In [6]:

```
import pandas as pd
import numpy as np
import sklearn
from sklearn.datasets import load_boston
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib
import warnings
from pandas_profiling import ProfileReport

warnings.filterwarnings('ignore')
np.random.seed(23)
```

In [7]:

```
pd1_df = pd.read_csv('forest_fires_dataset.csv')
pd1_df.info()
#The Fine Fuel Moisture Code (FFMC)
#The Duff Moisture Code (DMC)
#The Drought Code (DC)
#The Initial Spread Index (ISI)
#relative humidity RH
```

```
<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    float64
 10  wind    517 non-null    float64
 11  rain    517 non-null    float64
 12  area    517 non-null    float64
dtypes: float64(9), int64(2), object(2)
memory usage: 52.6+ KB
```

In [8]:

```
pd1_df.describe()
```

Out[8]:

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	ra
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.0000
mean	4.669246	4.299807	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	0.0216
std	2.313778	1.229900	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.2959
min	1.000000	2.000000	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	0.0000
25%	3.000000	4.000000	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	0.0000
50%	4.000000	4.000000	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	0.0000
75%	7.000000	5.000000	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	0.0000
max	9.000000	9.000000	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000	6.4000

In [9]:

```
pd1_df['area'].sort_values().tail(10)
```

Out[9]:

```
233     105.66
234     154.88
377     174.63
420     185.76
235     196.48
236     200.94
237     212.88
479     278.53
415     746.28
238    1090.84
Name: area, dtype: float64
```

2 największe wyniki, które widocznie różnią się od pozostałych

In [10]:

```
fire_df = pd1_df[pd1_df['area']<700]
fire_df.area.size
```

Out[10]:

```
515
```

Sprawdzmy ile jest przykładów kiedy area=0

In [11]:

```
fire_df[fire_df['area'] == 0].area.size
```

Out[11]:

```
247
```

247/515,prawię połowa

In [12]:

```
burned_df = fire_df[fire_df['area']!= 0]
safe_df = fire_df[fire_df['area']== 0]
```

Sprawdzmy ile jest wierszy dla których rain=0?

In [13]:

```
norain_df = fire_df[fire_df['rain'] == 0]
norain_df.area.size
```

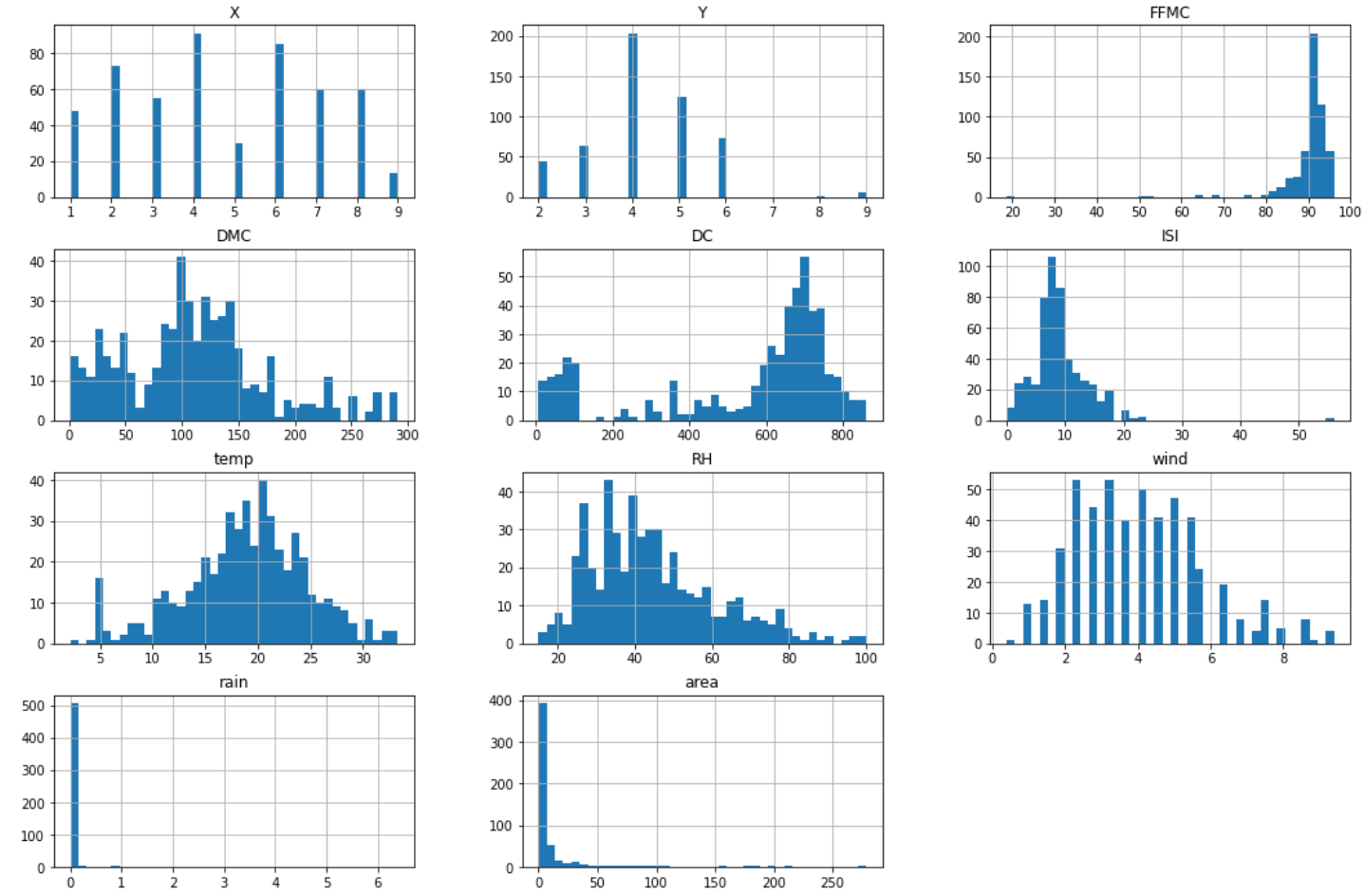
Out[13]:

```
507
```

507 z 515, ta zmienna jest prawię zawszę równa 0

In [15]:

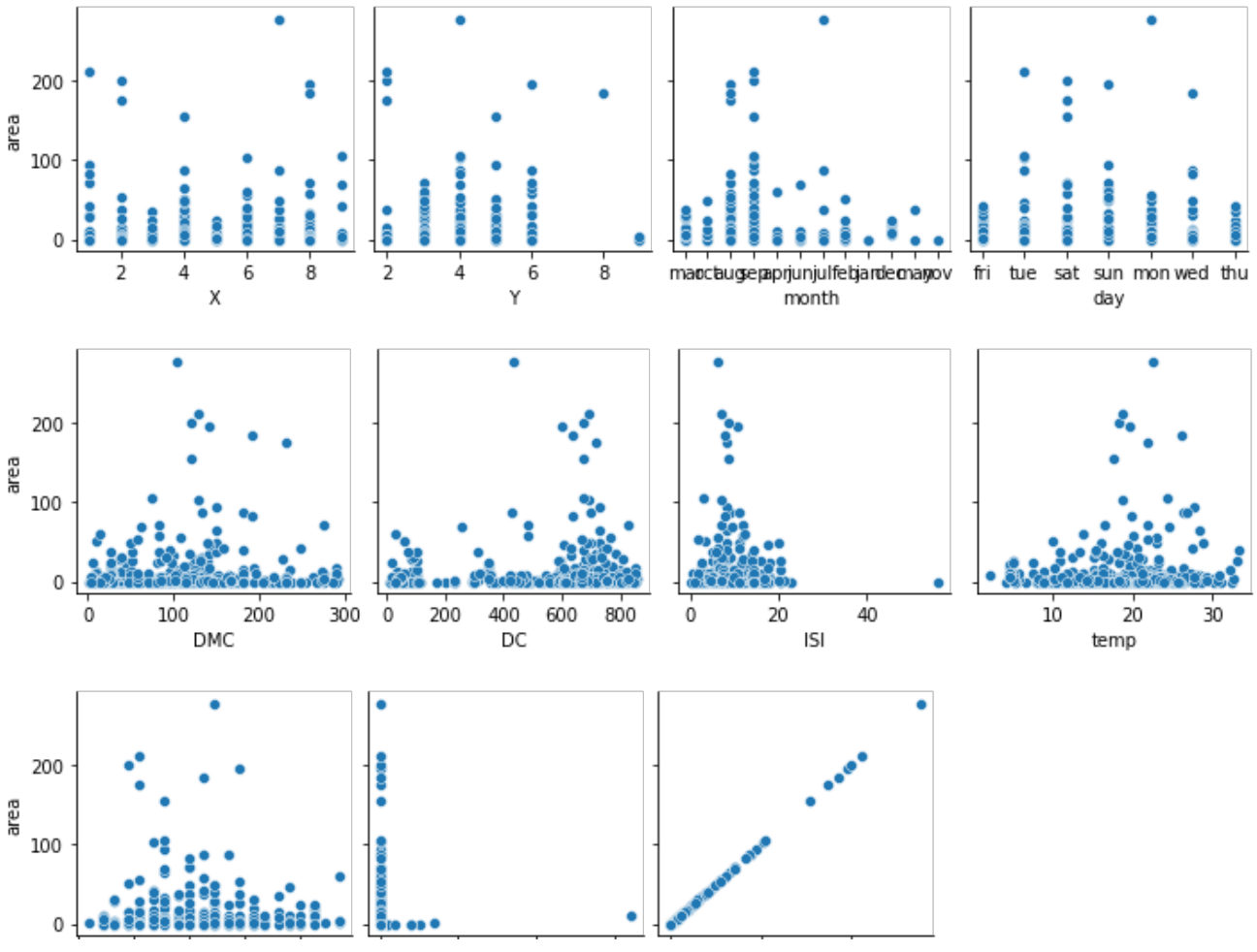
```
fire_df.hist(bins = 40, figsize=(18, 12))
plt.show()
```

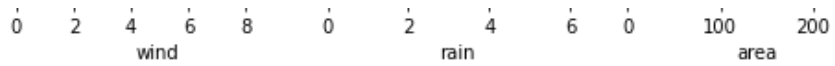


In [16]:

```
sns.pairplot(fire_df, y_vars="area", x_vars=fire_df.columns.values[:4], diag_kind=None)
sns.pairplot(fire_df, y_vars="area", x_vars=fire_df.columns.values[5:9], diag_kind=None)
sns.pairplot(fire_df, y_vars="area", x_vars=fire_df.columns.values[10:], diag_kind=None)

plt.show()
```



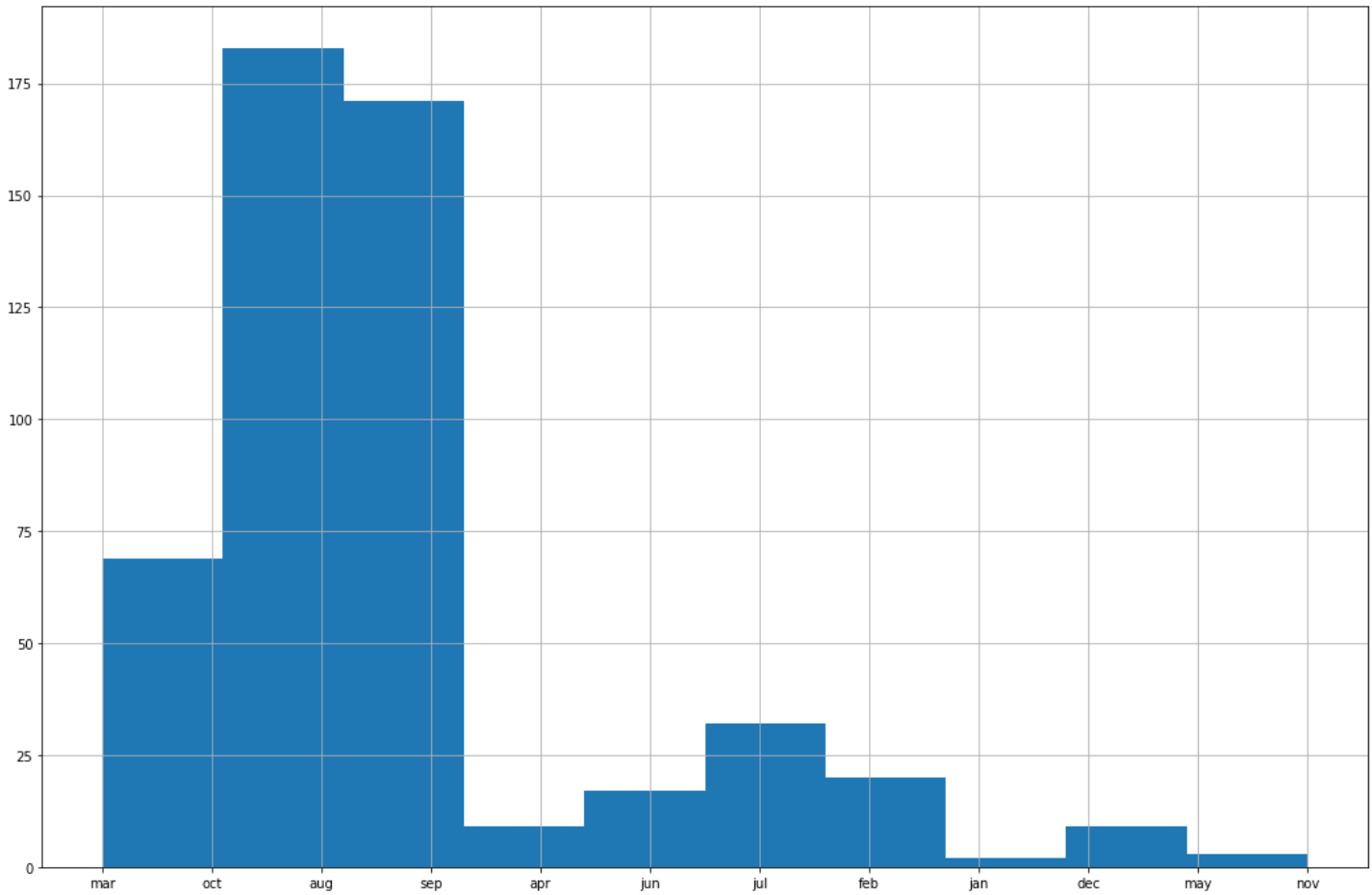


In [17]:

```
fire_df['month'].hist(figsize=(18, 12))
# najwięcej przypadków w sep i aug
```

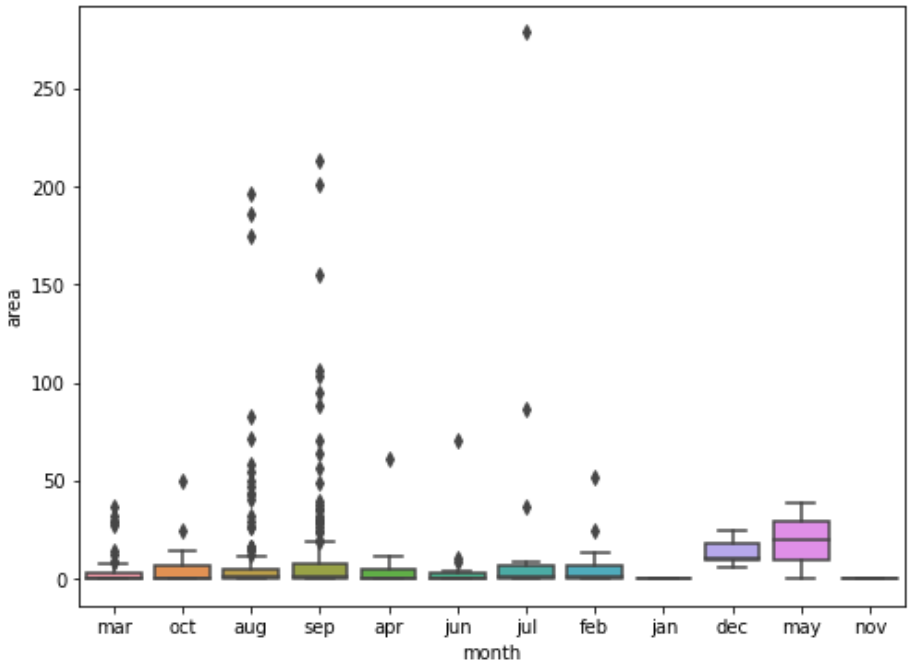
Out[17]:

<AxesSubplot:>



In [18]:

```
data = pd.concat([fire_df['area'], fire_df['month']], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x="month", y="area", data=data)
```



