

# Meteo\_Data\_Cleaning

January 22, 2026

```
[1]: import pandas as pd
```

## 0.0.1 Importation des données

```
[2]: data = pd.read_csv("london_merged.csv")
shape = data.shape
print(f'the meteo data contains {shape[0]} rows, {shape[1]} columns')
print(f'with the columns being {list(data.columns)}')
data.head(10)
```

the meteo data contains 17414 rows, 10 columns

with the columns being ['timestamp', 'cnt', 't1', 't2', 'hum', 'wind\_speed', 'weather\_code', 'is\_holiday', 'is\_weekend', 'season']

```
[2]:
```

	timestamp	cnt	t1	t2	hum	wind_speed	weather_code	\
0	2015-01-04 00:00:00	182	3.0	2.0	93.0	6.0	3.0	
1	2015-01-04 01:00:00	138	3.0	2.5	93.0	5.0	1.0	
2	2015-01-04 02:00:00	134	2.5	2.5	96.5	0.0	1.0	
3	2015-01-04 03:00:00	72	2.0	2.0	100.0	0.0	1.0	
4	2015-01-04 04:00:00	47	2.0	0.0	93.0	6.5	1.0	
5	2015-01-04 05:00:00	46	2.0	2.0	93.0	4.0	1.0	
6	2015-01-04 06:00:00	51	1.0	-1.0	100.0	7.0	4.0	
7	2015-01-04 07:00:00	75	1.0	-1.0	100.0	7.0	4.0	
8	2015-01-04 08:00:00	131	1.5	-1.0	96.5	8.0	4.0	
9	2015-01-04 09:00:00	301	2.0	-0.5	100.0	9.0	3.0	

	is_holiday	is_weekend	season
0	0.0	1.0	3.0
1	0.0	1.0	3.0
2	0.0	1.0	3.0
3	0.0	1.0	3.0
4	0.0	1.0	3.0
5	0.0	1.0	3.0
6	0.0	1.0	3.0
7	0.0	1.0	3.0
8	0.0	1.0	3.0
9	0.0	1.0	3.0

## 0.0.2 Exploration Dataset

```
[3]: for col in data.columns.drop("timestamp"):
    print(f'{col} :')
    unique_values = data[col].unique()
    if(len(unique_values) < 20):
        print(f'la colonne {col} contient les valeurs unique:\n {data[col].
↵unique()}\n')
    else:
        print(f'la colonne {col} contient des valeurs entre:\n {data[col].
↵min()}\n et {data[col].max()}\n')
        print(f'avec un moyenne de {round(data[col].mean(),2)}\n')
```

```
cnt :
la colonne cnt contient des valeurs entre:
 0 et 7860
avec un moyenne de 1143.1
t1 :
la colonne t1 contient des valeurs entre:
-1.5 et 34.0
avec un moyenne de 12.47
t2 :
la colonne t2 contient des valeurs entre:
-6.0 et 34.0
avec un moyenne de 11.52
hum :
la colonne hum contient des valeurs entre:
20.5 et 100.0
avec un moyenne de 72.32
wind_speed :
la colonne wind_speed contient des valeurs entre:
0.0 et 56.5
avec un moyenne de 15.91
weather_code :
la colonne weather_code contient les valeurs unique:
[ 3.  1.  4.  7.  2. 26. 10.]
is_holiday :
la colonne is_holiday contient les valeurs unique:
[0. 1.]
is_weekend :
la colonne is_weekend contient les valeurs unique:
[1. 0.]
season :
la colonne season contient les valeurs unique:
[3. 0. 1. 2.]
```

### 0.0.3 Ni valeurs nulls ni duplications

```
[4]: data.isnull().sum()
```

```
[4]: timestamp      0
     cnt            0
     t1            0
     t2            0
     hum           0
     wind_speed     0
     weather_code   0
     is_holiday     0
     is_weekend     0
     season         0
     dtype: int64
```

```
[5]: print("Number of duplicated lines:",data.duplicated().sum())
```

