spytek_mikolaj_pd1

March 8, 2021

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from pandas_profiling import ProfileReport
sns.set()
```

1 Praca domowa 1

Eksploracja zbioru danych dotyczącego pożarów lasów w północno-wschodnej Portugalii. O parku można poczytać sobie tutaj, jest też mapka, o której mowa w opisie kolumn na stronie, z której pobraliśmy dataset.

[2]:
```
#wczytanie danych do ramki
fires_df = pd.read_csv("https://lovespreadsheet-tutorials.s3.amazonaws.com/
↪APIDatasets/forest_fires_dataset.csv")

#informacje o kolumnach
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
```

Zmienną wyjaśnianą w tym przypadku jest `area`, czyli 12-sta kolumna ramki, a pozostałe kolumny to zmienne wyjaśniające.

Zauważamy też od razu, że wszystkie pola w ramce danych są niepuste, więc nie musimy martwić się ani zajmować wypełnianiem braków.

Warto sprawdzić co poszczególne zmienne określają. W szczególności te, które nazwane są tylko tajemniczym skrótem. Informacje o systemie Fire Weather Index, zaczerpnąłem ze strony https://cwfis.cfs.nrcan.gc.ca/background/summary/fwi. Mimo, że jest to strona organizacji kanadyjskiej, a mamy analizować dane dotyczące Portugalii, to sam system FWI jest światowym standardem, więc opisy poszczególnych wskaźników powinny być spójne. Na tej stronie były opisane w sposób najbardziej przystępny.

Mamy więc:

- `X`, `Y` (zmienne kategoryczne) - są to współrzędne obszaru w parku Montesinho, określają więc o której części analizowanego obszaru mamy dane w tym rekordzie. Obie zmenne to liczby naturalne do 9;
- `month` (zmienna kategoryczna) - słowna nazwa miesiąca;
- `day` (zmienna kategoryczna) - słowna nazwa dnia tygodnia;
- `FFMC` (zmienna numeryczna) - skrót rozwija się do Fine Fuel Moisture Code, jest to wskaźnik, który określa wilgotność ściółki w lesie. Im wyższa jego wartość, tym bardziej mokra ściółka, z czego możemy wnioskować, że być może tym mniejsze ryzyko pożaru;
- `DMC` (zmienna numeryczna) - ten skrót rozwija się do Duff Moisture Code. Wskaźnik ten określa wilgotność warstw organicznych średniej głębokości, bierze więc pod uwagę dłuższy trend w suchości podłoża. Aby głębsze warstwy wyschły, brak opadów musi być ciągły. Tak jak z poprzednim wskaźnikiem większe wartości oznaczają większą wilgotność podłoża;
- `DC` (zmienna numeryczna) - ten wskaźnik, którego pełna nazwa to Drought Code, jest bardzo podobny do poprzednich dwóch. Określa on jednak wilgotność w głębokich warstwach podłoża. Może więc być używany jako indykator wilgotności na poziomie pór roku, a nie poszczególnych dni, czy tygodni. Podobnie do poprzednich dwóch zmiennych, wyższe wartości oznaczają większą wilgotność podłoża;
- `ISI` (zmienna numeryczna) - ostatni ze wskaźników pochodzących z systemu FWI. Ta zmienna (Initial Spread Index) jest liczbą określającą jak szybko pożar będzie się prawdopodobnie rozprzestrzeniał, określoną na podstawie warunków atmosferycznych. Im wyższe wartości, tym ogień prawdopodobnie będzie się szybciej rozszerzał;
- `temp` (zmienna numeryczna) - temperatura w stopniach Celsiusza;
- `RH` (zmienna numeryczna) - względna wilgotnosc powietrza w %;
- `wind` (zmienna numeryczna) - prędkość wiatru w km/h;
- `rain` (zmienna numeryczna) - ilość opadów w mm/m^2.

Natomiast zmienna wyjaśniana `area` to obszar spalonego lasu w hektarach.

Z opisu wyznaczania wskaźników z systemu FWI dowiadujemy się jeszcze, że większość z nich, jako składową zawiera w sobie, temperaturę, względną wilgotność, prędkość wiatru i poziom opadów. Ponieważ jednak mamy też te dane w ramce, przy dalszej analizie trzeba wziąć pod uwagę fakt, że są one zależne, nawet jeśli nie będzie to prosta liniowa, czy kwadratowa zależność, którą możemy zobaczyć na wykresie.

Na podstawie tego krótkiego rozeznania możemy dojść też do wniosku, że z pierwszych trzech wskaźników z FWI, to prawdopodobnie ten pierwszy, czyli FFMC, będzie najlepiej nam określał ryzyko pożaru, gdyż to wierzchnia warstwa ściółki jest narażona na zaprószenie ognia.

