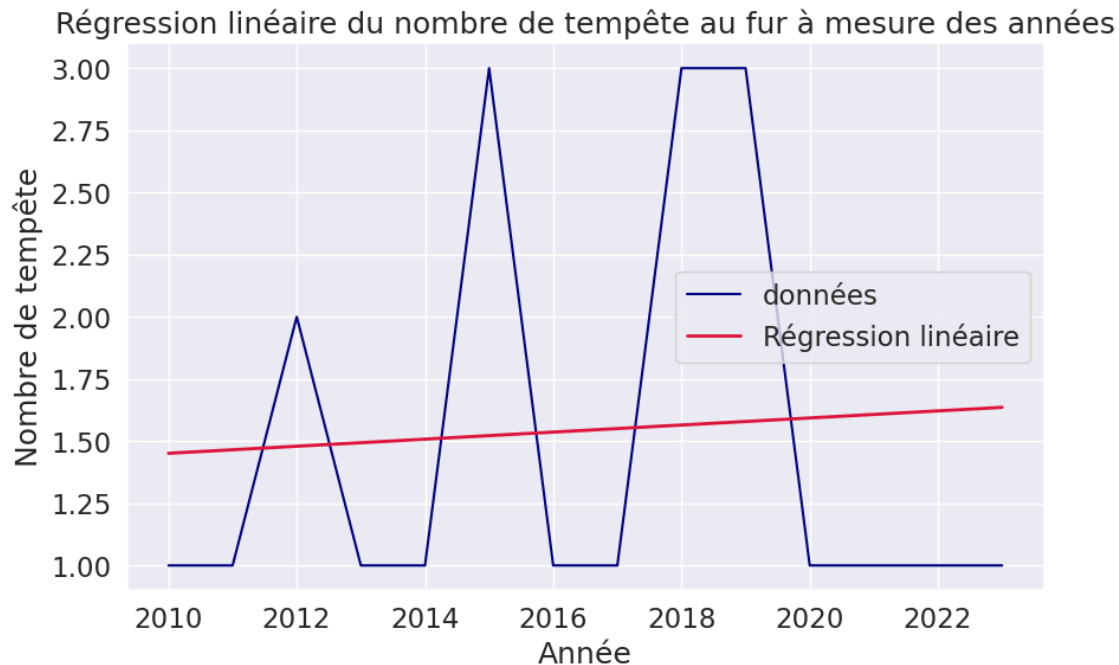


```
[75]: df = create_dataframe(result, 'Valeur')
X = df.index.values.reshape(-1, 1)
Y = df['Valeur'].values

model = LinearRegression()
fct = model.fit(X, Y)
y_pred = model.predict(X)
print("Estimation : a = %.1f" %fct.coef_[0] + " et b = %.1f" %fct.intercept_)

plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.plot(X, y_pred, color='crimson', linewidth=2, label='Régression linéaire')
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()
```

Estimation : a = 0.0 et b = -27.1



```
[76]: X_Cor = [[2030]]
Cor = model.predict(X_Cor)
print(Cor)
```

```
[1.73522812]
```

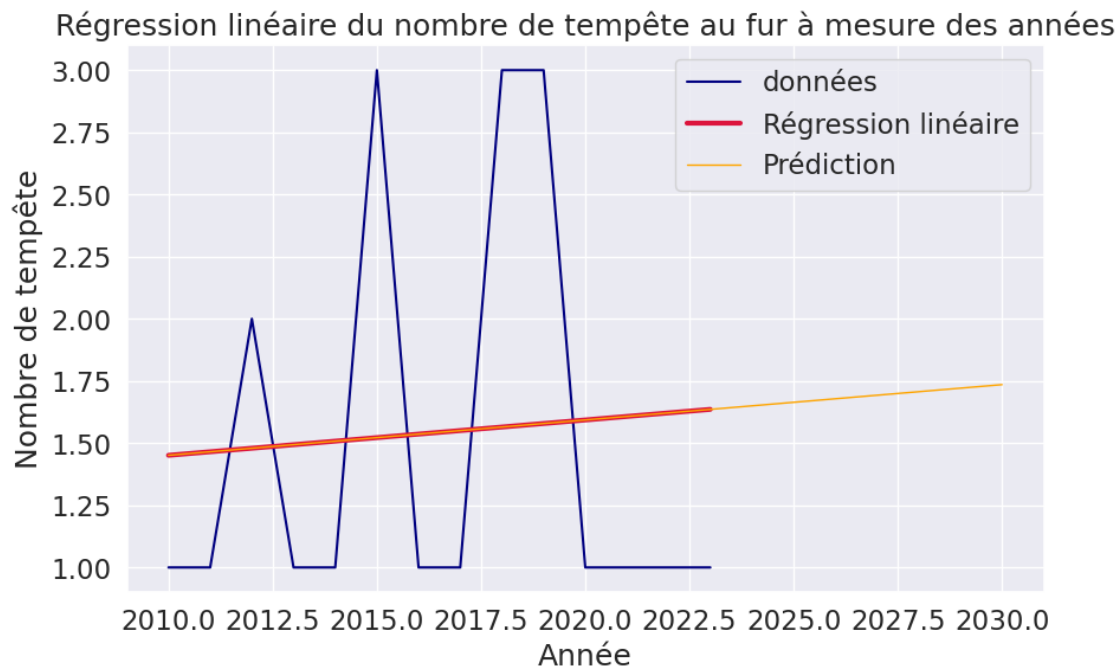
Nous avons ici une **régression linéaire** du nombre de tempête au fur à mesure des années pour la région **Corse**. On remarque ici que notre régression linéaire est **croissante**.

```
[77]: nouvelles_années = np.array([[2024], [2025], [2026], [2027], [2028], [2029],
    ↪ [2030]])
X2 = np.concatenate((X, nouvelles_années))

linear_regressor = LinearRegression(copy_X=True, fit_intercept=True)
fct = linear_regressor.fit(X, Y)
y_pred = fct.intercept_ + X*fct.coef_[0]

plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.plot(X, y_pred, color='crimson', linewidth=3, label='Régression linéaire')
plt.plot([np.min(X2), np.max(X2)], [fct.intercept_ + np.min(X2)*fct.coef_[0],
    ↪ fct.intercept_ + np.max(X2)*fct.coef_[0]], linewidth = 1, color='orange',
    ↪ label="Prédiction")
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
```

```
plt.legend()
plt.show()
```



