pd1

March 9, 2021

1 Praca domowa 1- eksploracja danych

[2]:
```
import pandas_profiling
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
import math
from mpl_toolkits.mplot3d import Axes3D
```

[34]:
```
data=pd.DataFrame(pd.read_json('https://api.apispreadsheets.com/api/dataset/
↪forest-fires/',orient='split'))
```

[35]:
```
print(f'liczba obserwacji: {data.shape[0]}, Liczba kolumn: {data.shape[1]}')
```

```
liczba obserwacji: 517, Liczba kolumn: 13
```

1.1 Rozpoczniemy od sprawdzenia kompletności danych.

[36]:
```
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
```

1.1.1 Żadna z kolumn nie zawiera nulli więc dane nie są wybrakowane

1.2 Pprzebadania naszego targetu,który chcemy miodelować, czyli daną area.

[37]:
```
data['area'].describe()
```

[37]:
```
count     517.000000
mean       12.847292
std        63.655818
min         0.000000
25%         0.000000
50%         0.520000
75%         6.570000
max      1090.840000
Name: area, dtype: float64
```

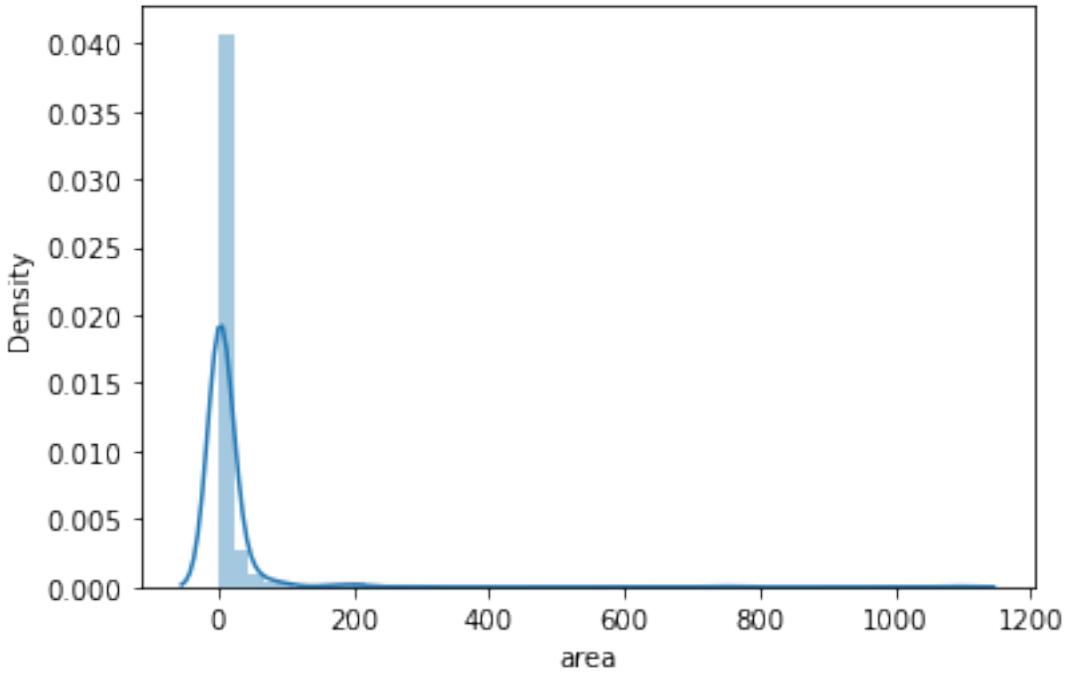
1.3 Łatwo zauważyć, że większość obserwacji znajduje się blisko zera.

[38]:
```
print(f'Kurtoza: {data["area"].kurt()}')
print(f'Skośność: {data["area"].skew()}')
sns.distplot(data['area'])
```

```
Kurtoza: 194.1407210942299
Skośność: 12.846933533934868
```

```
D:\anaconda\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

[38]:
```
<AxesSubplot:xlabel='area', ylabel='Density'>
```



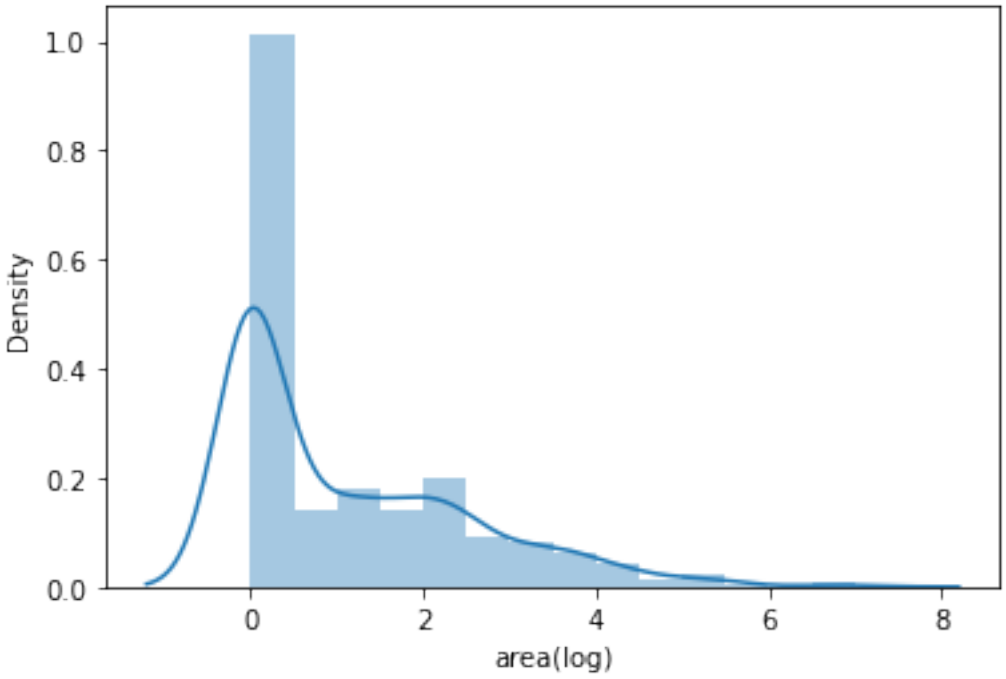
1.4 Z wykresu szybko wnioskujemy, że przydałoby się transformować tą zmienną do jej analizy. Skorzystamy z $\log(x)$ zgodnie z sugestią autorów danych.

[39]:
```
data['area(log)']=data['area'].map(lambda x: np.log(x+1))
```

[40]:
```
print(f'Kurtoza: {data["area(log)"].kurt()}')
print(f'Skośność: {data["area(log)"].skew()}')
sns.distplot(data['area(log)'])
```

```
Kurtoza: 0.9456680757207487
Skośność: 1.2178376559535011

D:\anaconda\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

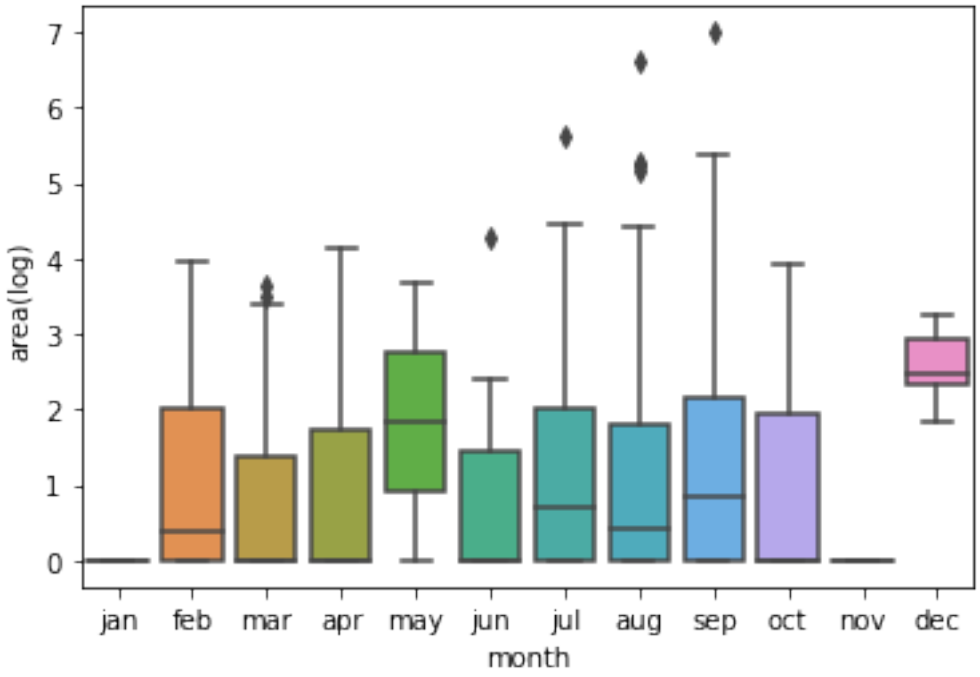
[40]:
```
<AxesSubplot:xlabel='area(log)', ylabel='Density'>
```



1.5 Od razu widać większą przejrzystość

[41]:
```
sns.
↪boxplot(x='month',y='area(log)',data=data,order=['jan','feb','mar','apr','may','jun','jul','a
```

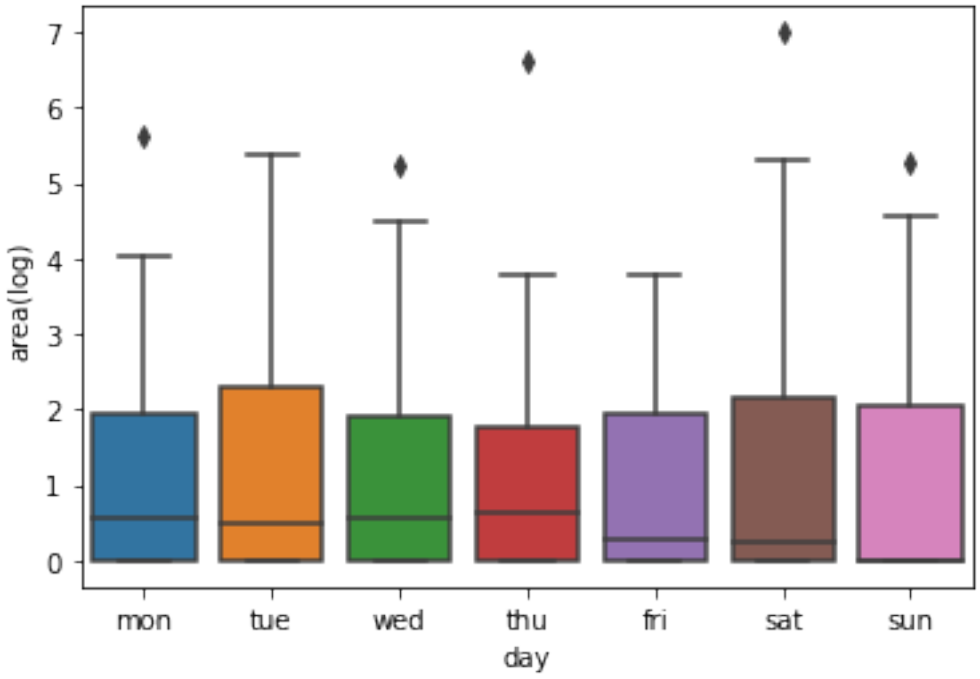
[41]:
```
<AxesSubplot:xlabel='month', ylabel='area(log)'>
```



1.6 Łatwo zauważyć, że to właśnie podczas miesięcy letnich miały miejsce najgorsze pożary, co jednak ciekawe to w maju i w grudniu dolny kwartyl nie jest w zerze(bądź bardzo blisko niego).

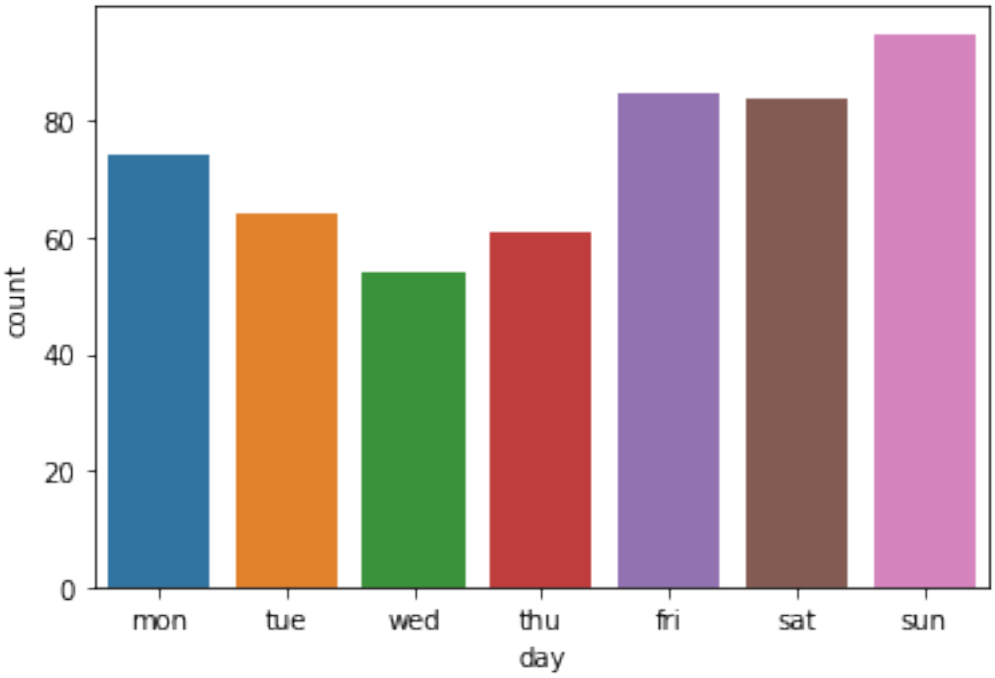
[42]:
```
sns.
↪boxplot(x='day',y='area(log)',data=data,order=['mon','tue','wed','thu','fri','sat','sun'])
```

[42]:
```
<AxesSubplot:xlabel='day', ylabel='area(log)'>
```



[43]:
```
sns.
↪countplot(data=data,x='day',order=['mon','tue','wed','thu','fri','sat','sun'])
```

[43]: <AxesSubplot:xlabel='day', ylabel='count'>



1.7 Tu bez większych niespodzianek dzień tygodnia nie wpływa zbytnio na rozmiar, ani na częstotliwość pożarów pożarów.

1.8 Następnie spojrzymy na cechy ciągłe w poszukiwaniu większej ilości zależności

[44]:
```
data.describe()
```

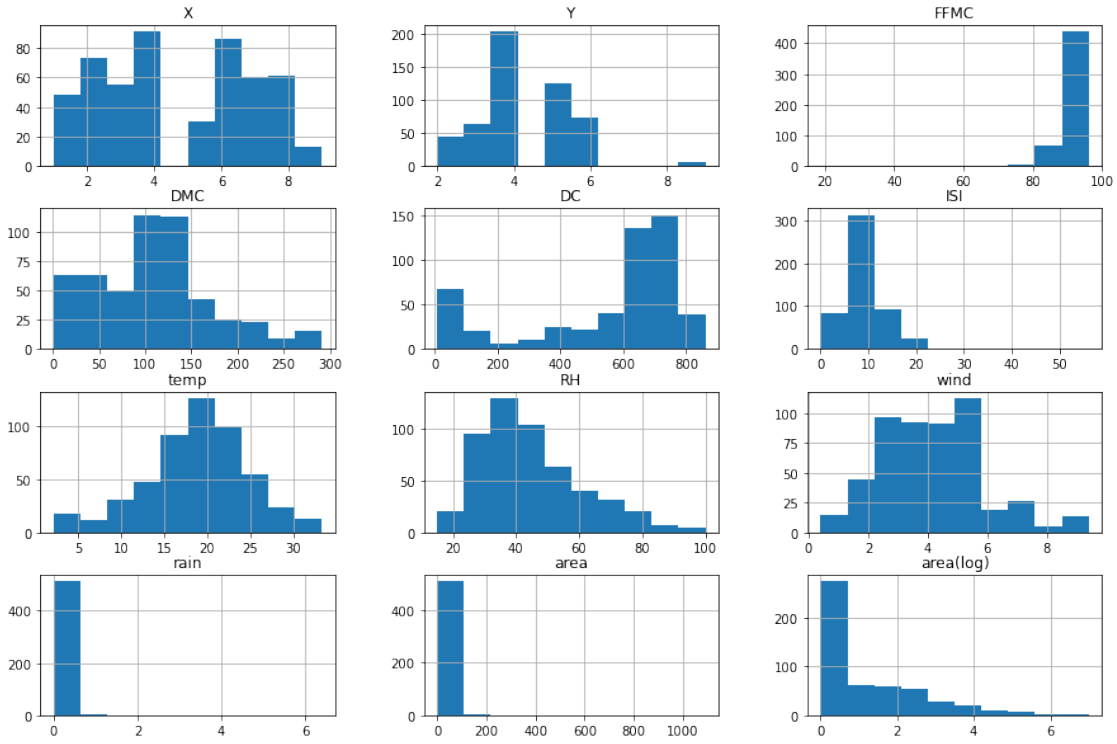
[44]:

	X	Y	FFMC	DMC	DC	ISI
count	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
std	2.313778	1.229900	5.520111	64.046482	248.066192	4.559477
min	1.000000	2.000000	18.700000	1.100000	7.900000	0.000000
25%	3.000000	4.000000	90.200000	68.600000	437.700000	6.500000
50%	4.000000	4.000000	91.600000	108.300000	664.200000	8.400000
75%	7.000000	5.000000	92.900000	142.400000	713.900000	10.800000
max	9.000000	9.000000	96.200000	291.300000	860.600000	56.100000

	temp	RH	wind	rain	area	area(log)
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000
mean	18.889168	44.288201	4.017602	0.021663	12.847292	1.111026
std	5.806625	16.317469	1.791653	0.295959	63.655818	1.398436
min	2.200000	15.000000	0.400000	0.000000	0.000000	0.000000
25%	15.500000	33.000000	2.700000	0.000000	0.000000	0.000000
50%	19.300000	42.000000	4.000000	0.000000	0.520000	0.418710
75%	22.800000	53.000000	4.900000	0.000000	6.570000	2.024193
max	33.300000	100.000000	9.400000	6.400000	1090.840000	6.995620

[45]:
```
data.hist(bins=10, figsize=(15,10))
```

[45]:
```
array([[<AxesSubplot:title={'center':'X'}>,
        <AxesSubplot:title={'center':'Y'}>,
        <AxesSubplot:title={'center':'FFMC'}>],
       [<AxesSubplot:title={'center':'DMC'}>,
        <AxesSubplot:title={'center':'DC'}>,
        <AxesSubplot:title={'center':'ISI'}>],
       [<AxesSubplot:title={'center':'temp'}>,
        <AxesSubplot:title={'center':'RH'}>,
        <AxesSubplot:title={'center':'wind'}>],
       [<AxesSubplot:title={'center':'rain'}>,
        <AxesSubplot:title={'center':'area'}>,
        <AxesSubplot:title={'center':'area(log)'}>]], dtype=object)
```



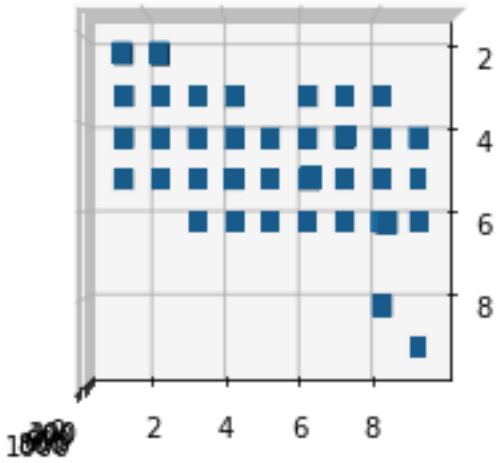
1.9 Co ciekawe partie lasu najczęściej dotykane przez pożary są przedzielone pasmami, w których pożary nie występuję(z rozkładu zmiennych X i Y). Widzimy również, że FFMC oraz rain są skupione blisko jednej wartości. A ISI i temp najbardziej przypominają rozkład normalny.

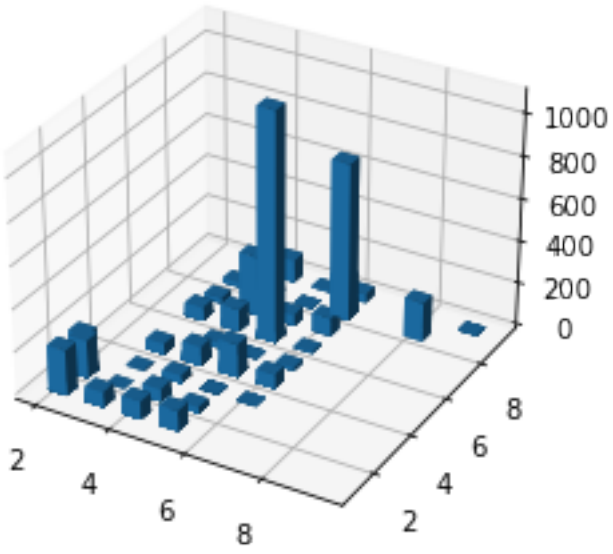
[ ]:

1.10 Sprawdźmy więc, w których częściach lasu występują najgroźniejsze pożary.

[46]:
```
df=data.iloc[:,[0,1,12]]


plt.show()
fig = plt.figure()
ax = fig.gca(projection='3d')
ax.bar3d(df['Y'], df['X'],0,0.5,0.5,df['area'], cmap=plt.cm.viridis, linewidth=0.
 ↪02)
ax.view_init(90, 0)


plt.show()
fig = plt.figure()
```

```
ax = fig.gca(projection='3d')
ax.bar3d(df['Y'], df['X'], 0,0.5,0.5,df['area'], cmap=plt.cm.viridis,␣
 ↪linewidth=0.02)

plt.show()
```

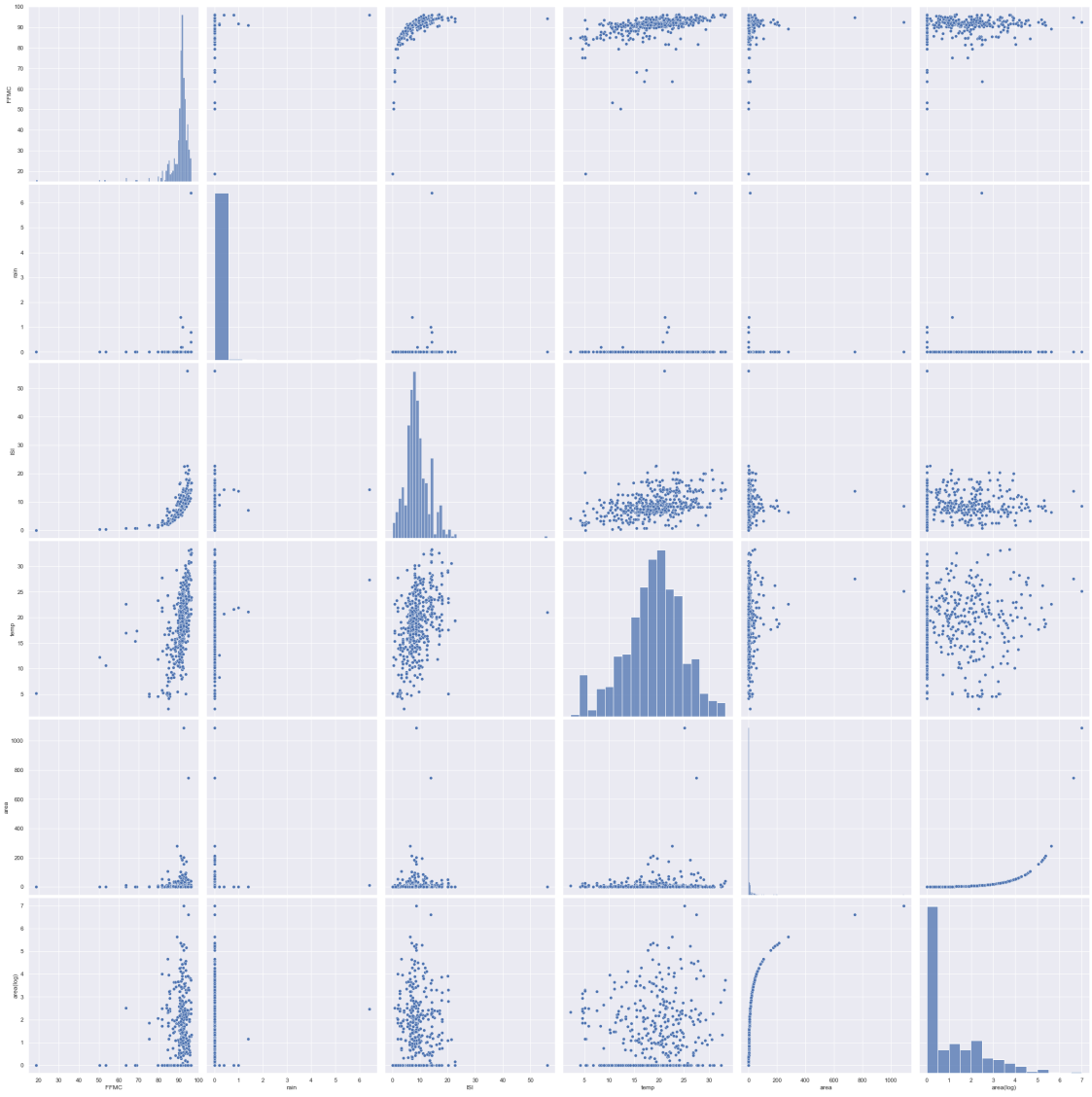




1.11 Z wykresów widzimy, że najgorsze pożary wybuchają w centralnej części lasu oraz na obrzeżach.

[47]:
```
data_subset=data[['FFMC','rain','ISI','temp','area','area(log)']]
sns.set()
sns.pairplot(data_subset, height=5)
```

[47]: <seaborn.axisgrid.PairGrid at 0x20be905f340>

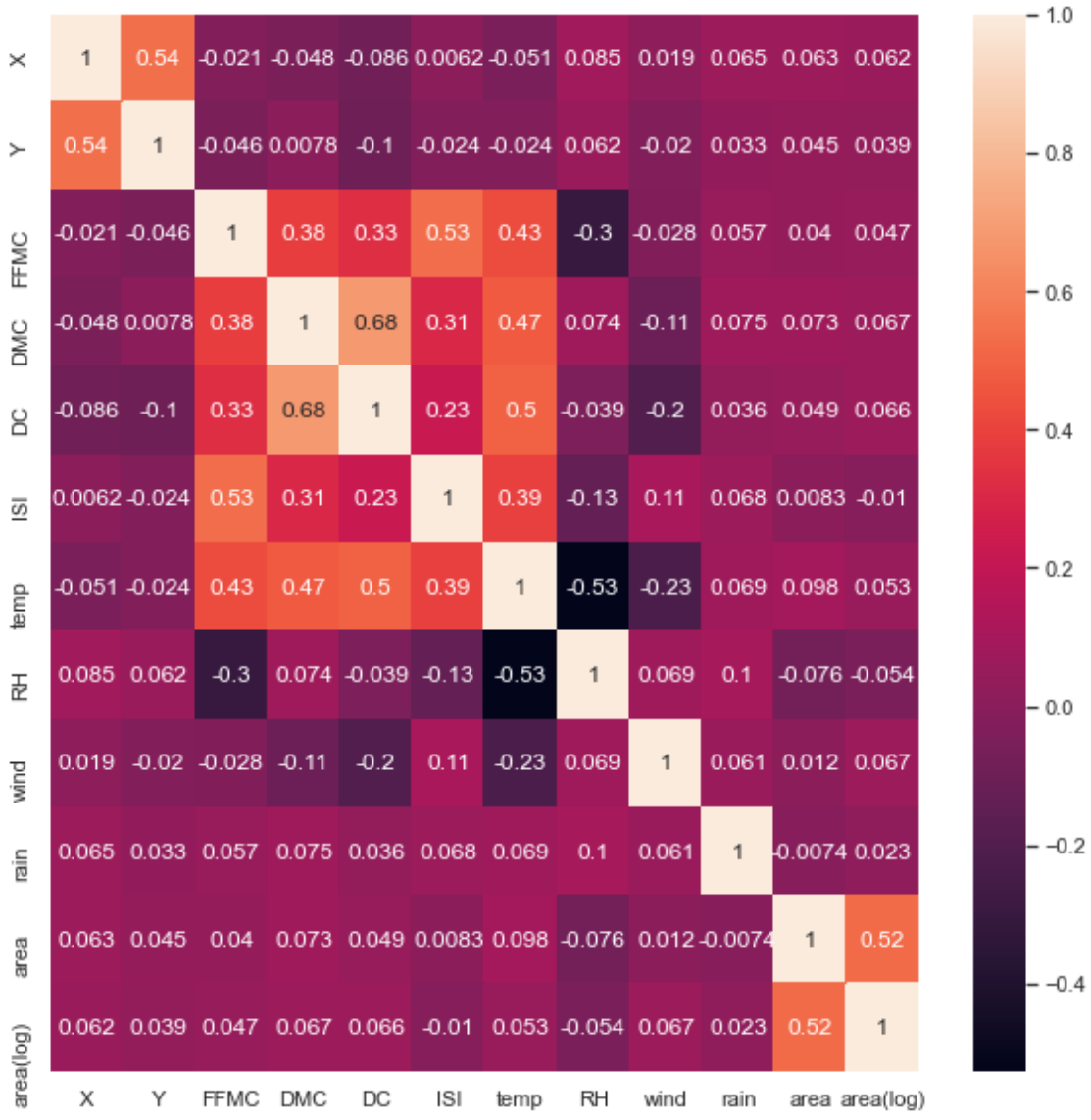


1.12 Zauważamy pewne zależności pomiędzy ISI,a FFMC oraz pomiędzy temp, a FFMC.

1.13 Sprawdzamy, więc współczynniki korelacji pomiędzy zmiennymi.

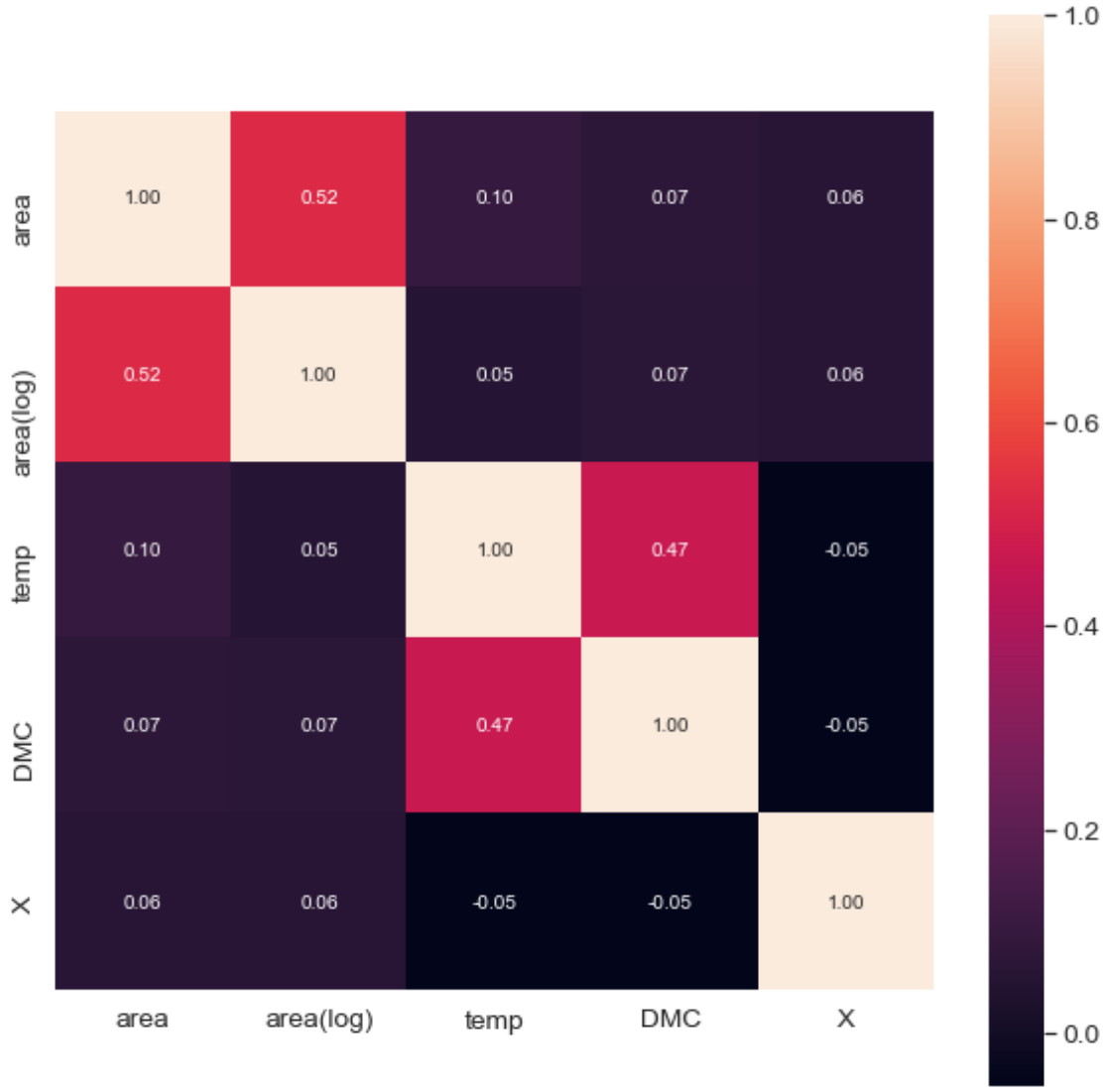
[48]:
```
plt.figure(figsize=(10,10))
sns.heatmap(data.corr(),annot=True)
```

[48]: <AxesSubplot:>



1.14 Zauważamy, że X i Y są ze sobą dość mocno skorelowane podobnie jak zmienne FFMC, DMC, DC, ISI oraz temp. Niestety nie znajdujemy żadnych korelacji pomiędzy area, a innymi zmiennymi.

[49]:
```
plt.figure(figsize=(10,10))
k = 5
cols = data.corr().nlargest(k, 'area')['area'].index
cm = np.corrcoef(data[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f',␣
 ↪annot_kws={'size': 10}, yticklabels=cols.values, xticklabels=cols.values)
plt.show()
```



1.15 Zmiennymi najbardziej skorelowanymi do area okazują się temp, DMC i X.

[50]:
```
pandas_profiling.ProfileReport(data)
```

```
HBox(children=(HTML(value='Summarize dataset'), FloatProgress(value=0.0, max=27.
↪0), HTML(value='')))


HBox(children=(HTML(value='Generate report structure'), FloatProgress(value=0.0,␣
↪max=1.0), HTML(value='')))


HBox(children=(HTML(value='Render HTML'), FloatProgress(value=0.0, max=1.0),␣
↪HTML(value='')))


<IPython.core.display.HTML object>
```

[50]:

1.16 Automatyczny sposób eksploracji danych pozwala szybko zapoznać się ze strukturą danych, jednak nie daje ich pełnej analizy. Jest to dobry punkt początkowy ukazujący podstawowe zależności między danymi. Nie daje on jednak zbytniej sfobody co do zależności, które wyświetla np. nie możemy sprawdzić korelacji zmiennych ciągłych z kategorycznymi.

[ ]:

[ ]:

[ ]: