pd1_sinski (2)

March 8, 2021

1 Praca domowa 1

1.1 Bartosz Siński

[4]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
from pandas_profiling import ProfileReport
import datetime as dt
from datetime import datetime
```

[6]:
```
fires_df = pd.read_csv("./src/forest_fires_dataset.csv")
```

[8]:
```
fires_df.head()
```

[8]:

	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	51.0	6.7	0.0	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33.0	0.9	0.0	0.0
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33.0	1.3	0.0	0.0
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97.0	4.0	0.2	0.0
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99.0	1.8	0.0	0.0

[10]:
```
fires_df.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    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
```

[12]:
```
fires_df.describe()
```

[12]:

	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
count	517.000000	517.000000	517.000000	517.000000	517.000000
mean	18.889168	44.288201	4.017602	0.021663	12.847292
std	5.806625	16.317469	1.791653	0.295959	63.655818
min	2.200000	15.000000	0.400000	0.000000	0.000000
25%	15.500000	33.000000	2.700000	0.000000	0.000000
50%	19.300000	42.000000	4.000000	0.000000	0.520000
75%	22.800000	53.000000	4.900000	0.000000	6.570000
max	33.300000	100.000000	9.400000	6.400000	1090.840000

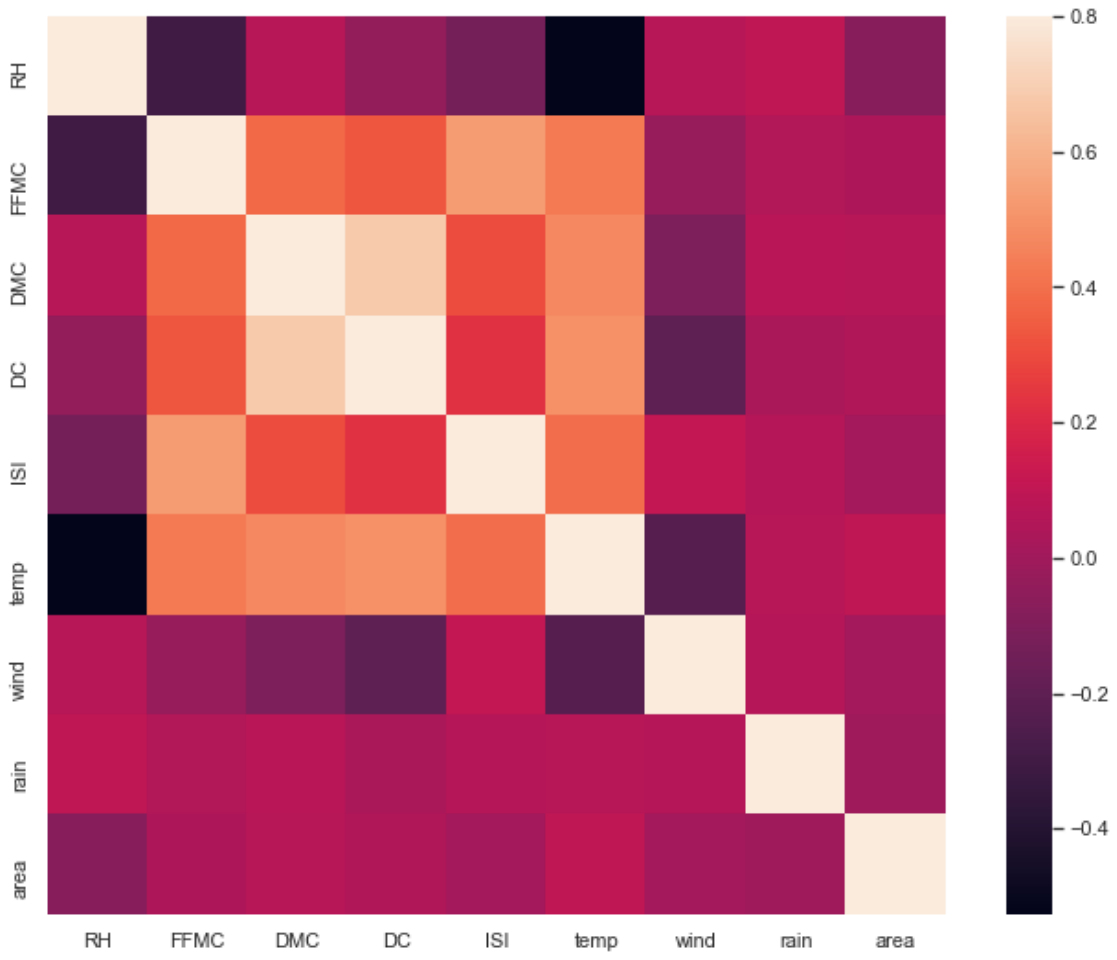
[14]:
```
fires_df.isnull().sum()
```

[14]:
```
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
```

Po wstępnym przejrzeniu naszych danych widzimy, że raczej będzie problemu z brakujacymi

danymi. Jednak już teraz widać, że rozkład wartości przy niektórych atrybutach będzie daleki od normalnego.