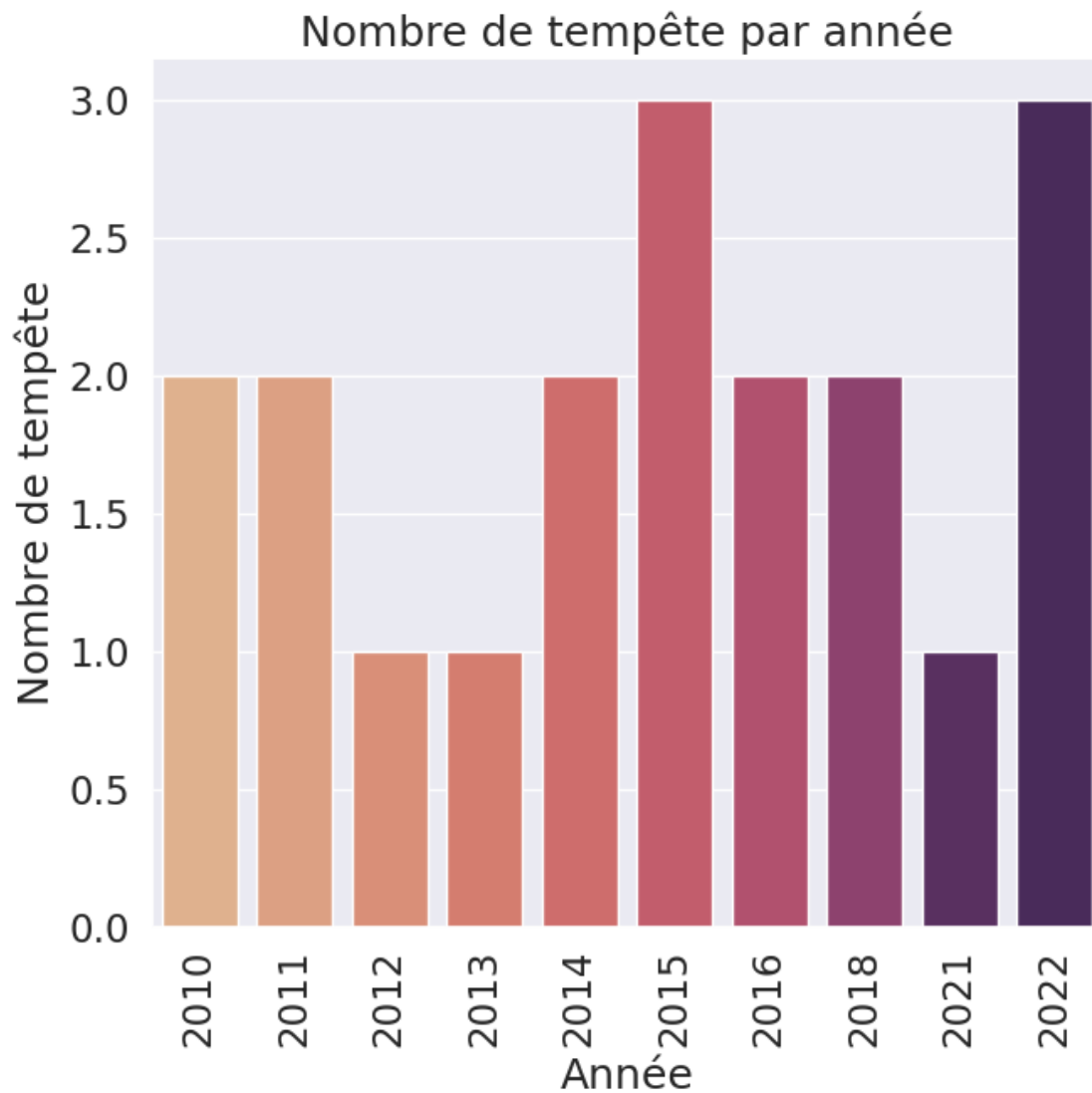
```
[78]: r_sq2 = fct.score(X, Y)
print('Coefficient de détermination:', r_sq2)
```

Coefficient de détermination: 0.004500124657192406

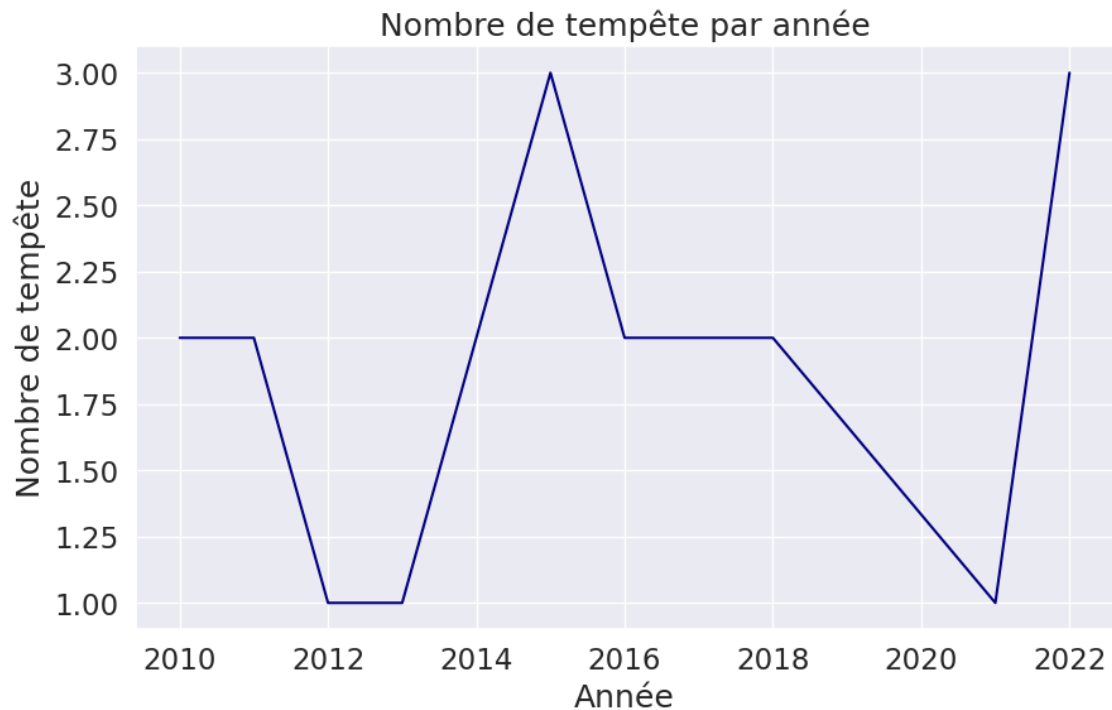
Occitanie :

```
[79]: Occi = top5_regions[top5_regions["region (name)"] == "Occitanie"]
mesures = pd.DataFrame(Occi)
mesures.index = pd.to_datetime(mesures.index, utc=True)
mesures['Année'] = mesures.index.year
result = mesures.groupby('Année').size()
```

```
[80]: sns.catplot( data=mesures, x='Année',
    ↪kind="count",height=7,hue="Année",legend=False, palette="flare")
plt.xlabel("Année")
plt.ylabel("Nombre de tempête")
plt.title("Nombre de tempête par année")
plt.xticks(rotation = 90)
plt.tight_layout()
plt.show()
```



```
[81]: plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Nombre de tempête par année')
plt.show()
```

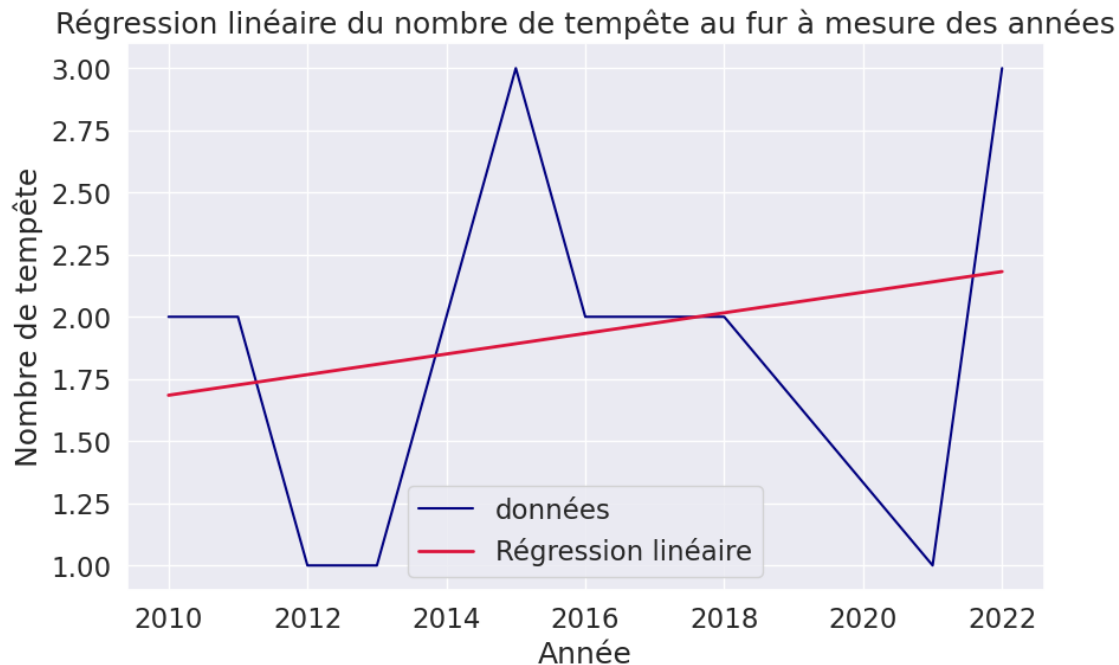



```
[82]: df = create_dataframe(result, 'Valeur')
X = df.index.values.reshape(-1, 1)
Y = df['Valeur'].values

model = LinearRegression()
fct = model.fit(X, Y)
y_pred = model.predict(X)
print("Estimation : a = %.1f" %fct.coef_[0] + " et b = %.1f" %fct.intercept_)

plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.plot(X, y_pred, color='crimson', linewidth=2, label='Régression linéaire')
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()
```

Estimation : a = 0.0 et b = -81.6



```
[83]: X_Occi = [[2030]]
      Occi = model.predict(X_Occi)
      print(Occi)
```

[2.51336898]

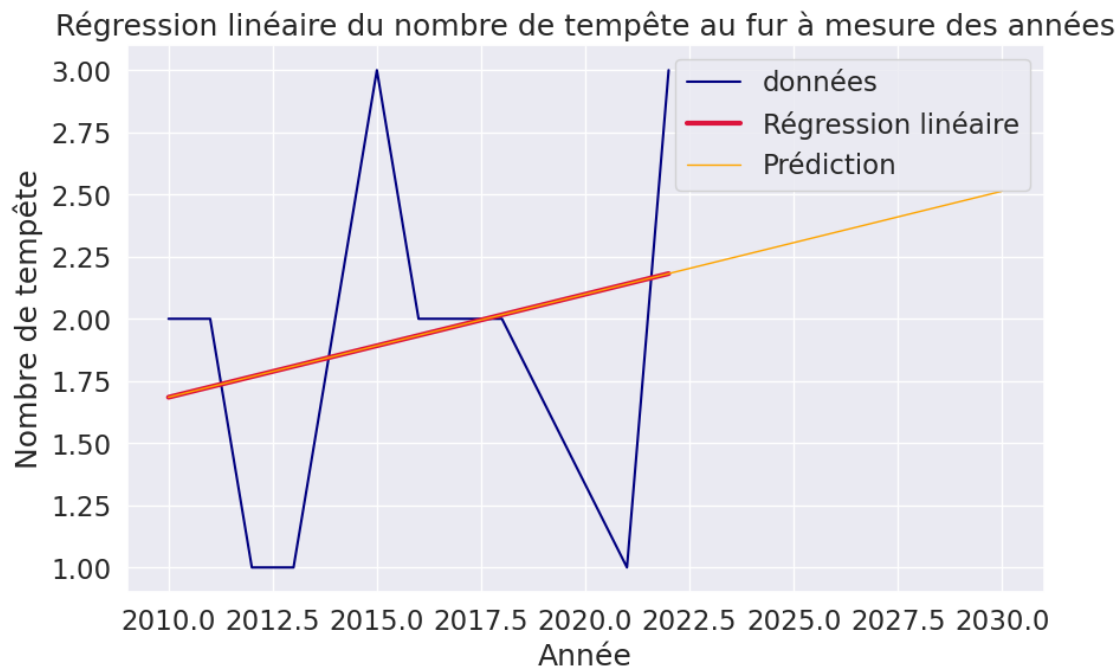
Nous avons ici une **régression linéaire** du nombre de tempête au fur à mesure des années pour la région **Occitanie**. On remarque ici que notre régression linéaire est **croissante**.

```
[84]: nouvelles_années = np.array([[2024], [2025], [2026], [2027], [2028], [2029],
      ↪ [2030]])
      X2 = np.concatenate((X, nouvelles_années))

      linear_regressor = LinearRegression(copy_X=True, fit_intercept=True)
      fct = linear_regressor.fit(X, Y)
      y_pred = fct.intercept_ + X*fct.coef_[0]

      plt.figure(figsize=(10, 6))
      result.plot(color = "navy", label = "données")
      plt.plot(X, y_pred, color='crimson', linewidth=3, label='Régression linéaire')
      plt.plot([np.min(X2), np.max(X2)], [fct.intercept_ + np.min(X2)*fct.coef_[0],
      ↪ fct.intercept_ + np.max(X2)*fct.coef_[0]], linewidth = 1, color='orange',
      ↪ label="Prédiction")
      plt.xlabel('Année')
      plt.ylabel('Nombre de tempête')
      plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
```

```
plt.legend()
plt.show()
```



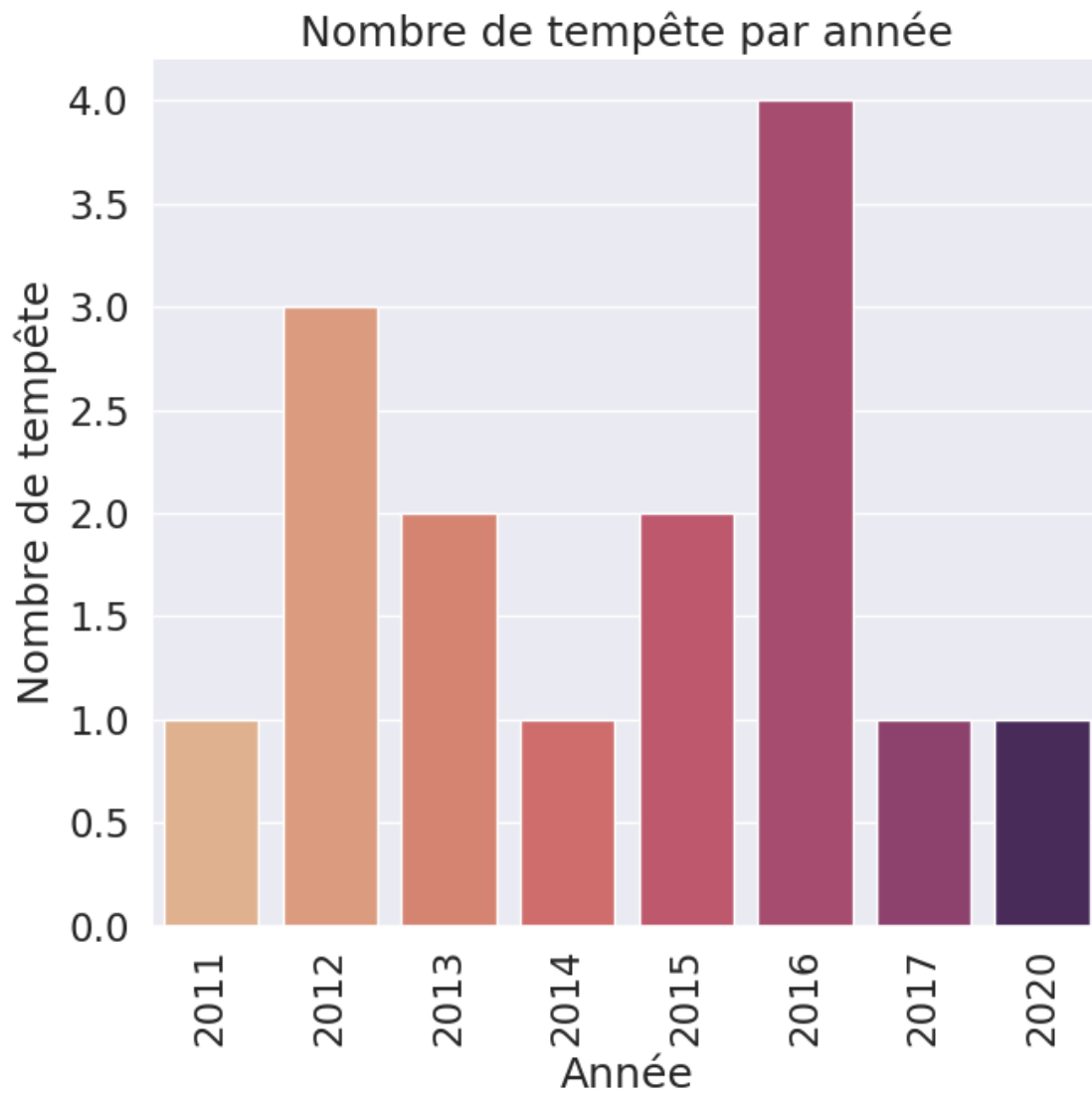
```
[85]: r_sq3 = fct.score(X, Y)
print('Coefficient de détermination:', r_sq3)
```

Coefficient de détermination: 0.052439157481172116

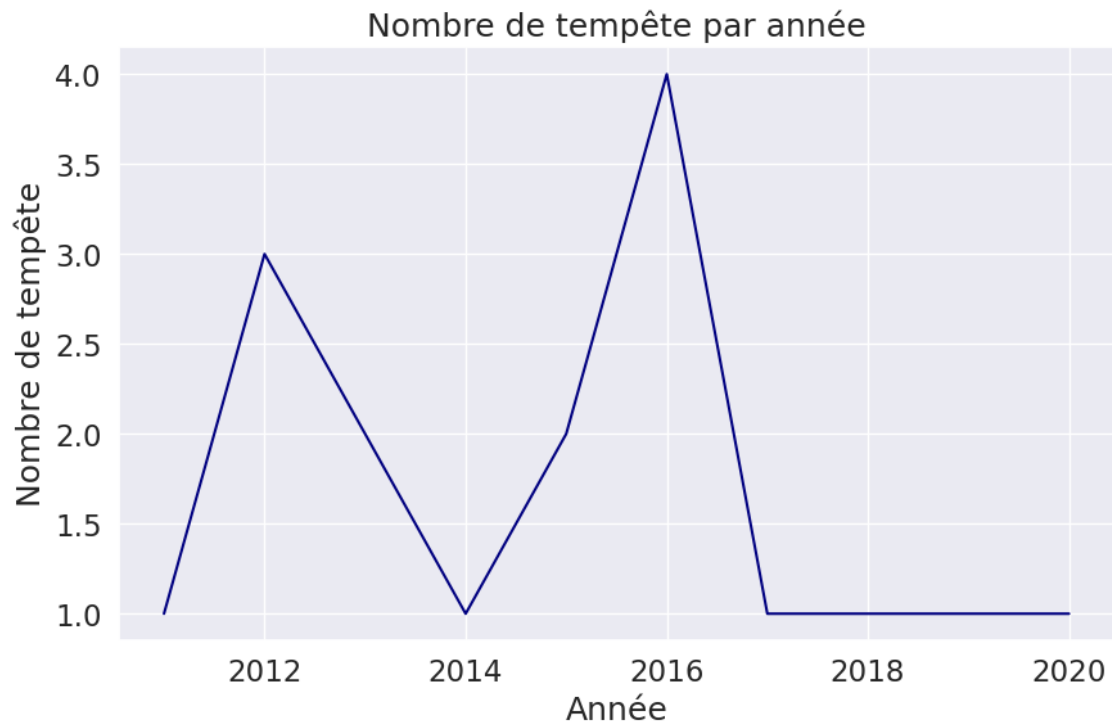
Auvergne-Rhône-Alpes :

```
[86]: ARA = top5_regions[top5_regions["region (name)"] == "Auvergne-Rhône-Alpes"]
mesures = pd.DataFrame(ARA)
mesures.index = pd.to_datetime(mesures.index, utc=True)
mesures['Année'] = mesures.index.year
result = mesures.groupby('Année').size()
```

```
[87]: sns.catplot( data=mesures, x='Année',
    ↪kind="count",height=7,hue="Année",legend=False, palette="flare")
plt.xlabel("Année")
plt.ylabel("Nombre de tempête")
plt.title("Nombre de tempête par année")
plt.xticks(rotation = 90)
plt.tight_layout()
plt.show()
```



```
[88]: plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Nombre de tempête par année')
plt.show()
```

