

```
In [1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/forest-fire-dataset/forestfires.names
/kaggle/input/forest-fire-dataset/forestfires.csv
```

READ AND STORE THE DATA

Read and store the data in the variable using the read_csv()

```
In [2]: data=pd.read_csv("/kaggle/input/forest-fire-dataset/forestfires.csv")
data.head()
```

```
Out[2]:
```

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0

CHECKING FOR ANY NULL VALUES

```
In [3]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 13 columns):
```

```

#   Column  Non-Null Count  Dtype
---  -
0    X      517 non-null    int64
1    Y      517 non-null    int64
2    month   517 non-null    object
3    day     517 non-null    object
4    FFMC    517 non-null    float64
5    DMC     517 non-null    float64
6    DC      517 non-null    float64
7    ISI     517 non-null    float64
8    temp    517 non-null    float64
9    RH      517 non-null    int64
10   wind    517 non-null    float64
11   rain    517 non-null    float64
12   area    517 non-null    float64
dtypes: float64(8), int64(3), object(2)
memory usage: 52.6+ KB

```

In [4]: `data.describe()`

```

Out[4]:

```

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000
mean	4.669246	4.299807	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	0.021663	12.847292
std	2.313778	1.229900	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.295959	63.655818
min	1.000000	2.000000	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	0.000000	0.000000
25%	3.000000	4.000000	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	0.000000	0.000000
50%	4.000000	4.000000	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	0.000000	0.520000
75%	7.000000	5.000000	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	0.000000	6.570000
max	9.000000	9.000000	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000	6.400000	1090.840000

In [5]: `data.isna().sum()`

```

Out[5]:
X      0
Y      0
month   0

```

```
day      0
FFMC     0
DMC      0
DC       0
ISI      0
temp     0
RH       0
wind     0
rain     0
area     0
dtype: int64
```

VISUALIZATION

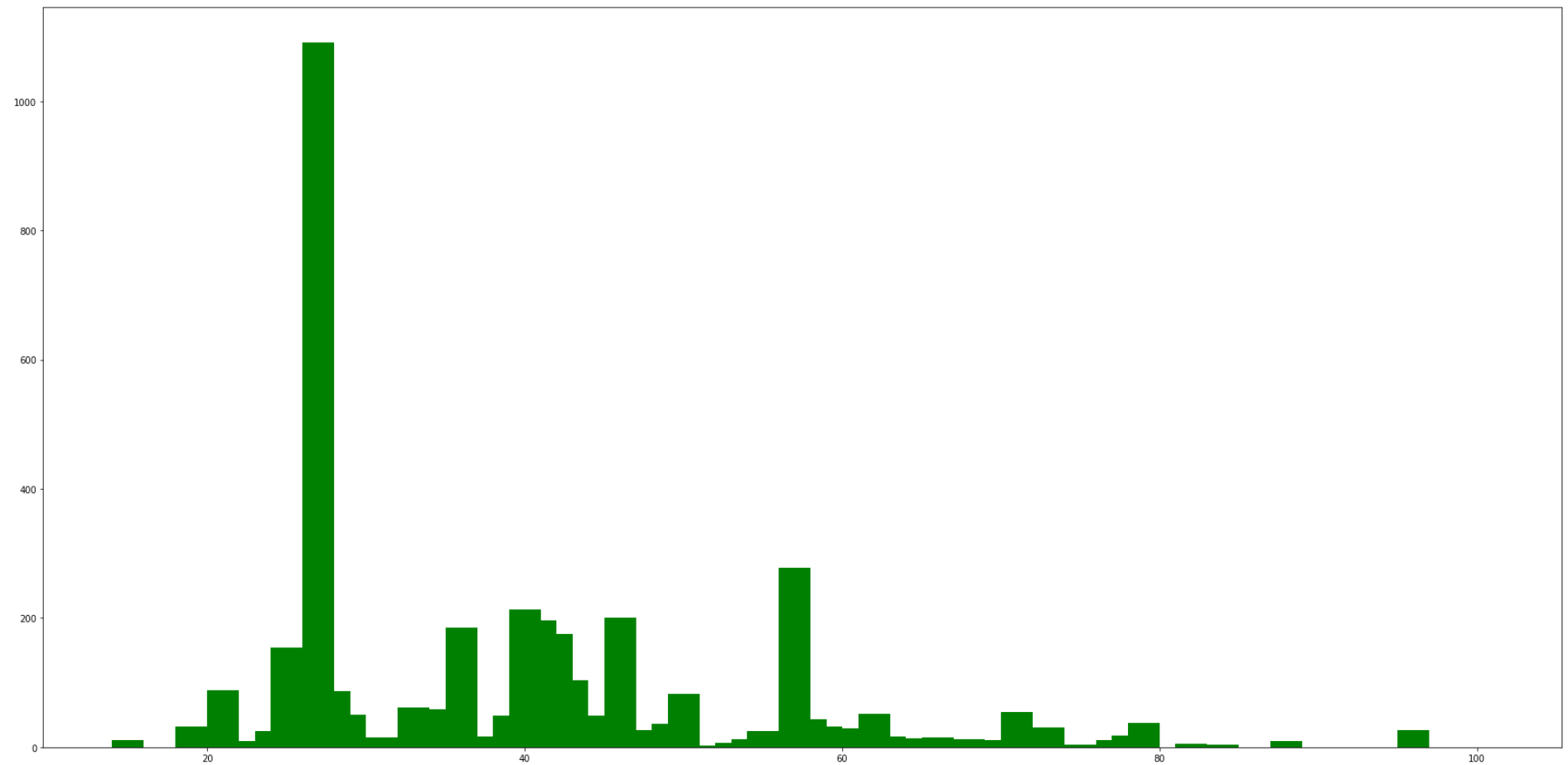
Visualizing the each and every Data with the Target Attribute

```
In [6]: import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [7]: rh=data["RH"].values
temp=data["temp"].values
ffmc=data["FFMC"].values
wind=data["wind"].values
rain=data["rain"].values
dc=data["DC"].values
isi=data["ISI"].values
dmc=data["DMC"].values
area=data["area"].values
```

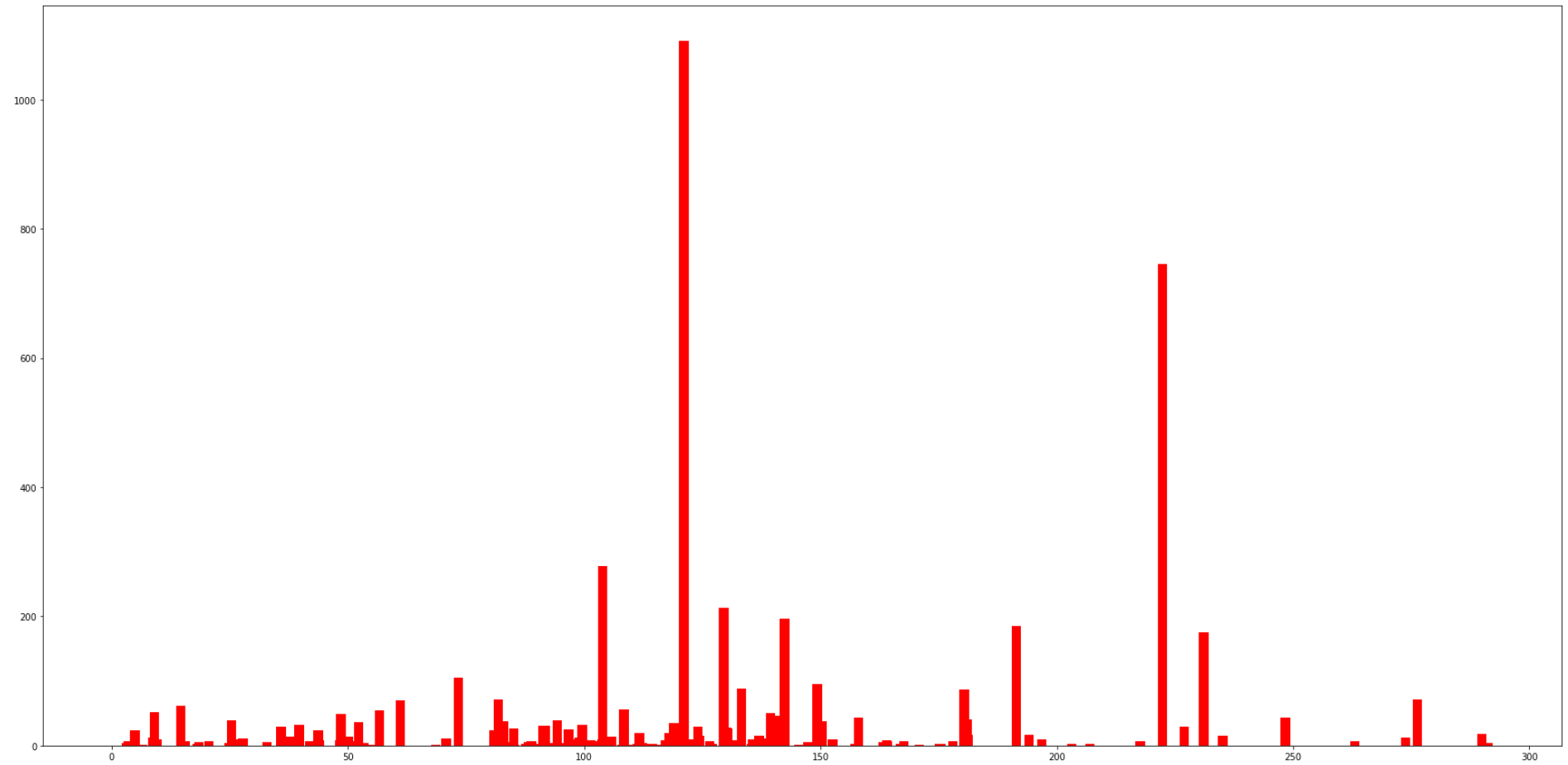
```
In [8]: plt.figure(figsize=(30,15))
plt.bar(rh,area,color="green",width=2)
```

```
Out[8]: <BarContainer object of 517 artists>
```