[16]:
```
fires_subset1 =␣
↪fires_df[['RH','FFMC','DMC','DC','ISI','temp','wind','rain','area']]
corrmat = fires_subset1.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True);
```

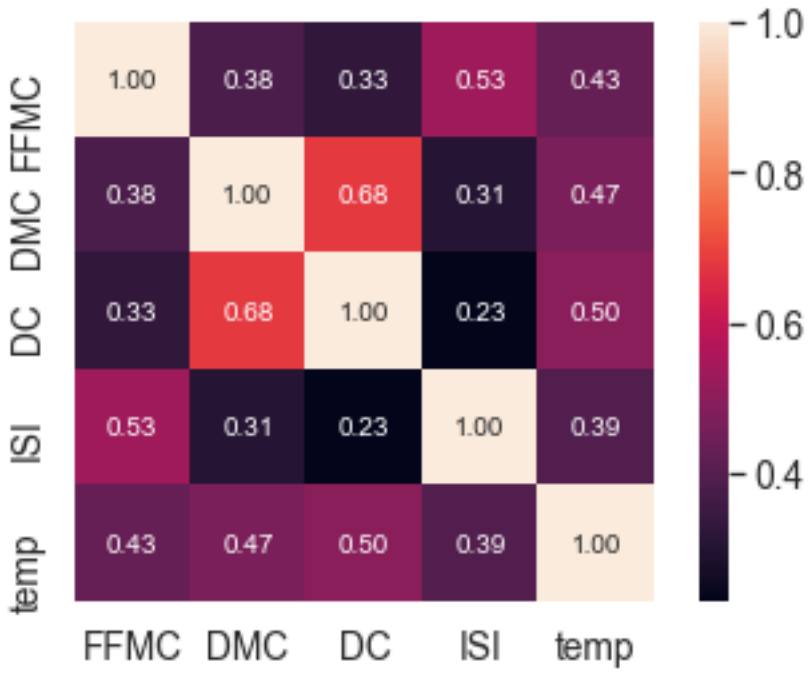


Widzimy, że najlepsze korelacje zachodzą pomiędzy poszczególnymi wyznacznikami wiglotności lasu: * FFMC - wilgotność ściółki * DMC - wilgotnosc wartwy pod ściółką * DC - wilgotnosc gleby

Nie jest to zbytnio odkrywcze jednak dobrze widac także zależnośći pomiędzy wspomnianymi wcześniej wyznacznikami, a temperaturą danego dnia i ryzykiem rozprzestrzenienia się pożaru. Przyjrzymy się bliżej temu jaśniejszemu kwadratowi.