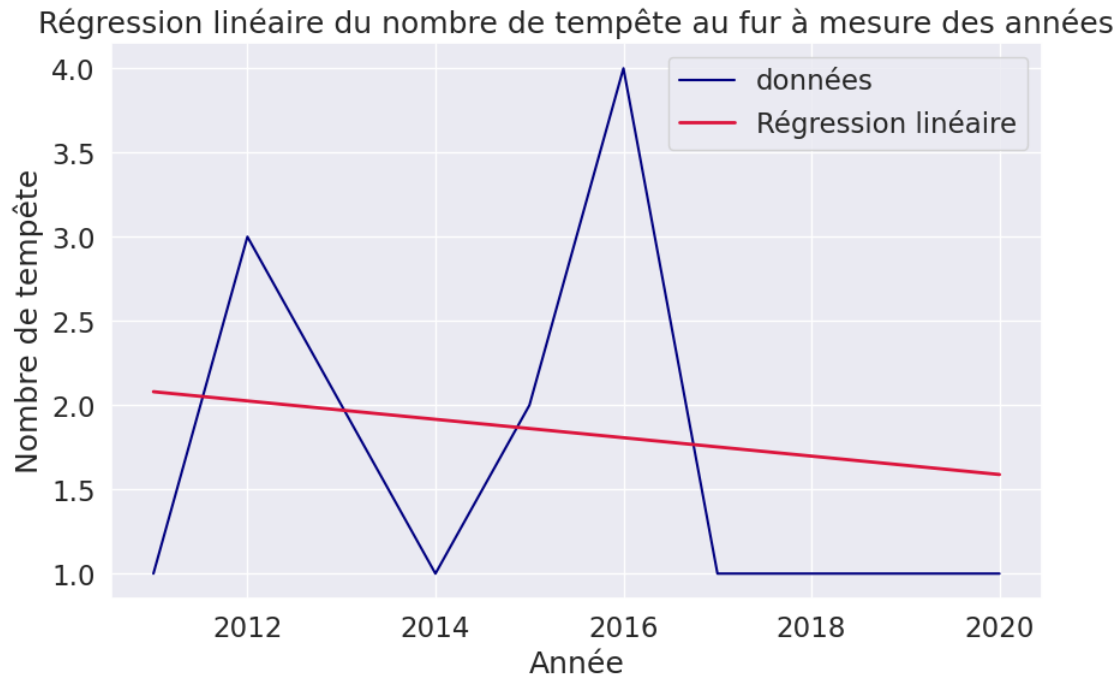


```
[89]: df = create_dataframe(result, 'Valeur')
X = df.index.values.reshape(-1, 1)
Y = df['Valeur'].values

model = LinearRegression()
fct = model.fit(X, Y)
y_pred = model.predict(X)
print("Estimation : a = %.1f" %fct.coef_[0] + " et b = %.1f" %fct.intercept_)

plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.plot(X, y_pred, color='crimson', linewidth=2, label='Régression linéaire')
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()
```

Estimation : a = -0.1 et b = 111.9



```
[90]: X_ARA = [[2030]]
      ARA = model.predict(X_ARA)
      print(ARA)
```

```
[1.04201681]
```

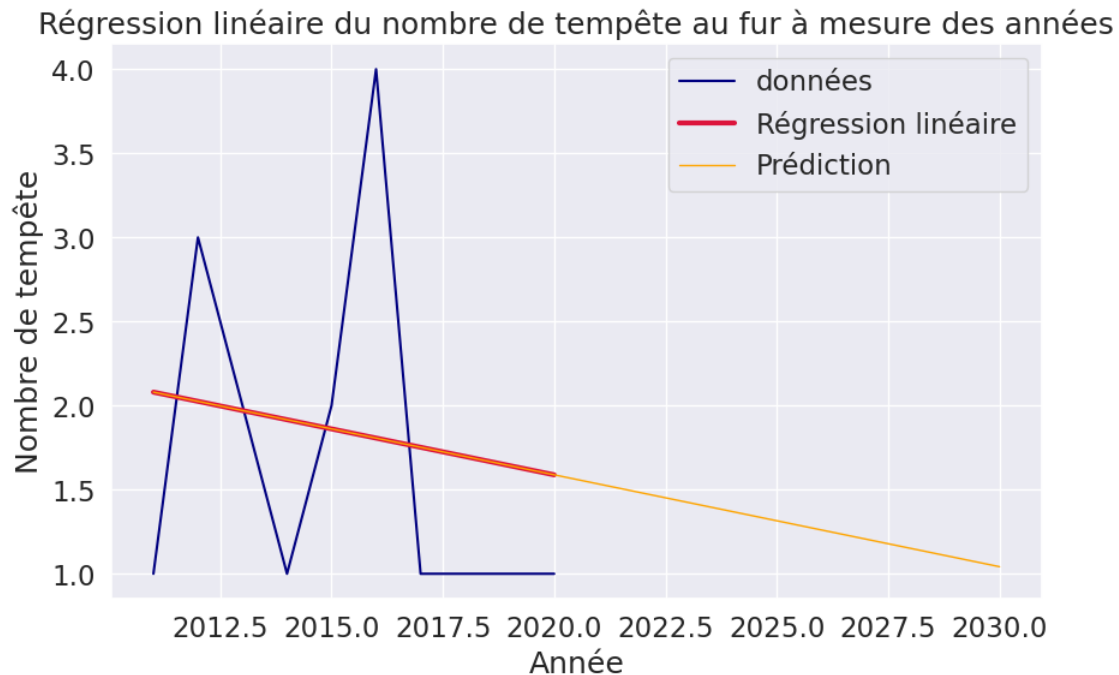
Nous avons ici une **régression linéaire** du nombre de tempête au fur à mesure des années pour la région **Auvergne-Rhône-Alpes**. On remarque ici que notre régression linéaire est **décroissante**.

```
[91]: nouvelles_années = np.array([[2024], [2025], [2026], [2027], [2028], [2029],
      ↪ [2030]])
      X2 = np.concatenate((X, nouvelles_années))

      linear_regressor = LinearRegression(copy_X=True, fit_intercept=True)
      fct = linear_regressor.fit(X, Y)
      y_pred = fct.intercept_ + X*fct.coef_[0]

      plt.figure(figsize=(10, 6))
      result.plot(color = "navy", label = "données")
      plt.plot(X, y_pred, color='crimson', linewidth=3, label='Régression linéaire')
      plt.plot([np.min(X2), np.max(X2)], [fct.intercept_ + np.min(X2)*fct.coef_[0],
      ↪ fct.intercept_ + np.max(X2)*fct.coef_[0]], linewidth = 1, color='orange',
      ↪ label="Prédiction")
      plt.xlabel('Année')
      plt.ylabel('Nombre de tempête')
```

```
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()
```