W tym miejscu warto wspomnieć, że wszystko, czego dokonuję poniżej raczej powinno być robione już po podziale zbioru, wyłącznie na zbiorze treningowym, a nie na całym datasecie. Jednak jako, że nic na ten temat nie było wspomniane w opisie pracy domowej, oraz nie omawialiśmy jeszcze metod podziału zbioru, zaznaczam tu tylko, że traktuję cały zbiór jako zbiór treningowy, zakładając, że mamy gdzieś odłożony już zbiór testowy.

[3]:
```
#podstawowe wartości opisujące poszczególne zmienne
fires_df.describe()
```

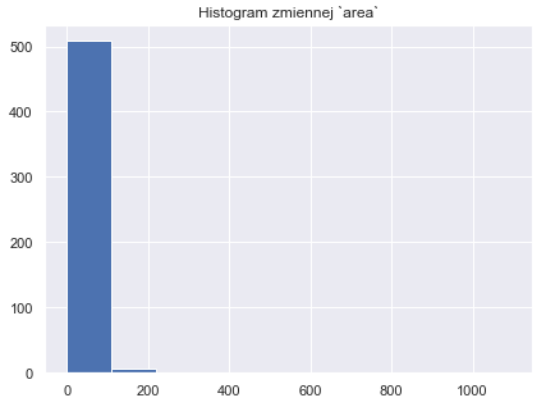
[3]:

	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	

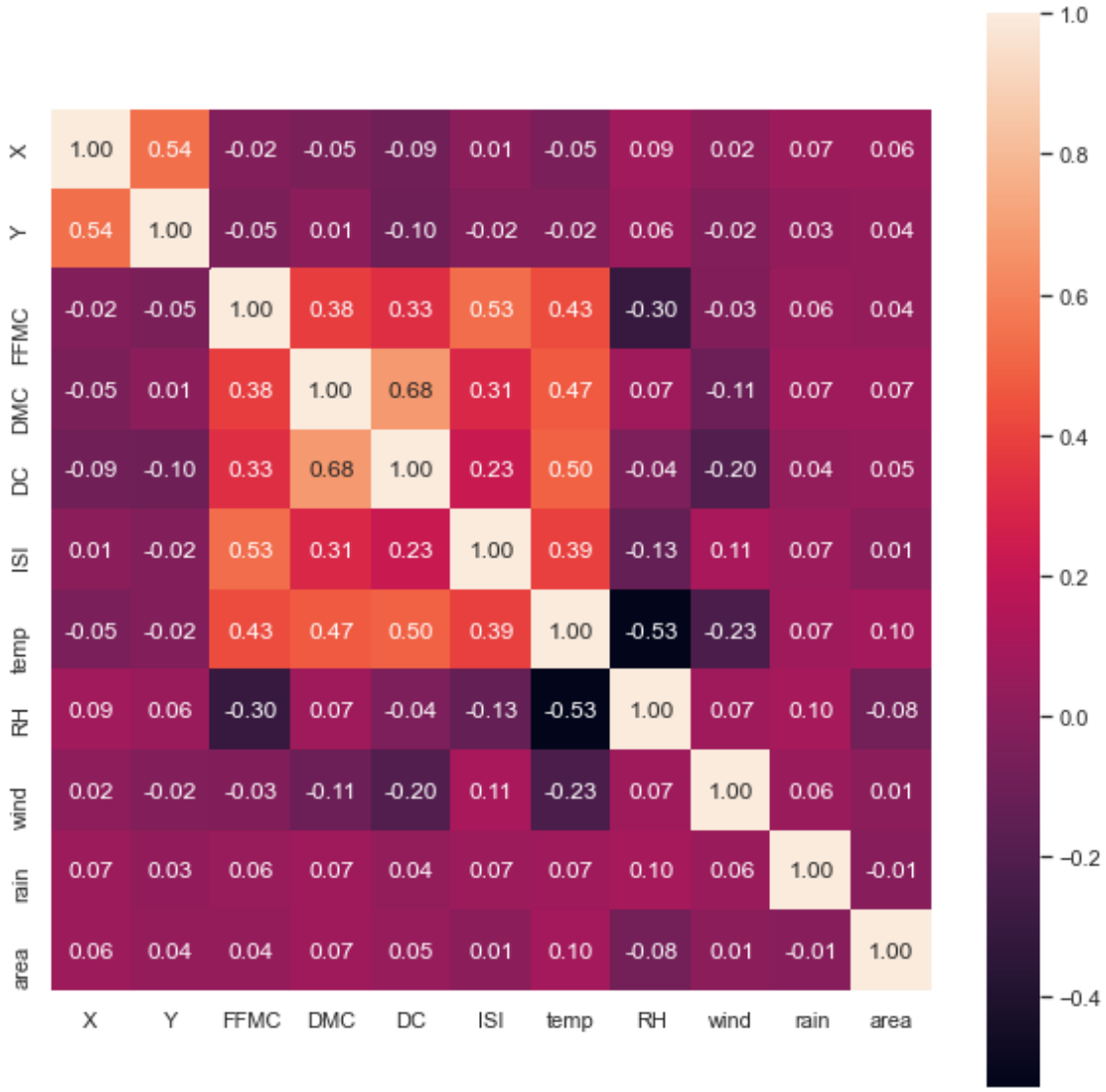
	temp	RH	wind	rain	area
count	517.000000	517.000000	517.000000	517.000000	517.000000
mean	18.889168	44.288201	4.017602	0.021663	12.847292
std	5.806625	16.317469	1.791653	0.295959	63.655818
min	2.200000	15.000000	0.400000	0.000000	0.000000
25%	15.500000	33.000000	2.700000	0.000000	0.000000
50%	19.300000	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