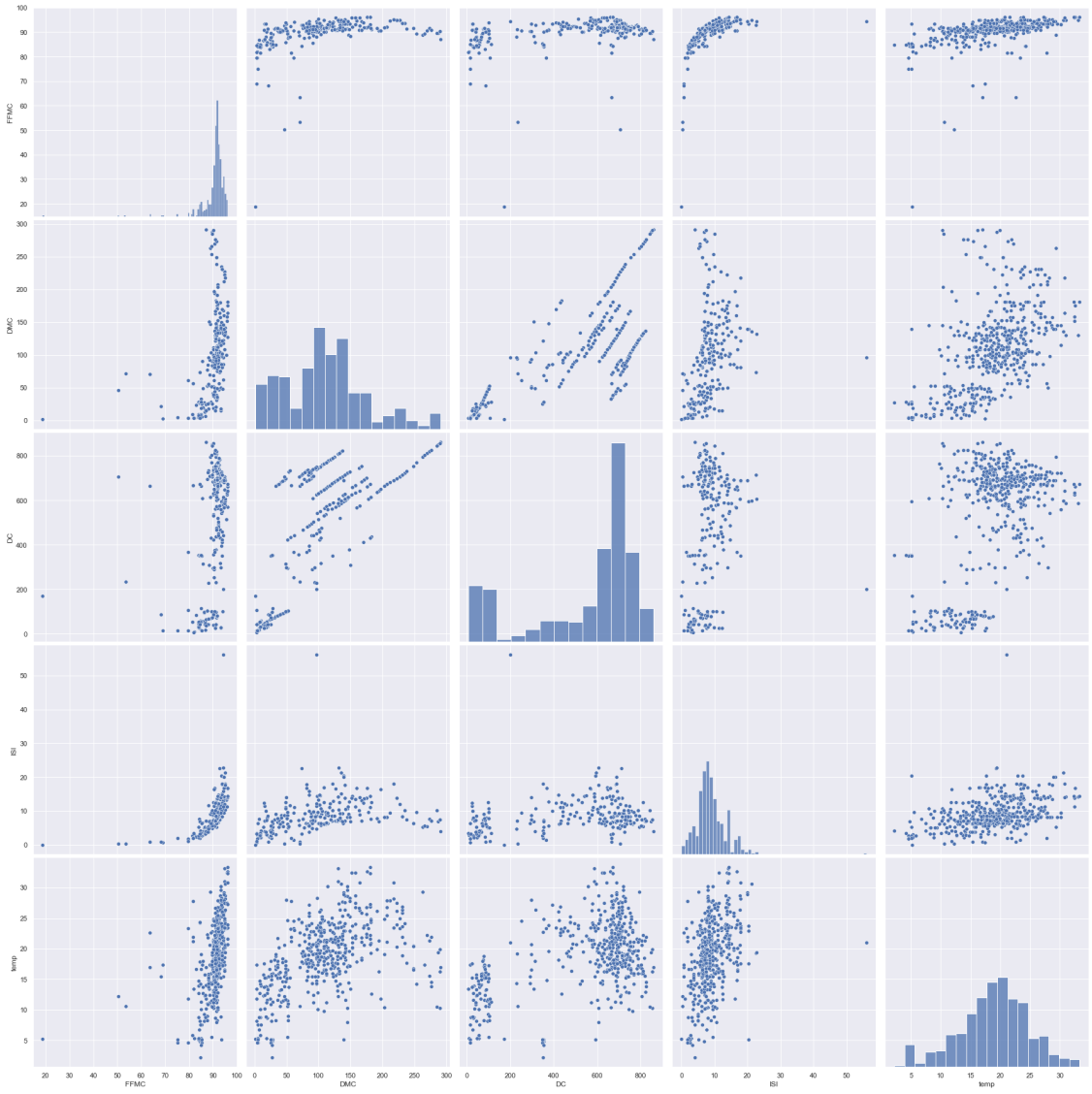
[18]:
```
fires_subset = fires_df[['FFMC','DMC','DC','ISI','temp']]
cm = np.corrcoef(fires_subset.values.T)
```

```
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f',␣
 ↪annot_kws={'size': 10}, yticklabels=fires_subset.columns,␣
 ↪xticklabels=fires_subset.columns)
plt.show()
```



Zmiennymi najbardziej skorelowanymi są DC i DMC. Najgorzej skorelowana z resztą wydaje się być zmienna ISI jednak wyjątkiem jest FFMC, która ma z ISI ma współczynnik korelacji 0.53.