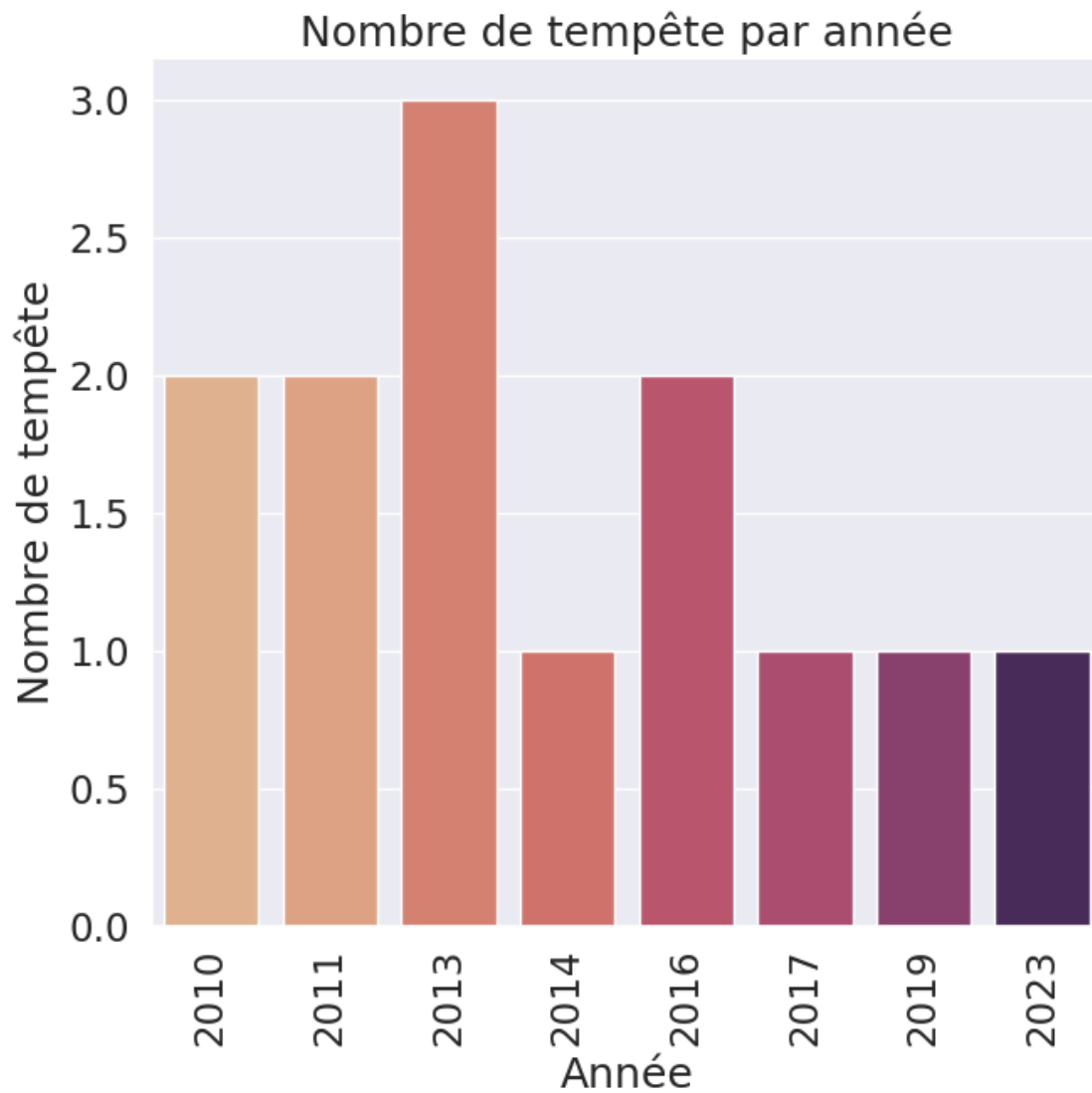
```
[92]: r_sq4 = fct.score(X, Y)
print('Coefficient de determination:', r_sq4)
```

Coefficient de determination: 0.0200023671440418

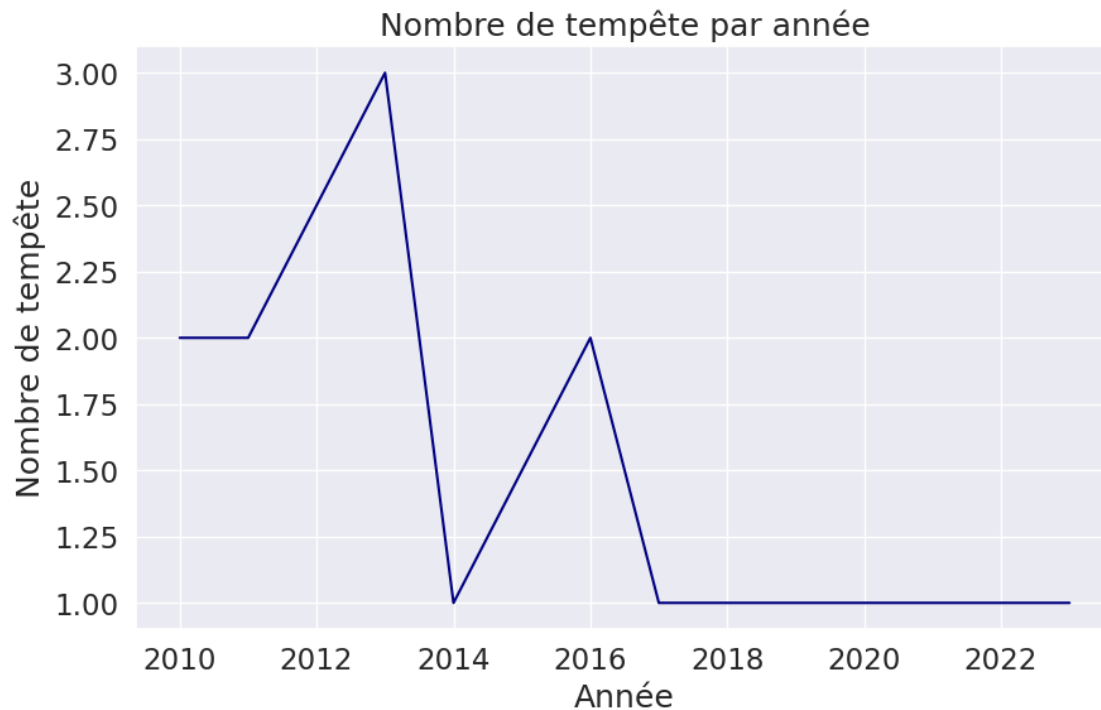
Bretagne :

```
[93]: Bret = top5_regions[top5_regions["region (name)"] == "Bretagne"]
mesures = pd.DataFrame(Bret)
mesures.index = pd.to_datetime(mesures.index, utc=True)
mesures['Année'] = mesures.index.year
result = mesures.groupby('Année').size()
```

```
[94]: sns.catplot( data=mesures, x='Année',
    ↪kind="count",height=7,hue="Année",legend=False, palette="flare")
plt.xlabel("Année")
plt.ylabel("Nombre de tempête")
plt.title("Nombre de tempête par année")
plt.xticks(rotation = 90)
plt.tight_layout()
plt.show()
```



```
[95]: plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Nombre de tempête par année')
plt.show()
```