Number of duplicated lines: 0

### 0.0.4 Changement des noms des colonnes

```
[6]: renaming = {"cnt":"Nombre de trajets","t1":"Température réelle (°C)","t2":
    ↪ "Température ressentie (°C)","hum":"Humidité","wind_speed":"Vitesse du vent",
    ↪ "(km/h)","weather_code":"Météo","season":"Saison"}

     data = data.rename(columns=renaming)
     del renaming
     data.head(10)
```

```
[6]:
```

	timestamp	Nombre de trajets	Température réelle (°C)	\
0	2015-01-04 00:00:00	182	3.0	
1	2015-01-04 01:00:00	138	3.0	
2	2015-01-04 02:00:00	134	2.5	
3	2015-01-04 03:00:00	72	2.0	
4	2015-01-04 04:00:00	47	2.0	
5	2015-01-04 05:00:00	46	2.0	
6	2015-01-04 06:00:00	51	1.0	
7	2015-01-04 07:00:00	75	1.0	
8	2015-01-04 08:00:00	131	1.5	
9	2015-01-04 09:00:00	301	2.0	

  

	Température ressentie (°C)	Humidité	Vitesse du vent (km/h)	Météo	\
0	2.0	93.0	6.0	3.0	
1	2.5	93.0	5.0	1.0	
2	2.5	96.5	0.0	1.0	
3	2.0	100.0	0.0	1.0	
4	0.0	93.0	6.5	1.0	

5	2.0	93.0	4.0	1.0
6	-1.0	100.0	7.0	4.0
7	-1.0	100.0	7.0	4.0
8	-1.0	96.5	8.0	4.0
9	-0.5	100.0	9.0	3.0

	is_holiday	is_weekend	Saison
0	0.0	1.0	3.0
1	0.0	1.0	3.0
2	0.0	1.0	3.0
3	0.0	1.0	3.0
4	0.0	1.0	3.0
5	0.0	1.0	3.0
6	0.0	1.0	3.0
7	0.0	1.0	3.0
8	0.0	1.0	3.0
9	0.0	1.0	3.0

### 0.0.5 Conversion et nettoyage

```
[7]: #Humidity values are between 0-100
data["Humidité"].describe()
```

```
[7]: count    17414.000000
      mean      72.324954
      std      14.313186
      min      20.500000
      25%      63.000000
      50%      74.500000
      75%      83.000000
      max      100.000000
      Name: Humidité, dtype: float64
```

```
[8]: data["Humidité"] = data["Humidité"] / 100
      data["Humidité"].describe()
```

```
[8]: count    17414.000000
      mean      0.723250
      std      0.143132
      min      0.205000
      25%      0.630000
      50%      0.745000
      75%      0.830000
      max      1.000000
      Name: Humidité, dtype: float64
```

### 0.0.6 Codes numériques → valeurs explicites

```
[9]: print(f'Valeurs enecin {data["Saison"].unique()}')
def conversion_saisons(n):
    match n:
        case 0:
            return 'Printemps'
        case 1:
            return 'Été'
        case 2:
            return 'Automne'
        case 3:
            return 'Hiver'

data["Saison"] = data["Saison"].apply(conversion_saisons).astype('string')
print(f'Nouveaus Valeurs {data["Saison"].unique()}')
```

```
Valeurs enecin [3. 0. 1. 2.]
Nouveaus Valeurs <StringArray>
['Hiver', 'Printemps', 'Été', 'Automne']
Length: 4, dtype: string
```

```
[10]: print(f'Valeurs enecin {data["Météo"].unique()}')
def conversion_Meteo(n):
    match n:
        case 1:
            return 'Clair'
        case 2:
            return 'Nuages épars'
        case 3:
            return 'Nuages fragmentés'
        case 4:
            return 'Couvert'
        case 7:
            return 'Pluie'
        case 10:
            return 'Pluie avec orage'
        case 26:
            return 'Neige'

data["Météo"] = data["Météo"].apply(conversion_Meteo).astype('string')
print(f'Nouveaus Valeurs {data["Météo"].unique()}')
```

```
Valeurs enecin [ 3.  1.  4.  7.  2. 26. 10.]
Nouveaus Valeurs <StringArray>
['Nuages fragmentés', 'Clair', 'Couvert',
 'Pluie', 'Nuages épars', 'Neige',
 'Pluie avec orage']
Length: 7, dtype: string
```