[20]:
```
sns.set()
sns.pairplot(fires_subset, height = 5)
plt.show();
```



Widzimy, że dla FFMC i ISI istnieją rekordy o wartościach mocno odstających

[22]:
```
isi_max = fires_subset.loc[fires_subset['ISI']==fires_subset['ISI'].max()]
isi_max
```

[22]:

	FFMC	DMC	DC	ISI	temp
266	94.3	131.7	607.1	22.7	19.4

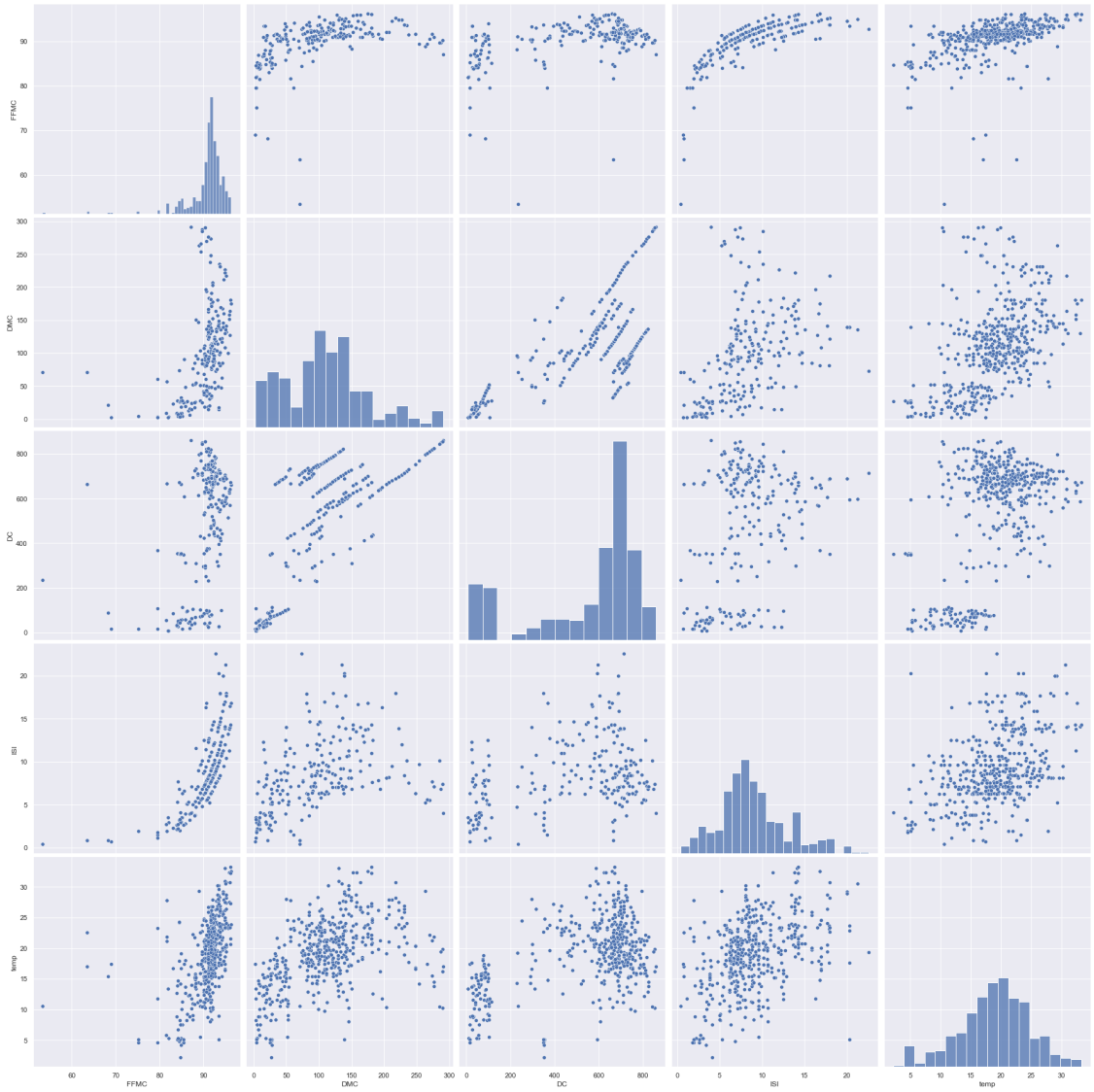
[24]:
```
ffmc_min = fires_subset.loc[fires_subset['FFMC']==fires_subset['FFMC'].min()]
ffmc_min
```

[24]:

	FFMC	DMC	DC	ISI	temp
312	50.4	46.2	706.6	0.4	12.2

W większości przypadku nie podążają one z pożądanymi przez nas trendami więc usuńmy je i zobaczmy czy zależności będą wyglądać inaczej.