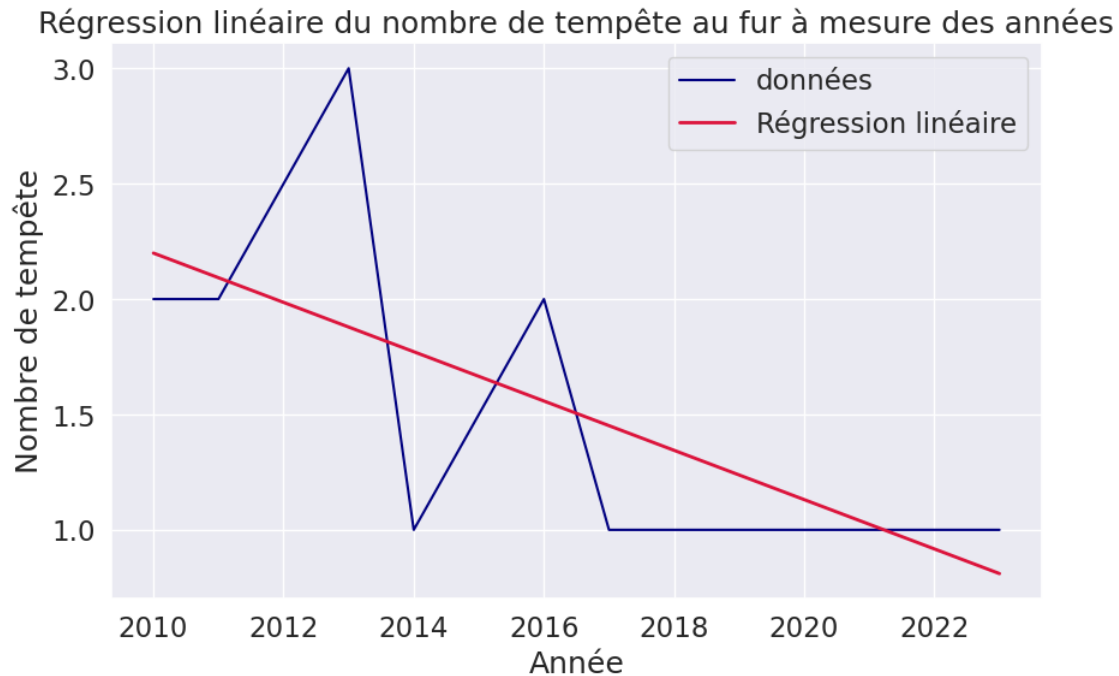



```
[96]: df = create_dataframe(result, 'Valeur')
X = df.index.values.reshape(-1, 1)
Y = df['Valeur'].values

model = LinearRegression()
fct = model.fit(X, Y)
y_pred = model.predict(X)
print("Estimation : a = %.1f" %fct.coef_[0] + " et b = %.1f" %fct.intercept_)

plt.figure(figsize=(10, 6))
result.plot(color = "navy", label = "données")
plt.plot(X, y_pred, color='crimson', linewidth=2, label='Régression linéaire')
plt.xlabel('Année')
plt.ylabel('Nombre de tempête')
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()
```

Estimation : a = -0.1 et b = 216.9



```
[97]: X_Bret = [[2030]]
      Bret = model.predict(X_Bret)
      print(Bret)
```

```
[0.06256015]
```

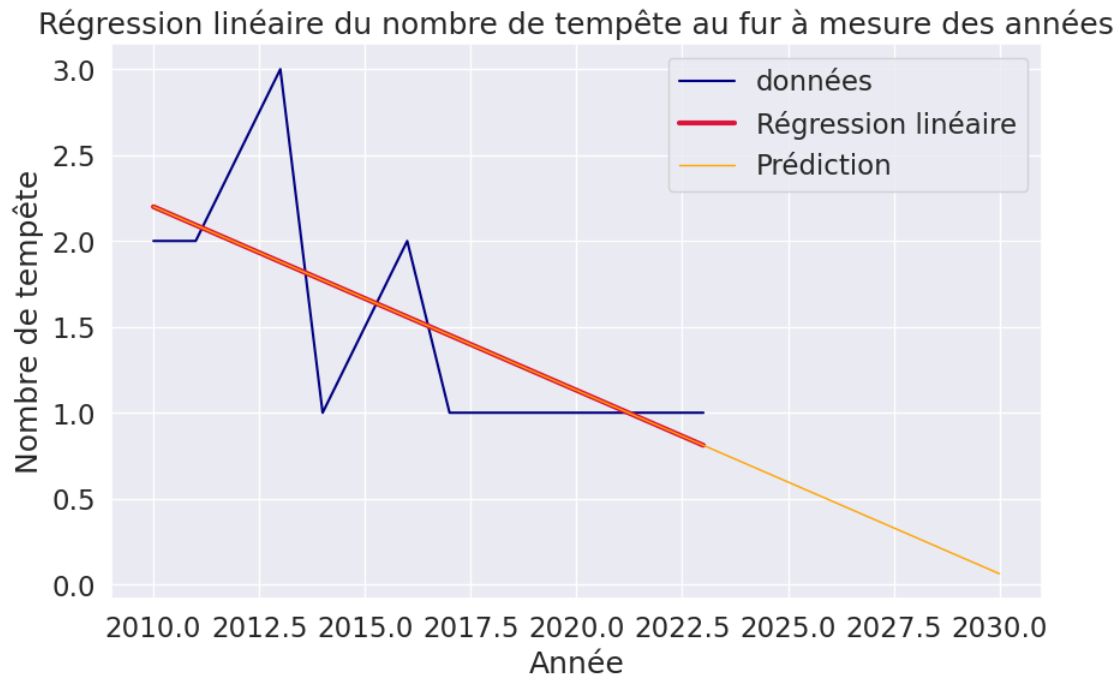
Nous avons ici une **régression linéaire** du nombre de tempête au fur à mesure des années pour la région **Bretagne**. On remarque ici que notre régression linéaire est **décroissante**.

```
[98]: nouvelles_années = np.array([[2024], [2025], [2026], [2027], [2028], [2029],
      ↪ [2030]])
      X2 = np.concatenate((X, nouvelles_années))

      linear_regressor = LinearRegression(copy_X=True, fit_intercept=True)
      fct = linear_regressor.fit(X, Y)
      y_pred = fct.intercept_ + X*fct.coef_[0]

      plt.figure(figsize=(10, 6))
      result.plot(color = "navy", label = "données")
      plt.plot(X, y_pred, color='crimson', linewidth=3, label='Régression linéaire')
      plt.plot([np.min(X2), np.max(X2)], [fct.intercept_ + np.min(X2)*fct.coef_[0],
      ↪ fct.intercept_ + np.max(X2)*fct.coef_[0]], linewidth = 1, color='orange',
      ↪ label="Prédiction")
      plt.xlabel('Année')
      plt.ylabel('Nombre de tempête')
```

```
plt.title('Régression linéaire du nombre de tempête au fur à mesure des années')
plt.legend()
plt.show()
```



```
[99]: r_sq5 = fct.score(X, Y)
print('Coefficient de determination:', r_sq5)
```