Zarówno z powyższej tabelki, jak i z opisu zbioru danych wynika, że nasza zmienna wyjaśniana jest bardzo skośna w kierunku zera. Gdybyśmy więc mieli trenować jakiś model, to prawdopodobnie łatwiej by było przewidzieć log(area), niż samo area. Pokazują to następujące histogramy:

[4]:
```
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(15,5))
ax1.hist(fires_df['area'])
ax1.set_title("Histogram zmiennej `area`")
ax2.hist(np.log1p(fires_df['area']))
ax2.set_title("Histogram logarytmu zmiennej `area`")
plt.show()
```



[5]:
```
#popatrzmy na heatmapę korelacji poszczególnych zmiennych
plt.figure(figsize=(10,10))
sns.heatmap(fires_df.corr(), square=True, annot=True, fmt=".2f")
plt.show()
```

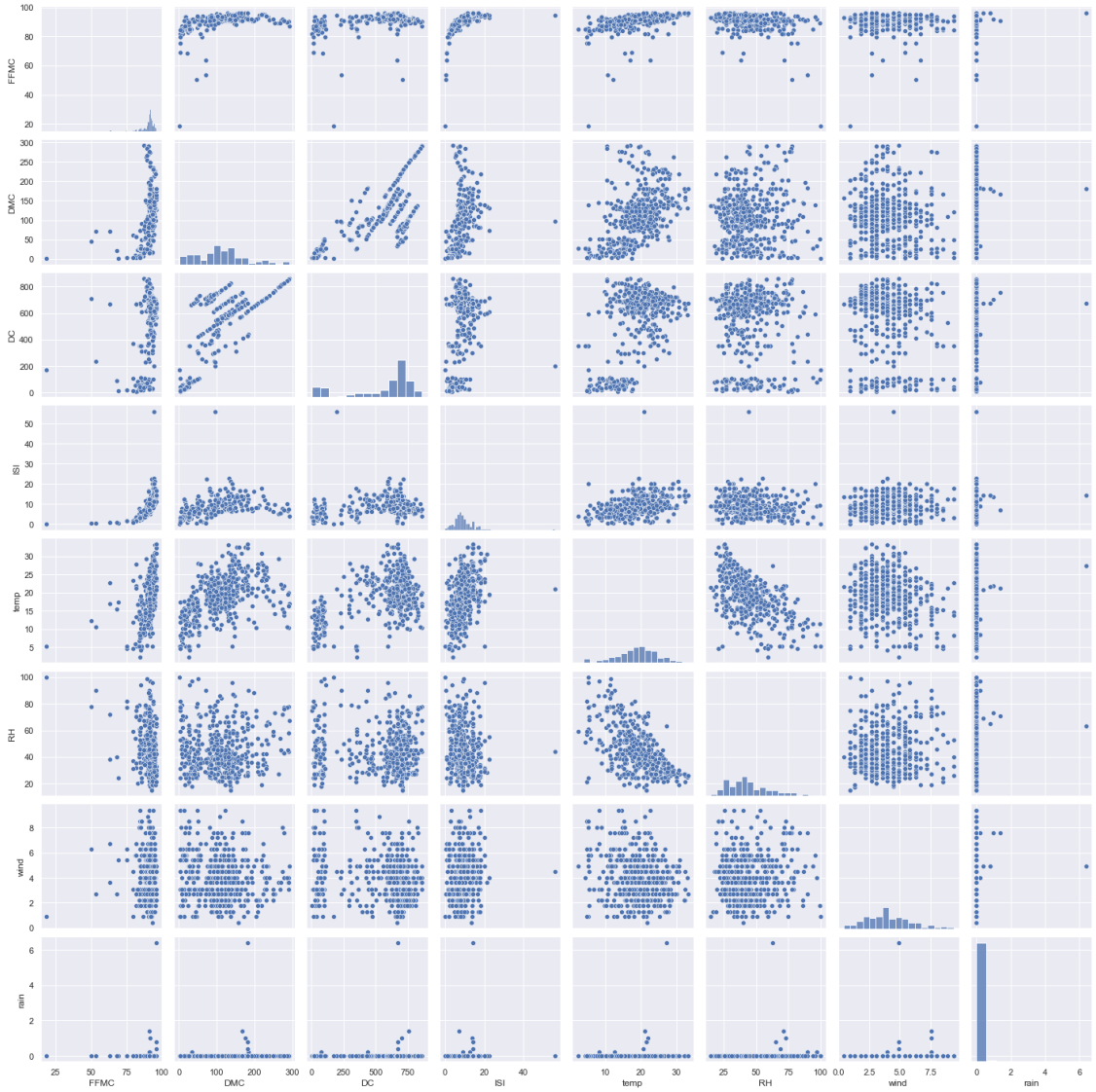


Wyraźnie widać, że wsystkie wskaźniki pochodzące z systemu FWI są ze sobą dość znacząco skorelowane, więc do modelu prawdopodobnie powinniśmy wybrać tylko jeden z nich, są one dodatkowo skorelowane z temperaturą, więc jeśli zdecydujemy się wziąć któryś z tych wskaźników, to prawdopodobnie nie powinniśmy brać kolumny z temperaturą.