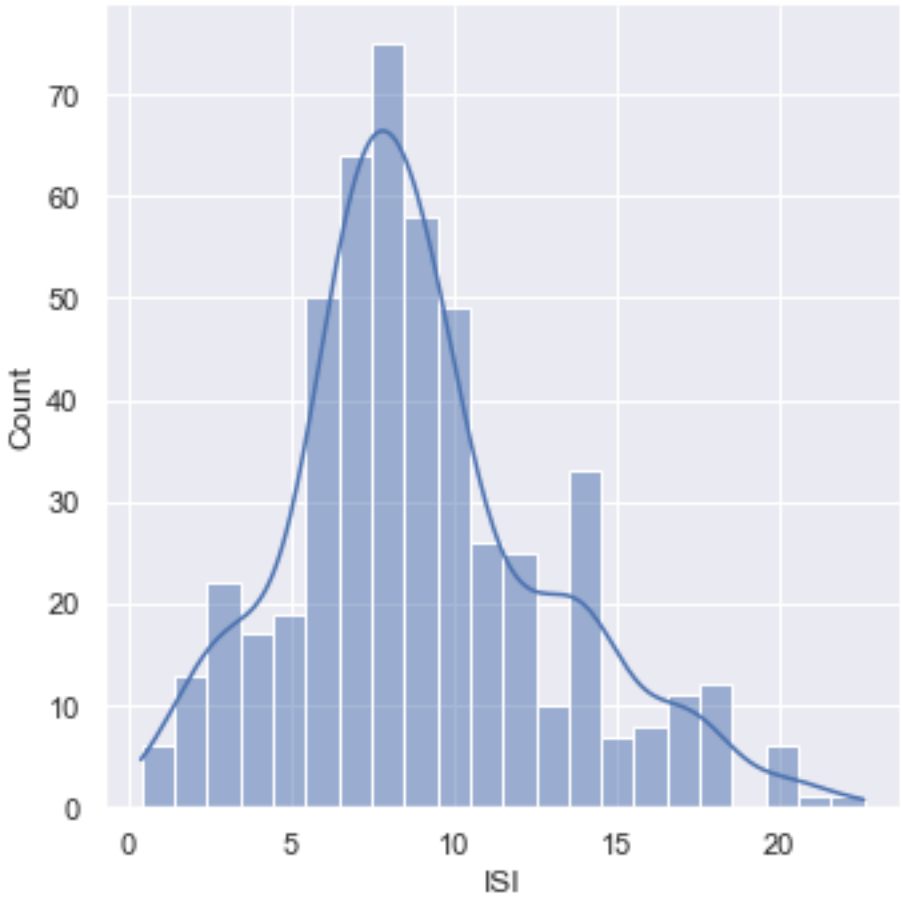
[26]:
```
fires_subset = fires_subset.loc[(fires_subset['ISI']!=fires_subset['ISI'].
 ↪max()) & (fires_subset['FFMC']!=fires_subset['FFMC'].min())]
sns.set()
sns.pairplot(fires_subset, height = 5)
plt.show();
```



Najbliżej rozkładu rozmalnego wydają się być wartośći temp i ISI. Potwierdzają się także nasze przewidywania, że najlepsze zależności widać pomiędzy DMC i DC. Przyjrzyjmy się teraz histogramom ICI i temp.

[28]:
```
sns.displot(fires_subset, x="ISI",kde=True)
```

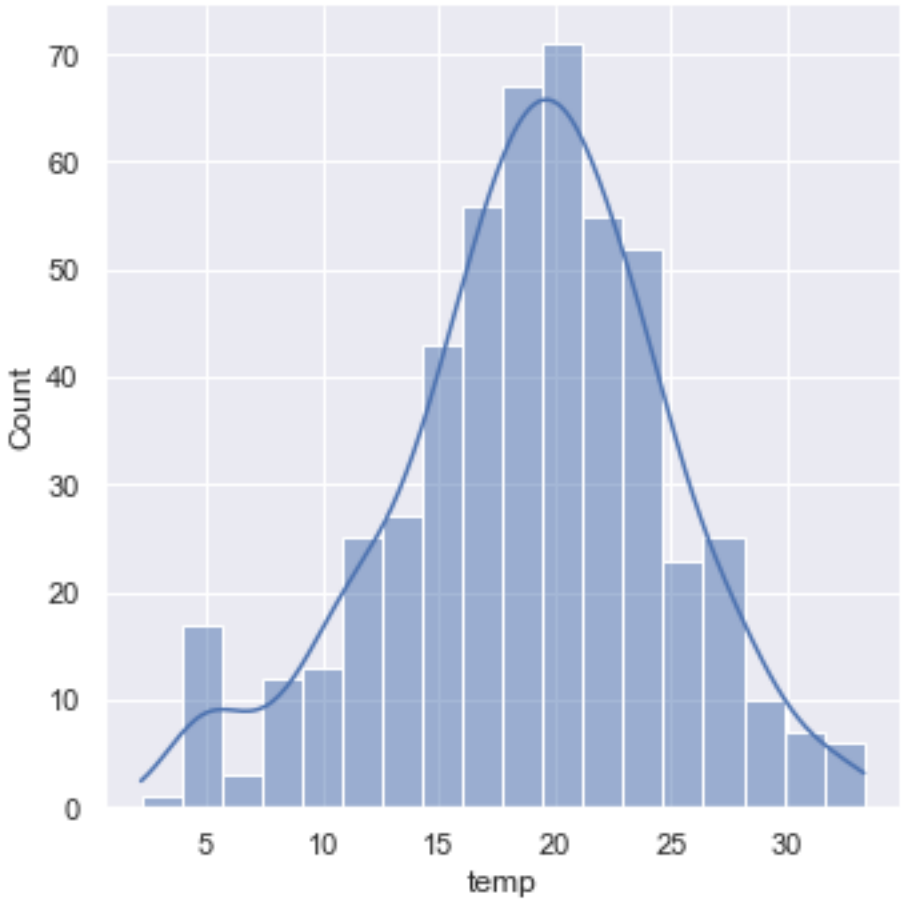
[28]:
```
<seaborn.axisgrid.FacetGrid at 0x2637ecf7850>
```



Widzimy, że rozkład jest leptokurtyczny i prawoskośny.