Coefficient de determination: 0.3825328324381294

8 7. Conclusion

Pour conclure, en ce qui concerne notre problématique initiale nous avons pu déterminer le **Top 5 des régions** qui ont été le plus touchées par les tempêtes depuis 2010 en France métropolitaine. En effet, ce Top 5 est le suivant : **Provence-Alpes-Côte d’Azur, Corse, Occitanie, Auvergne-Rhône-Alpes et Bretagne.**

En ce qui concerne ce résultat, il semble plutôt cohérent. Une question se pose sur la région **PACA**, ce résultat reste surprenant, peut-être qu’avec un peu plus de recherche ce résultat serait explicable.

En ce qui concerne maintenant la **seconde question** sur la prédiction, nous avons un résultat de même **assez surprenant**. En effet, la théorie dit qu’avec le réchauffement climatique le nombre de tempête est censé **augmenter au cours des années**. Or parmi nos prédictions seulement **2** confirment cette hypothèse tandis que les autres prédisent que le **nombre de tempêtes diminue**.

```
[100]: dico_coef = {"PACA": r_sq1, "Corse": r_sq2, "Occi": r_sq3, "ARA": r_sq4, "Breizh": r_sq5}
```

```
print("Ce dictionnaire représente l'ensemble des coefficient de determination,
↳pour chaque région :")
dico_coef
```

Ce dictionnaire représente l'ensemble des coefficient de determination pour chaque région :

```
[100]: {'PACA': 0.11118314424635589,
        'Corse': 0.004500124657192406,
        'Occi': 0.052439157481172116,
        'ARA': 0.0200023671440418,
        'Breizh': 0.3825328324381294}
```

Cependant, on remarque que la plupart de nos coefficient de détermination sont très proches de **0 voir égaux**. Ainsi, on peut remettre très facilement en question nos prédictions et cette remise en cause est logique lorsque l'on voit **l'hétérogénéité du nombre de tempêtes au cours des années pour chaque région**.

```
[101]: dico_predi = {
        "PACA": float(PACAX.item() if hasattr(PACAX, "item") else PACAX),
        "Corse": float(Cor.item() if hasattr(Cor, "item") else Cor),
        "Occi": float(Occi.item() if hasattr(Occi, "item") else Occi),
        "ARA": float(ARA.item() if hasattr(ARA, "item") else ARA),
        "Breizh": float(Bret.item() if hasattr(Bret, "item") else Bret)
    }

print("Ce dictionnaire représente l'ensemble des prédictions en 2030 pour chaque,
↳région :")
dico_predi
```

Ce dictionnaire représente l'ensemble des prédictions en 2030 pour chaque région :

```
[101]: {'PACA': 2.6021069692058347,
        'Corse': 1.7352281226626793,
        'Occi': 2.5133689839572213,
        'ARA': 1.0420168067226996,
        'Breizh': 0.06256015399424086}
```

Afin de répondre quand même à la **seconde problématique**, on peut grâce aux prédictions effectuer, affirmer que le **Top 5 des régions en 2030** changera et sera le suivant : **Provence-Alpes-Côte d'Azur, Occitanie, Corse, Auvergne-Rhône-Alpes et Bretagne**. Finalement, en 2030, la région **Occitanie** aura plus de tempêtes que la région **Corse**.

9 Références

[1] : Définitions - Tempêtes en France métropolitaine. (s. d.).
<http://tempetes.meteo.fr/Definitions.html#:~:text=L'appellation%20temp%C3%AAt%20est%20donc,sup%C3%A9rieur%20%C3%A0%201%20temp%C3%AAt%20par%20an>

- [2] : Dubault, F. (2020, 10 juin). Un record historique de pluie à Montpellier et des millions de dégâts dans l'Hérault. France 3 Occitanie. <https://france3-regions.francetvinfo.fr/occitanie/2014/09/30/un-record-historique-de-pluie-montpellier-et-des-millions-de-degats-dans-l-herault-561820.html>
- [3] : Vents violents et tempêtes : quelles sont les différences ? | Météo-France. (s. d.). <https://meteofrance.com/actualites-et-dossiers/comprendre-la-meteo/le-vent/vents-violents-et-tempetes-queelles-sont-les-differences>
- [4] : Les vents régionaux méditerranéens - Tempêtes en France métropolitaine. (s. d.). <http://tempetes.meteo.fr/Les-vents-regionaux-mediterraneens.html#:~:text=La%20rose%20des%20vents%20%C%A0%20Marignane%20>
- [5] : Quelle est la température la plus élevée enregistrée en France ? | Météo-France. (s. d.). <https://meteofrance.com/magazine/meteo-questions/quelle-est-la-temperature-la-plus-elevee-enregistree-en-france>
- [6] : Jérôme, D. (2023, 24 octobre). Impacts du changement climatique : Atmosphère, Températures et Précipitations. Ministère de la Transition Écologique et de la Cohésion des Territoires. <https://www.ecologie.gouv.fr/impacts-du-changement-climatique-atmosphere-temperatures-et-precipitations>