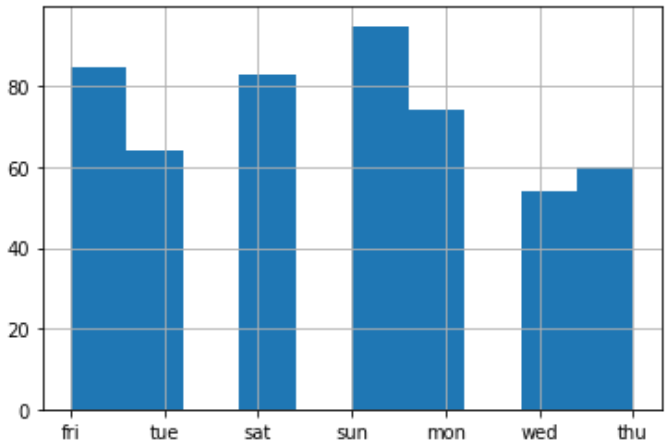
Tak jak się można domyślać w mesiącach letnich oprócz "mało" powierzchniowych pożarów występują także pożary o większej powierzchni

In [19]:

```
fire_df['day'].hist()
```

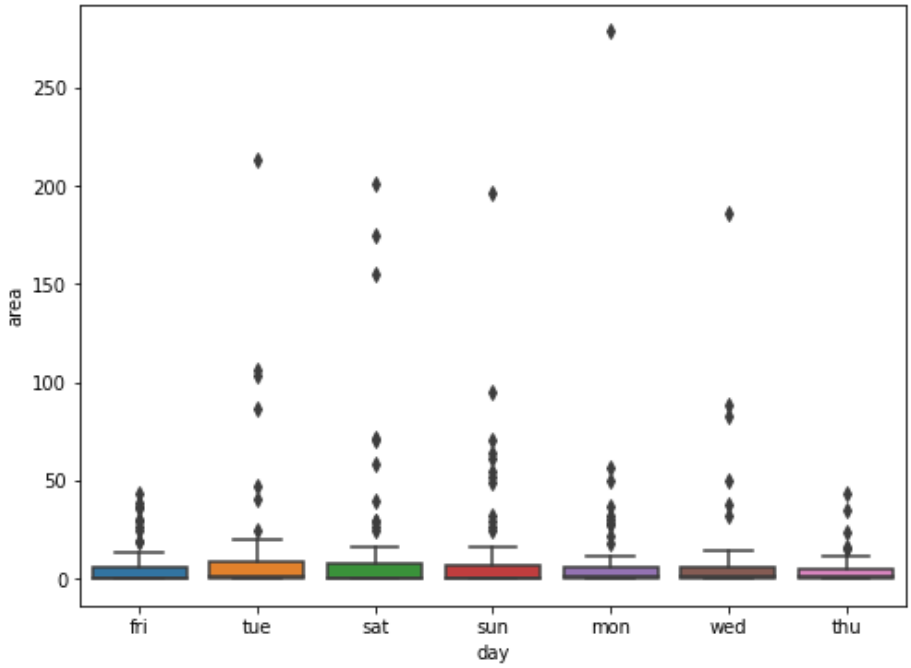
Out[19]:

<AxesSubplot:>



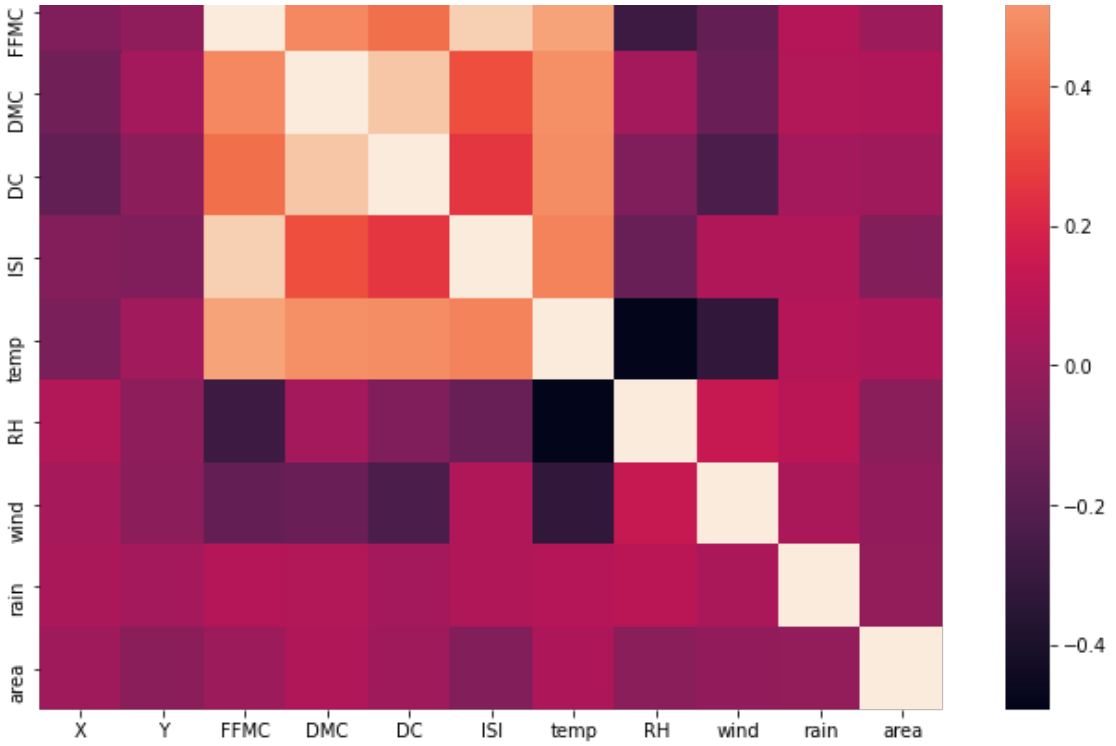
In [20]:

```
data = pd.concat([fire_df['area'], fire_df['day']], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x="day", y="area", data=data)
```



In [21]:

```
corrmat = burned_df.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True);
```

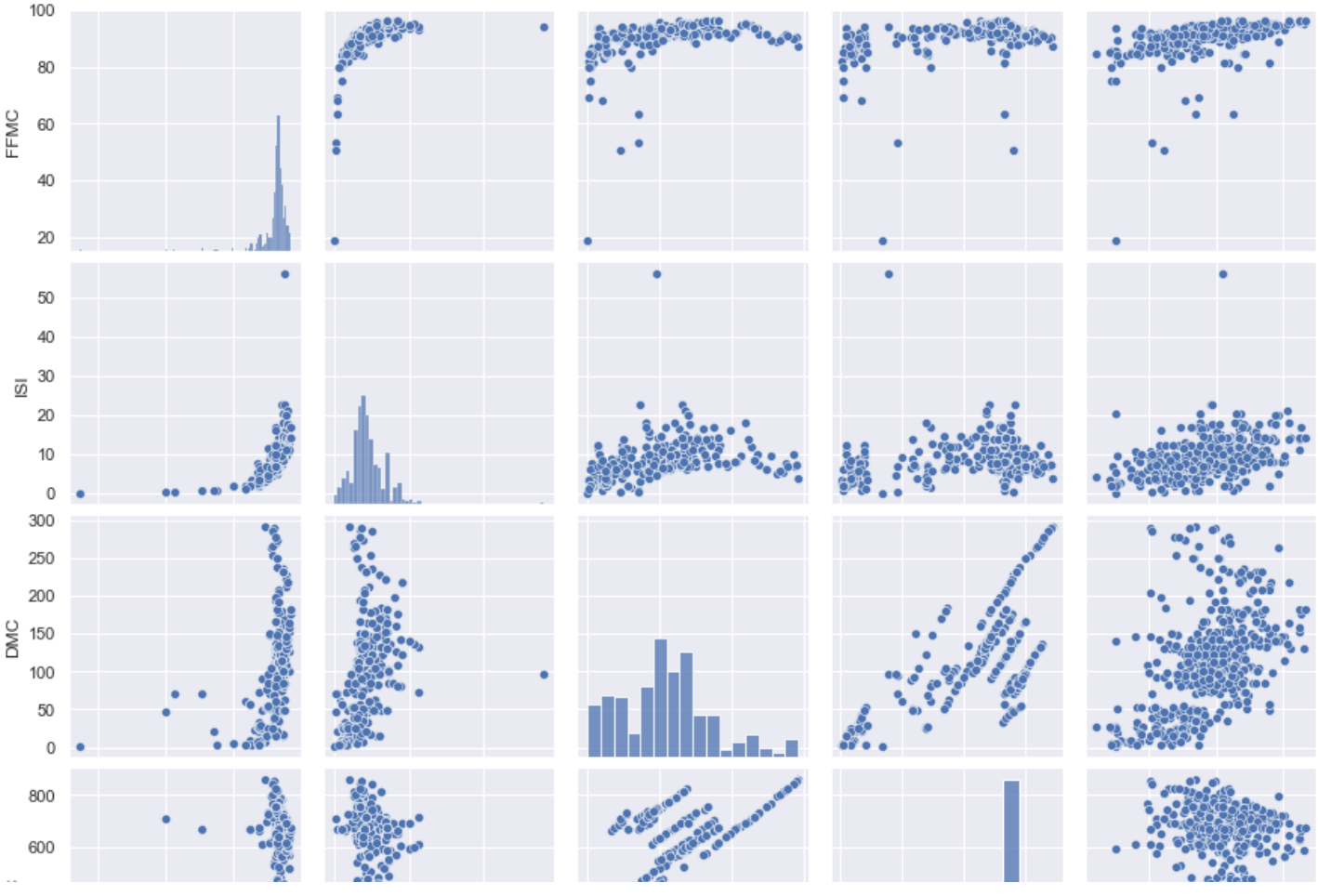


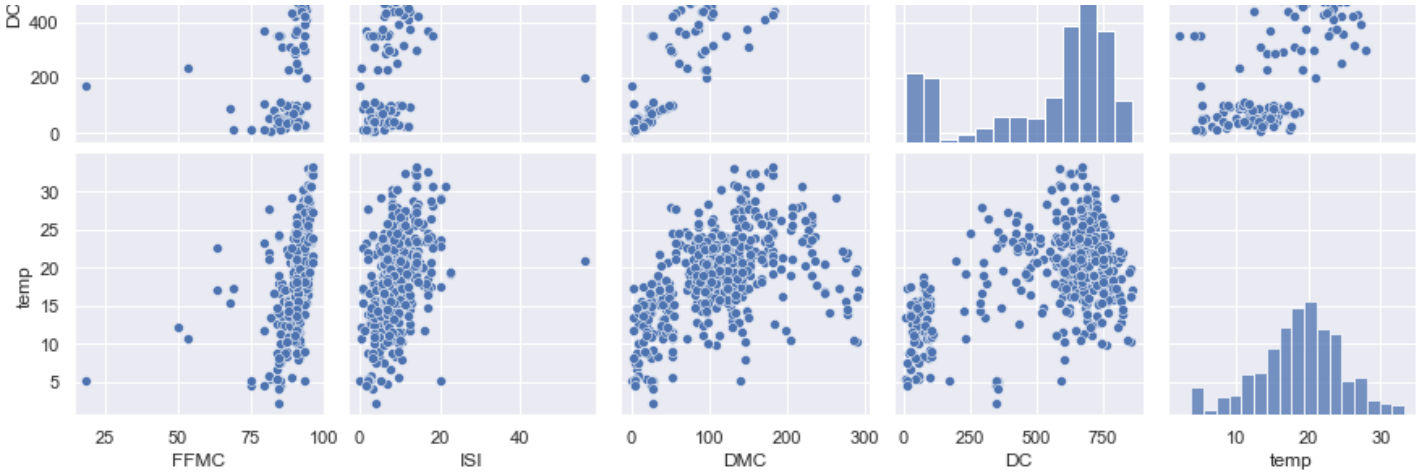
Możemy zaóważyć korelację pomiędzy:

- **X i Y**
- **FFMC i ISI**
- **DC i DMC**

In [22]:

```
sns.set()
cols = ['FFMC', 'ISI', 'DMC', 'DC', 'temp']
sns.pairplot(fire_df[cols], height = 2.5)
plt.show();
```





In [23]:

```
prof = ProfileReport(pd1_df)
prof.to_file(output_file='output.html')
```

Użycie pandas profiling pozwala na łatwę uzyskanie wstępnej eksploracji danych. Z raportu dowiedzieliśmy się o 4 duplikatach w danych oraz dużym procencie wartości zerowych dla kolumn area i rain. Ograniczaniem jest np. brak sprawdzania danych po usunięciu odstających wyników.