[30]:
```
sns.displot(fires_subset, x="temp",kde=True)
```

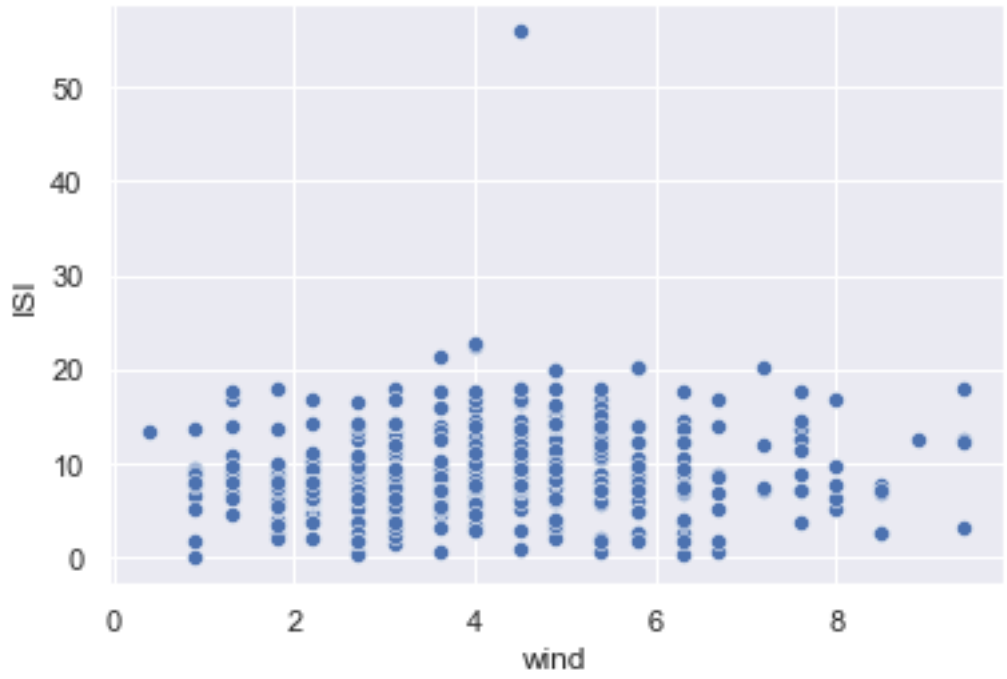
[30]:
```
<seaborn.axisgrid.FacetGrid at 0x2637d39fc40>
```



Widzimy, że rozkład jest leptokurtyczny i lewoskośny. Jednak jest on najbliższy rozkładowani normalnemu z badanych przez nas zmiennych. Dodatkowo poza badanym przez nas podbiorem danych sprawdzimy zależności pomiędzy wiatrem, a współczynnikiem ISI oznaczającym ryzyko rozprzestrzeniania się, które uważamy, że pomimo niskiego współczynnika korelacji powinny być od siebie zależne.

[32]:
```
sns.scatterplot(data = fires_df,x="wind",y="ISI")
```

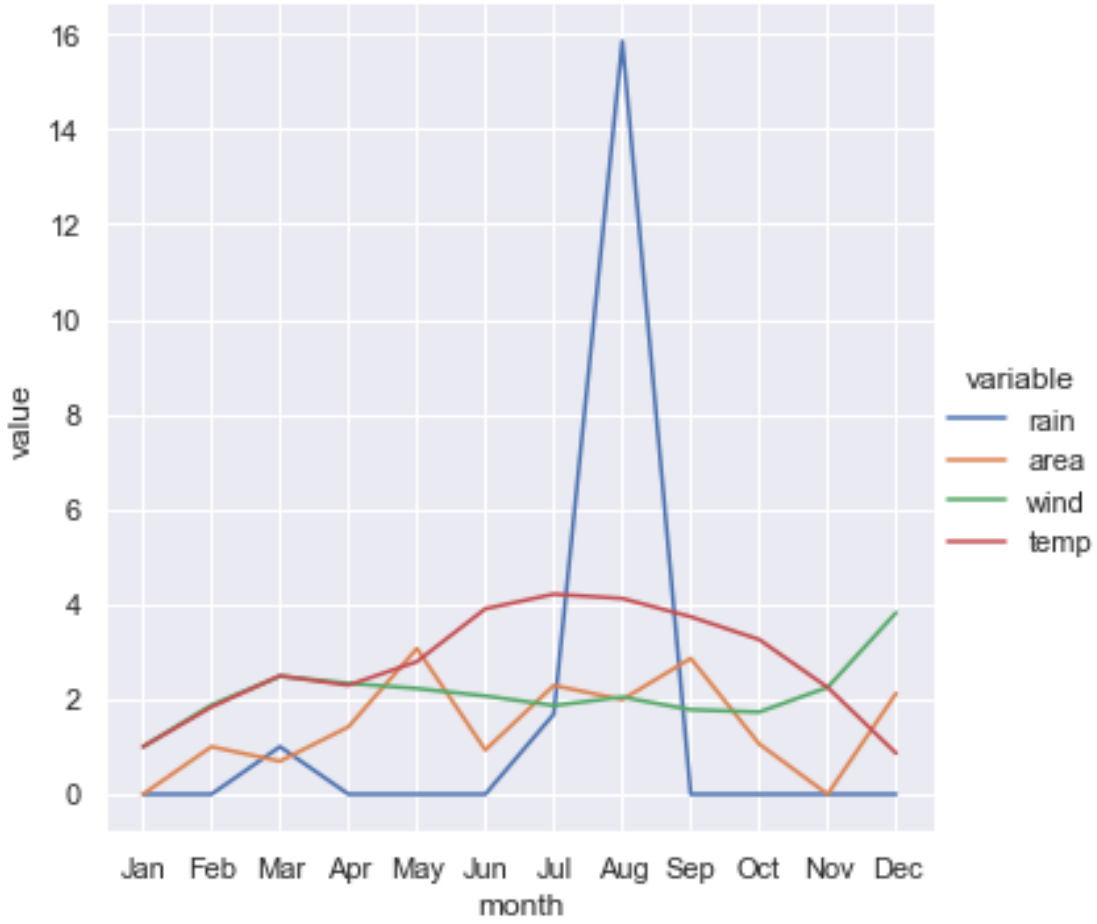
[32]:
```
<AxesSubplot:xlabel='wind', ylabel='ISI'>
```



Niestety nasze założenie się nie sprawdziło i żadnej zależności nie widać. Sprawdźmy jak pomiędzy miesiącami zmieniała się pogoda (temperatura, deszcz, wiatr) i czy wpływało to na obszar spalonego lasu.

[49]:
```
fires_df_grouped = fires_df[['month','rain','area','wind','temp']].
 ↪groupby('month')
fires_df_grouped = fires_df_grouped.mean()
fires_df_grouped = fires_df_grouped.reset_index()
fires_df_grouped['month'] = pd.
 ↪to_datetime(fires_df_grouped['month'],format="%b")
fires_df_grouped = fires_df_grouped.sort_values(by='month')
fires_df_grouped['rain'] = fires_df_grouped['rain'].
 ↪div(fires_df_grouped['rain'].iloc[2])
fires_df_grouped['temp'] = fires_df_grouped['temp'].
 ↪div(fires_df_grouped['temp'].iloc[0])
fires_df_grouped['wind'] = fires_df_grouped['wind'].
 ↪div(fires_df_grouped['wind'].iloc[0])
fires_df_grouped['area'] = fires_df_grouped['area'].