Poza nimi, na pierwszy rzut oka wyróżniają się jeszcze dość silne korelacje: wilgotności wzgldnej z FFMC, oraz wilgotności względnej z temperaturą. To też trzeba mieć na uwadze wybierając kolumny do trenowania.

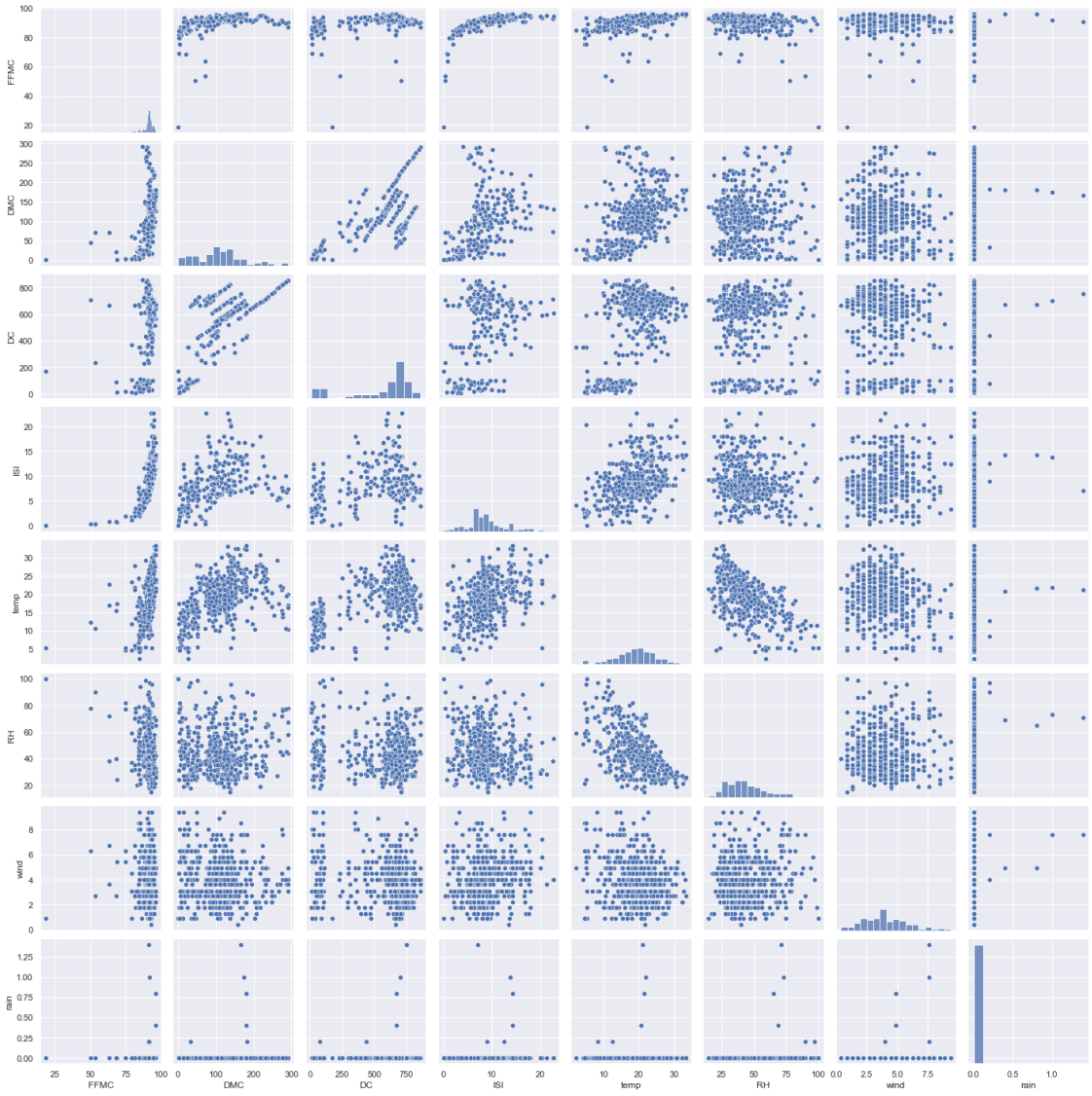
Oprócz takiej prostej heatmapy, warto poszukać bardziej skomplikowanych zależności. Przecież korelacja Pearsona nie powie nam nic, jeśli zmienne pozostaną powiązane np. kwadratowo, czy logarytmicznie.

[6]:
```
# zobaczmy, jak zmienne wyjaśniające są ze sobą powiązane
# sns.pairplot(fires_df.drop('area', axis=1))  # nie pokazuję tego wykresu bo
 ↪następny jest podobny

# Przede wszystkim z tych wykresów widzimy, że współrzędne `X` i `Y` nie mówią
 ↪nam nic, ponieważ wyglądają na zupełnie
# losowo rozrzucone jeśli mamy kilka rekordów na temat tego samego obszaru, to
 ↪są nałożone na siebie
# Aby nie utrudniały więc dalszej analizy, zróbmy taki sam wykres tylko bez
 ↪tych zmiennych.
sns.pairplot(fires_df.drop(['area', "X", "Y"], axis=1))
plt.show()
```



Jest już lepiej, łatwiej się przyglądać zależnościom. Przeszkadza jeszcze tylko jedna rzecz. W szczególności w czwartym i ostatnim wierszu widzimy pojedynczą obserwację odstającą, o kilka rzędów wielkości od pozostałych. Przez nią zaburzona jest skala wykresu i nie do końca możemy dobrze ocenić korelację. Wiec na potrzeby tej analizy, ukryjemy je, aby łatwiej było przyglądać się wykresom.

[7]:
```
dropped_columns = fires_df.drop(['area', "X", "Y"], axis=1)
hidden_outliers = dropped_columns.loc[(dropped_columns['rain']<=6) &␣
 ↪(dropped_columns['ISI']<40)]

sns.pairplot(hidden_outliers)
plt.show()
```

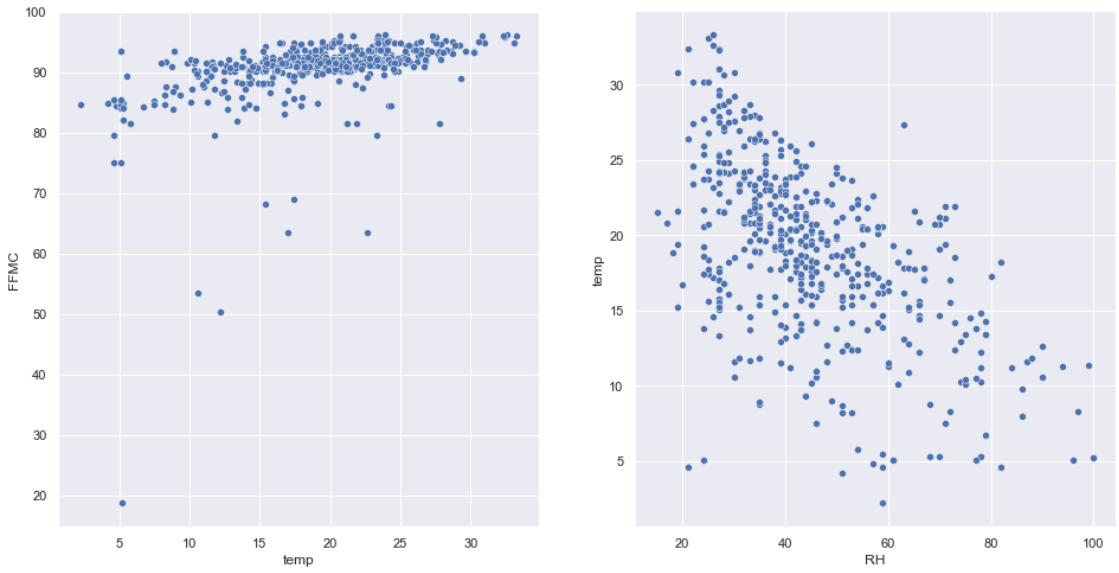


Z tych wykresów widzimy, że zmienne nie są ze sobą skorelowane w jakiś bardziej skomplikowany

sposób, ale potwierdzają się też nasze przypuszczenia, które wysnuliśmy na podstawie korelacji Pearsona. Widzimy, że bardzo mocno powiązane są ze sobą współczynniki z FWI. Widzimy też, że ten, który na początku wydawał się nam najodpowiedniejszy (FFMC) do modelowania spalonej powierzchni jest silnie i prawie liniowo skorelowany z **prawie wszystkimi** innymi zmiennymi wyjaśniającymi, co jest dość problematyczne - jeśli weźmiemy do trenowania tę kolumnę, to nie powinniśmy brać innych z nią zależnych, czyli prawie żadnych. A trenowanie modelu na jednej zmiennej wydaje się być jednak trochę komiczne i niepraktyczne.

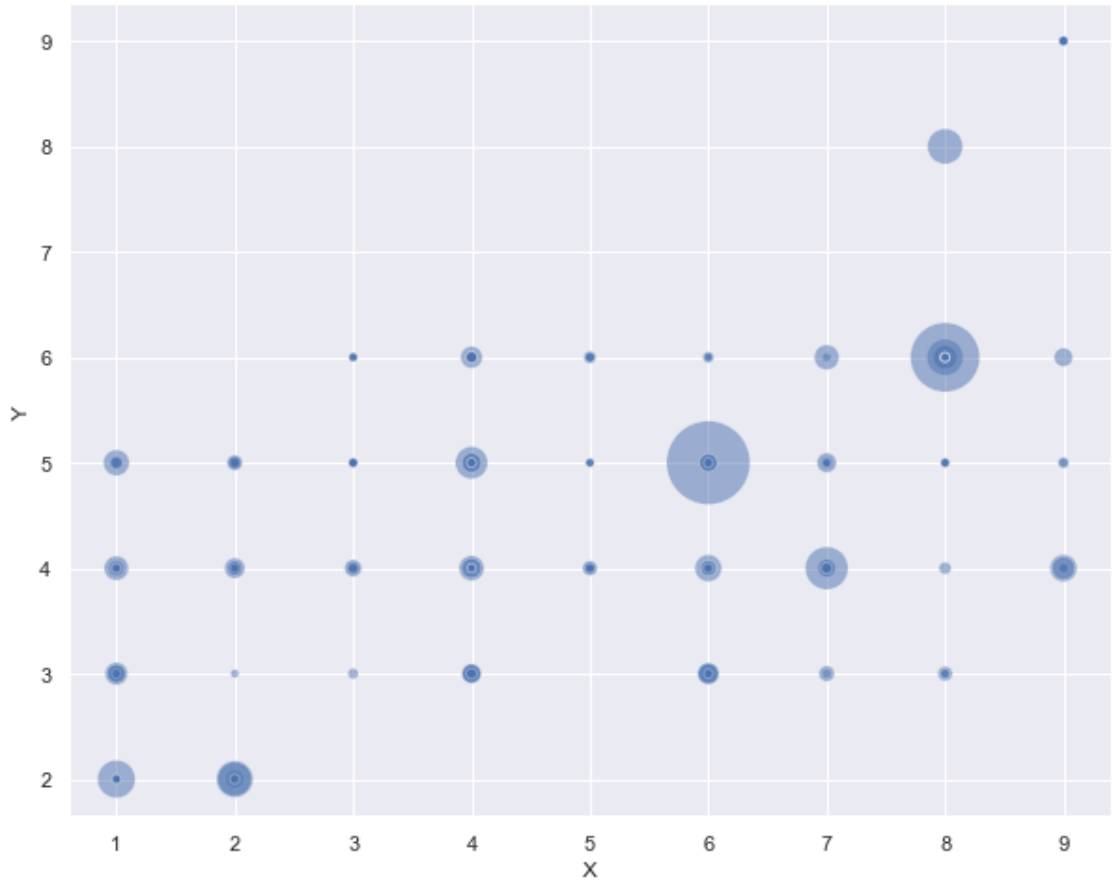
Możemy spróbować kilku wykresom przyjrzeć się bliżej. Pokażę tu dwa wykresy zmiennych o dużej korelacji:

[8]:
```
fig, (ax1, ax2) = plt.subplots(1,2, figsize=(16,8))
sns.scatterplot(data=fires_df, x="temp", y="FFMC", ax=ax1)
sns.scatterplot(data=fires_df, x="RH", y="temp", ax=ax2)
plt.show()
```



Warto też popatrzeć na informację jak przedstawiało się rozmieszczenie pożarów - informacja, której na powyższych wykresach nie moglibyśmy zobaczyć:

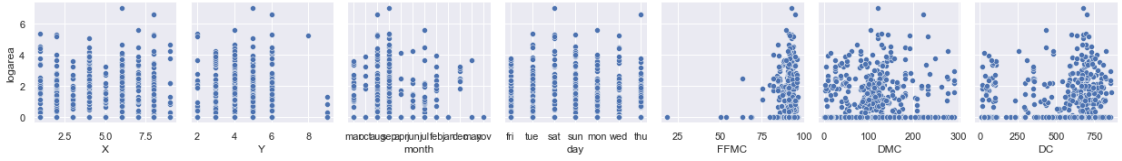
[9]:
```
plt.figure(figsize=(10,8))
sns.scatterplot(data=fires_df, x="X", y="Y", size="area", sizes=(20, 2000),␣
↪legend=None, alpha=0.5)
plt.show()
```

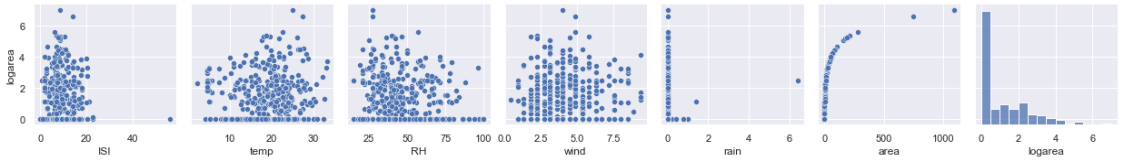


Widzimy, że bardzo ścicłych zależności nie ma, lecz niektóre obszary płoną częściej, możemy użyć tych zmiennych do naszego modelu.

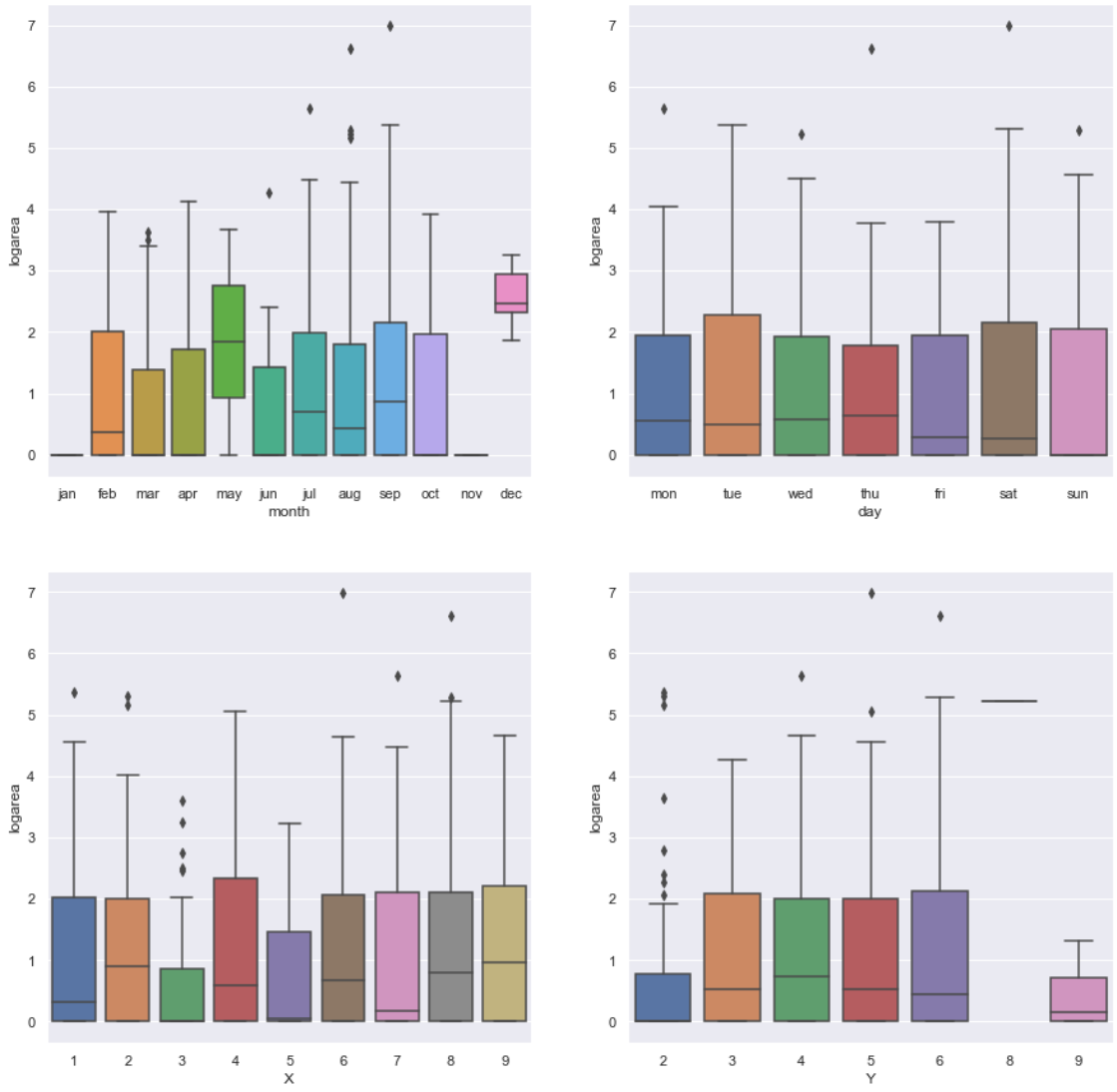
Kolejnym krokiem byłoby więc sprawdzenie, jak zmienne wyjaśniające są powiązane ze zmienną wyjaśnianą, ale popatrzymy na logarytm powierzchni, tak jak zasugerował autor zbioru danych. Jeśli tego nie zrobimy, to prawie nic nie zauważymy. Sporzymy na scatterploty, a dla zmiennych kategorycznych dodatkowo na boxploty:

[10]:
```
fires_df["logarea"] = np.log1p(fires_df["area"])
sns.pairplot(fires_df, y_vars="logarea", x_vars=fires_df.columns.values[:7])
sns.pairplot(fires_df, y_vars="logarea", x_vars=fires_df.columns.values[7:])

plt.show()
```





[11]:
```
fig, axes = plt.subplots(2,2, figsize=(15,15))
sns.boxplot(data=fires_df, x="month", y="logarea", ax = axes[0,0],␣
 ↪order=['jan', 'feb', 'mar', 'apr',␣
 ↪'may','jun','jul','aug','sep','oct','nov','dec'])
sns.boxplot(data=fires_df, x="day", y="logarea", ax= axes[0,1],␣
 ↪order=['mon','tue', 'wed', 'thu', 'fri', 'sat', 'sun'])
sns.boxplot(data=fires_df, x="X", y="logarea", ax= axes[1,0])
sns.boxplot(data=fires_df, x="Y", y="logarea", ax= axes[1,1])

plt.show()
```



Niestety widzimy, że z danych nie wynika nic oczywistego. Na pewno żadna ze zmiennych wyjaśniających nie ma prostej zależności ze zmienną wyjaśnianą. Możemy metodą eliminacji odrzucać poszczególne zmienne. - `X`, `Y`, oraz `day` wydają się być równomiernie rozłożone więc pewnie nie dadzą nam dobrej predykcji. Z nich rezygnujemy. - `rain` również wydaje się nic nie mówić o powierzchni spalonego lasu. Większość danych w tej kolumnie to zera, albo wartości bardzo bliskie

Nie nasuwa nam się jasny podzbiór zmiennych, których powinniśmy użyć do uczenia modelu. Możemy jednak spróbować kilku różnych i popatrzeć na wyniki. Moje propozycje to:

- Zestaw I: `month`, `temp`, `RH`, `wind`.
- Zestaw II: `X`, `Y`, `month`, `DC`, `ISI`.
- Zestaw III: `X`, `Y`, `wind`, `DMC`, `temp`.

1.1 Pakiet do automatycznej eksploracji

Użyję pakietu `pandas_profiling`, do przeprowadzenia automatycznej eksploracji. Można przewidywać, że część pracy, którą wykonaliśy ręcznie będzie zrobiona szybciej, i mniejszym kosztem, lecz pewnie będą jakieś wady tego podejścia.

[12]:
```
profile = ProfileReport(fires_df, title="Pandas Profiling Report")
# polecenie do wyświelenia raportu w jupyterze
profile.to_notebook_iframe()
```

```
Summarize dataset:   0%|          | 0/27 [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>
```

[13]:
```
# ponieważ oddajemy pracę domową w .pdf, to załączę dodatkowy plik z raportem
profile.to_file("pandas_profiling_report.html")
```

```
Export report to file:   0%|          | 0/1 [00:00<?, ?it/s]
```

Dostajemy HTMLowy, interaktywny container w którym możemy sobie większość z ręcznie wygenerowanych danych wyklikać. Niestety przy zapisie do pdf'a nie generuje się on, więc pokrótcę opiszę, co możemy tam znaleźć.

W raporcie dostajemy między innymi: - rozkłady i charakterystyki wszystkich zmiennych, - możliwość stworzenia scatterplotów opisujących pary zmiennych, - heatmapę korelacji, - informacje o zduplikowanych wierszach.

Posiadanie takiego narzędzia jest wygodne - to, co wcześniej wykonaliśmy za pomocą kilkunastu linii kodu, możemy dzięki tej bibliotece otrzymać, pisząc tylko jedną linię.

Jednak ma to pewne ograniczenia, co widać już na przykładzie czynności, które wcześniej wykonaliśmy ręcznie:

- przede wszystkim próbowaliśmy modelować logartym z naszej zmiennej wyjaśnianej, co w łatwy sposób można było podmienić. Jednak kiedy korzystamy z takiej biblioteki, tracimy taką możliwość.
- przy oglądaniu scatterplotów zauważyliśmy, że jeden punkt 'psuł' nam skalę i nie można było zauważyć ogólnego trendu. Kiedy wszystko robimy ręcznie, łatwo jest go tymczasowo 'ukryć'.
- dla zmiennych kategorycznych scatterploty mają mało sensu, przynajmniej w formie domyślnej, generowanej przez matplotliba, czy seaborna - punkty nakładają się na siebie i nie widać, gdzie jest ich najwięcej - nie możemy poznać ich rozkładu. Do tego celu lepiej sprawdzają się boxploty.

1.2 Podsumowanie

Patrząc na wszystko co wykonaliśmy możemy dojść do następujących wniosków.

- Ten zbiór danych jest dość trudny - ciężko jednoznacznie wybrać cechy do uczenia modelu.
- Zmienne z systemu FWI są ze sobą mocno powiązane.

- Pierwsze intuicje i przeczucia bywają błędne. Ilość deszczu oraz temeratura mają mniejszy wpływ na pożary niż mogłoby się wydawać.
- Narzędzia automatyczne są fajne, ale nie można ich w prosty sposób dostosować do konkretnej potrzeby.

Mikołaj Spytek, 08.03.2021