## 0.0.7 Exportation de data nettoyé

```
[11]: data.head(10)
```

```
[11]:          timestamp  Nombre de trajets  Température réelle (°C) \
0  2015-01-04 00:00:00                182                3.0
1  2015-01-04 01:00:00                138                3.0
2  2015-01-04 02:00:00                134                2.5
3  2015-01-04 03:00:00                 72                2.0
4  2015-01-04 04:00:00                 47                2.0
5  2015-01-04 05:00:00                 46                2.0
6  2015-01-04 06:00:00                 51                1.0
7  2015-01-04 07:00:00                 75                1.0
8  2015-01-04 08:00:00                131                1.5
9  2015-01-04 09:00:00                301                2.0
```

```
          Température ressentie (°C)  Humidité  Vitesse du vent (km/h) \
0                2.0      0.930                6.0
1                2.5      0.930                5.0
2                2.5      0.965                0.0
3                2.0      1.000                0.0
4                0.0      0.930                6.5
5                2.0      0.930                4.0
6               -1.0      1.000                7.0
7               -1.0      1.000                7.0
8               -1.0      0.965                8.0
9               -0.5      1.000                9.0
```

```
          Météo  is_holiday  is_weekend  Saison
0  Nuages fragmentés      0.0         1.0  Hiver
1              Clair      0.0         1.0  Hiver
2              Clair      0.0         1.0  Hiver
3              Clair      0.0         1.0  Hiver
4              Clair      0.0         1.0  Hiver
5              Clair      0.0         1.0  Hiver
6              Couvert      0.0         1.0  Hiver
7              Couvert      0.0         1.0  Hiver
8              Couvert      0.0         1.0  Hiver
9  Nuages fragmentés      0.0         1.0  Hiver
```

```
[12]: data["timestamp"] = pd.to_datetime(data["timestamp"])
data["Température ressentie (°C)"] = data["Température ressentie (°C)"].
↳ astype(int)
data["Température réelle (°C)"] = data["Température réelle (°C)"].astype(int)
data["Vitesse du vent (km/h)"] = data["Vitesse du vent (km/h)"].astype(int)
data["is_holiday"] = data["is_holiday"].astype(int)
```

```
data.head(10)
```

```
[12]:
```

	timestamp	Nombre de trajets	Température réelle (°C)	\
0	2015-01-04 00:00:00	182	3	
1	2015-01-04 01:00:00	138	3	
2	2015-01-04 02:00:00	134	2	
3	2015-01-04 03:00:00	72	2	
4	2015-01-04 04:00:00	47	2	
5	2015-01-04 05:00:00	46	2	
6	2015-01-04 06:00:00	51	1	
7	2015-01-04 07:00:00	75	1	
8	2015-01-04 08:00:00	131	1	
9	2015-01-04 09:00:00	301	2	

	Température ressentie (°C)	Humidité	Vitesse du vent (km/h)	\
0	2	0.930	6	
1	2	0.930	5	
2	2	0.965	0	
3	2	1.000	0	
4	0	0.930	6	
5	2	0.930	4	
6	-1	1.000	7	
7	-1	1.000	7	
8	-1	0.965	8	
9	0	1.000	9	

	Météo	is_holiday	is_weekend	Saison
0	Nuages fragmentés	0	1.0	Hiver
1	Clair	0	1.0	Hiver
2	Clair	0	1.0	Hiver
3	Clair	0	1.0	Hiver
4	Clair	0	1.0	Hiver
5	Clair	0	1.0	Hiver
6	Couvert	0	1.0	Hiver
7	Couvert	0	1.0	Hiver
8	Couvert	0	1.0	Hiver
9	Nuages fragmentés	0	1.0	Hiver

```
[211]: data.to_excel("london_bikes_final.xlsx",sheet_name="Londres",index=False)
```