 ↪div(fires_df_grouped['area'].iloc[1])
fires_df_long = pd.melt(fires_df_grouped,id_vars␣
 ↪=['month'],value_vars=['rain','area','wind','temp'])
fires_df_long['month'] = fires_df_long['month'].dt.strftime('%b')
sns.relplot(data=fires_df_long,kind="line",x="month",y='value',hue="variable")
```

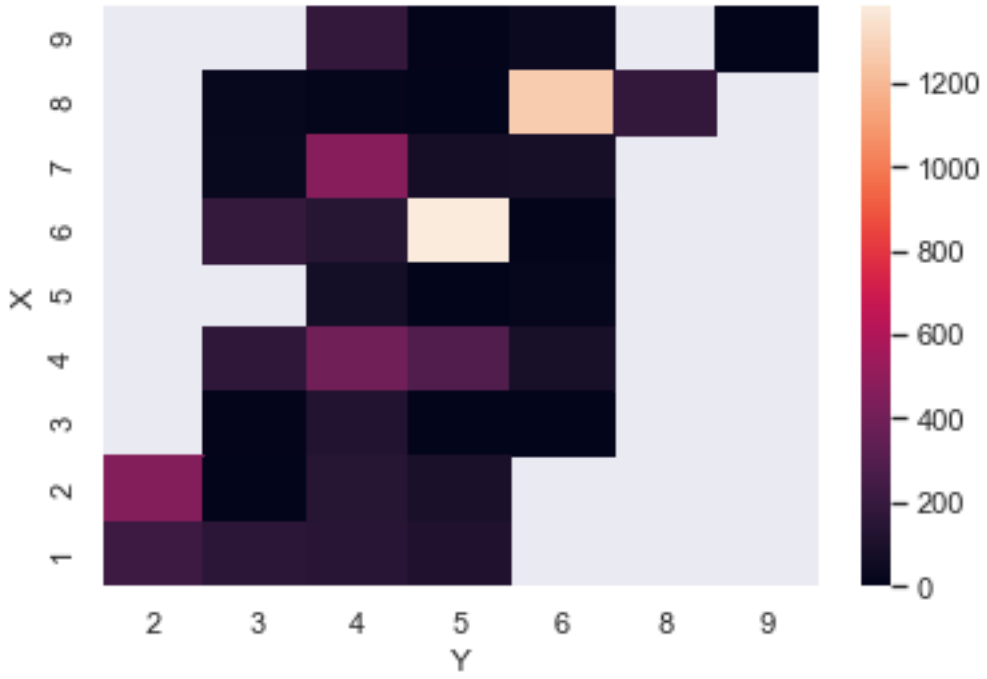
[49]: <seaborn.axisgrid.FacetGrid at 0x2637fbf3730>



Powyższy wykres przedstawia jak zmieniały się poszczególne wskaźniki pogodowe w danych miesiącach. Miedzy innymi widać, że w sierpniu gdy padało najwięcej deszczu, zmniejszył sie obszar spalonego lasu. Teraz sprawdźmy gdzie spłoneło go najwięcej.

[35]:
```
fires_area = fires_df[["X", "Y", "area"]]
fires_area = fires_area.groupby(["X","Y"])
fires_area = fires_area.sum()
fires_area = fires_area.reset_index()
fires_area= fires_area.pivot("X","Y","area")
fires_area= fires_area.sort_index(ascending=False)
sns.heatmap(fires_area)
```

[35]: <AxesSubplot:xlabel='Y', ylabel='X'>



X i Y to koordynaty, więc widzimy, że najwięcej pożarów było w centrum parku. Teraz do eksploracji skorzystamy z pandas profiling:

[36]:
```
profile = ProfileReport(fires_df, title='Fires Data Exploration')
profile
```

```
Summarize dataset:   0%|          | 0/26 [00:00<?, ?it/s]

Generate report structure:   0%|          | 0/1 [00:00<?, ?it/s]

Render HTML:   0%|          | 0/1 [00:00<?, ?it/s]

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

[36]:

Narzędzie pandas-profiling bardzo ułatwiwa eksploracje danych. Największą jego zaletą jest interaktywne szukanie zależności pomiędzy zmiennymi oraz podsumowania wraz ze szczegółami dla każdej zmiennej. Bardzo przyspieszyło to analize naszego zbioru danych. Do ograniczeń należy między innymi brak możliwości porównywania między sobą zmodyfikowanych zbiorów danych na przykład zbioru oryginalnego ze zbiorem z usuniętymi obserwacjami odstającymi.