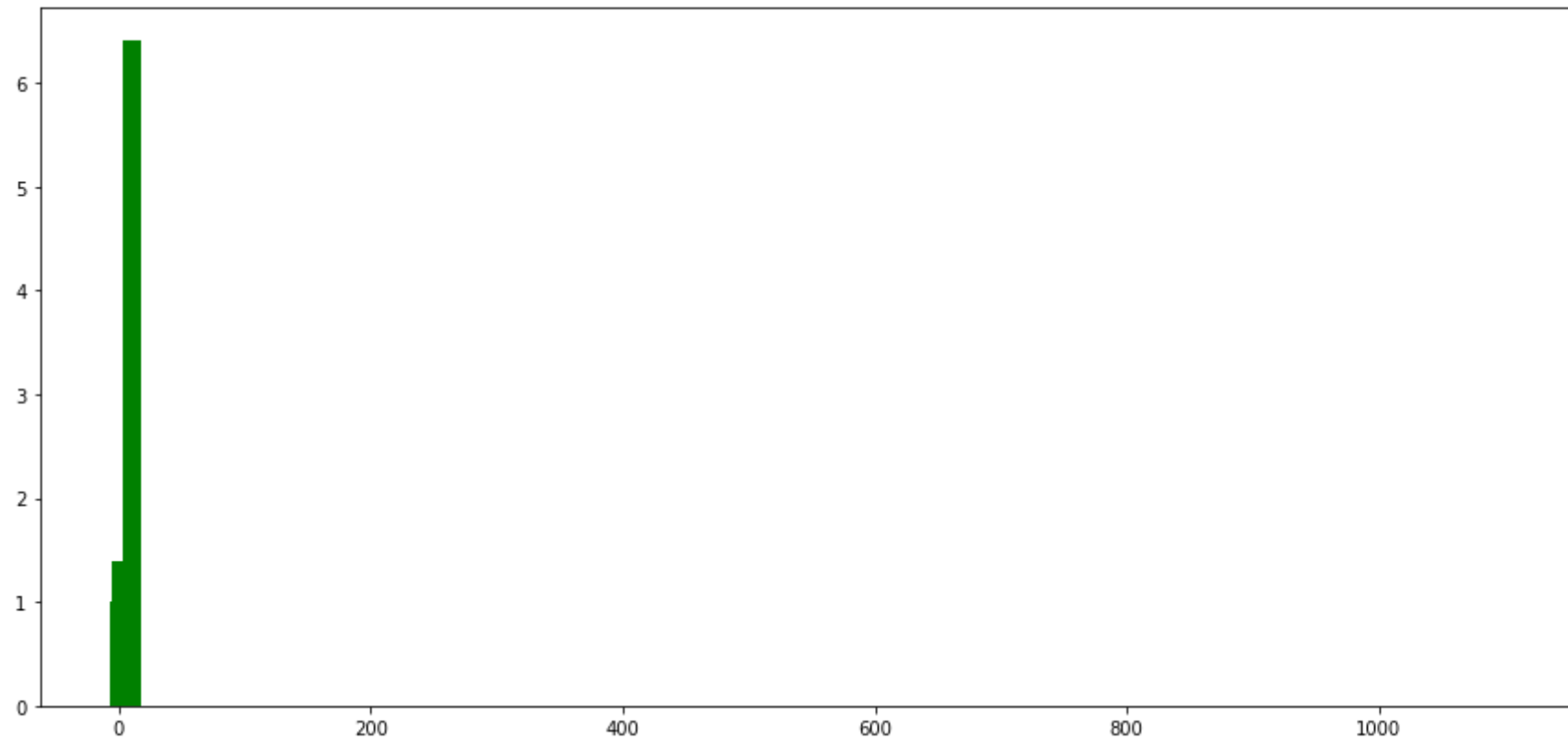
```
In [9]: plt.figure(figsize=(30,15))  
plt.bar(dmc,area,color="red",width=2)
```

```
Out[9]: <BarContainer object of 517 artists>
```



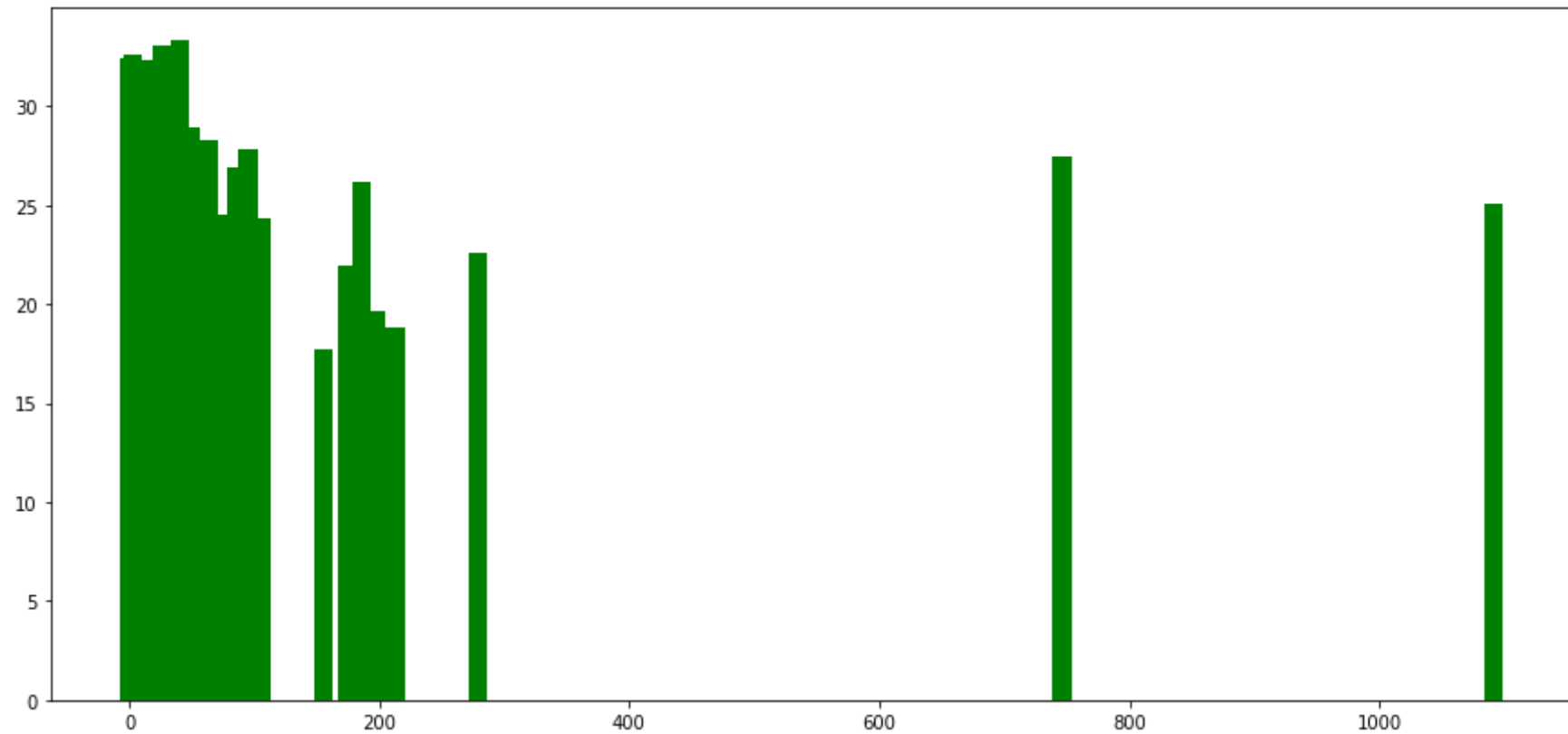
```
In [10]: plt.figure(figsize=(15,7))  
plt.bar(area,rain,color="green",width=15)
```

```
Out[10]: <BarContainer object of 517 artists>
```



```
In [11]: plt.figure(figsize=(15,7))  
plt.bar(area,temp,color="green",width=15)
```

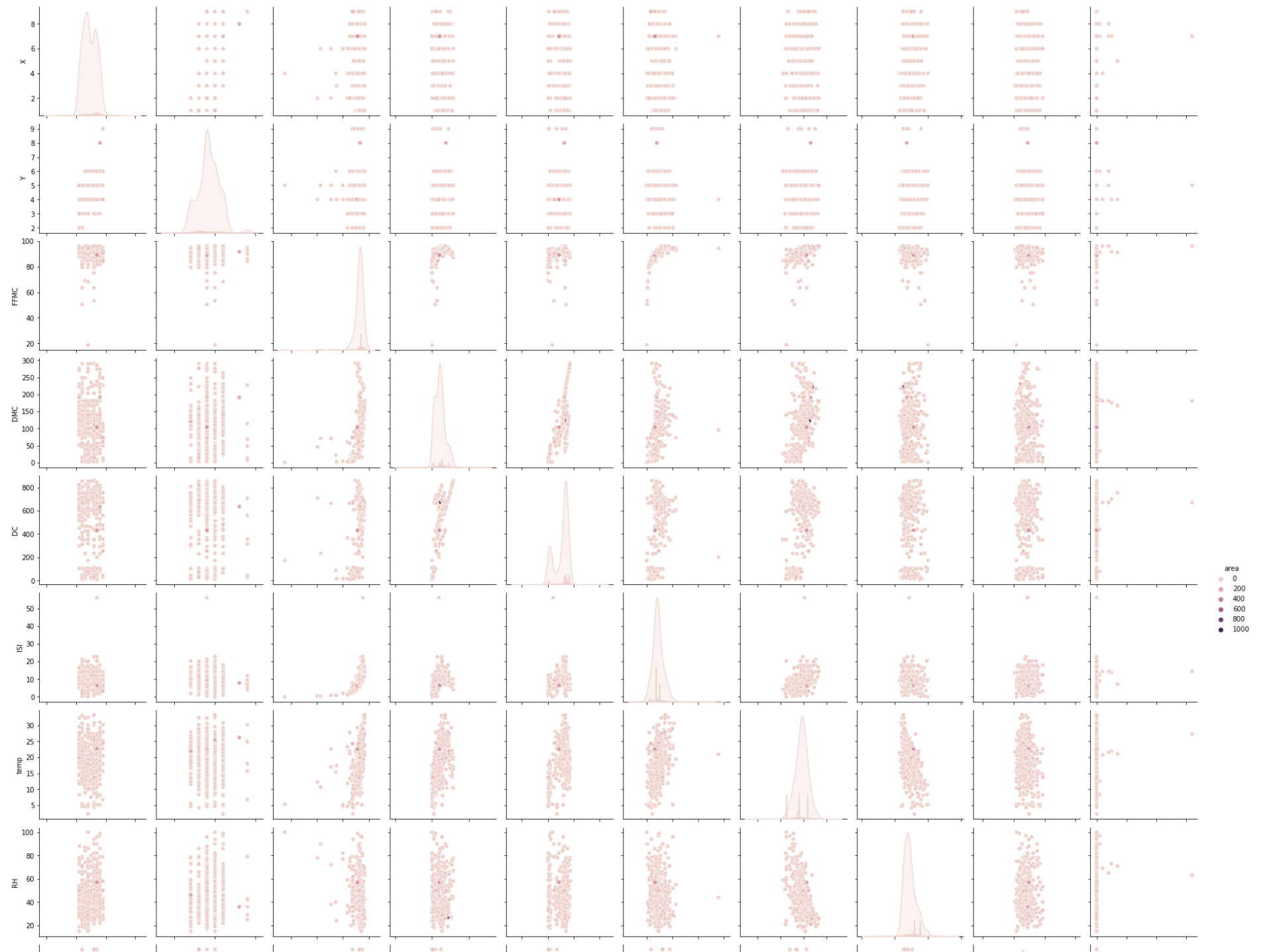
```
Out[11]: <BarContainer object of 517 artists>
```



In [12]:

```
plt.figure(figsize=(40,40))
sns.pairplot(data,hue="area")
plt.show()
```

<Figure size 2880x2880 with 0 Axes>



In []:

