WUM_PD1

March 8, 2021

0.1 Zbiór danych

Dane pochodzą ze zbioru: https://www.apispreadsheets.com/datasets/129.

Przedstawiają one informacje meteorologiczne dotyczące pożarów lasu w 'Parque Natural de Montesinho' w Portugali.

0.2 Preprocessing

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

[2]:
```
month_order = ['jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug', 'sep',␣
↪'oct', 'nov', 'dec']
day_order = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun']
fires_data = pd.read_csv("forest_fires_dataset.csv")
fires_data['area_log'] = fires_data['area'].apply(np.log1p)
fires_data.columns
```

[2]:
```
Index(['X', 'Y', 'month', 'day', 'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH',
       'wind', 'rain', 'area', 'area_log'],
      dtype='object')
```

0.3 Eksploracyjna Analiza Danych

[3]:
```
fires_data.head()
```

[3]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	area_log
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51.0	6.7	0.0	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	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	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	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	0.0

```
0       0.0
1       0.0
2       0.0
3       0.0
4       0.0
```

[4]:
```
fires_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 14 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
 13  area_log  517 non-null    float64
dtypes: float64(10), int64(2), object(2)
memory usage: 56.7+ KB
```

[5]:
```
fires_data.describe()
```

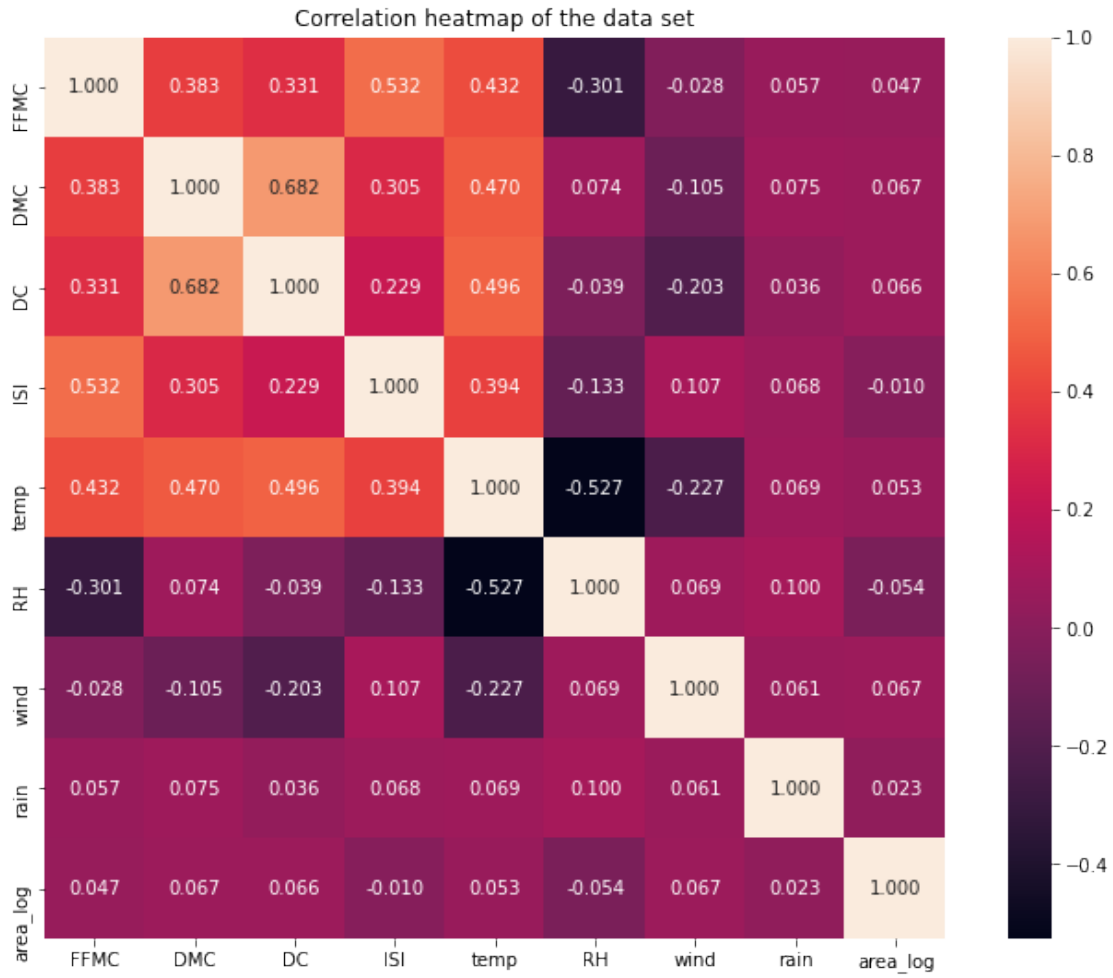
[5]:

	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	

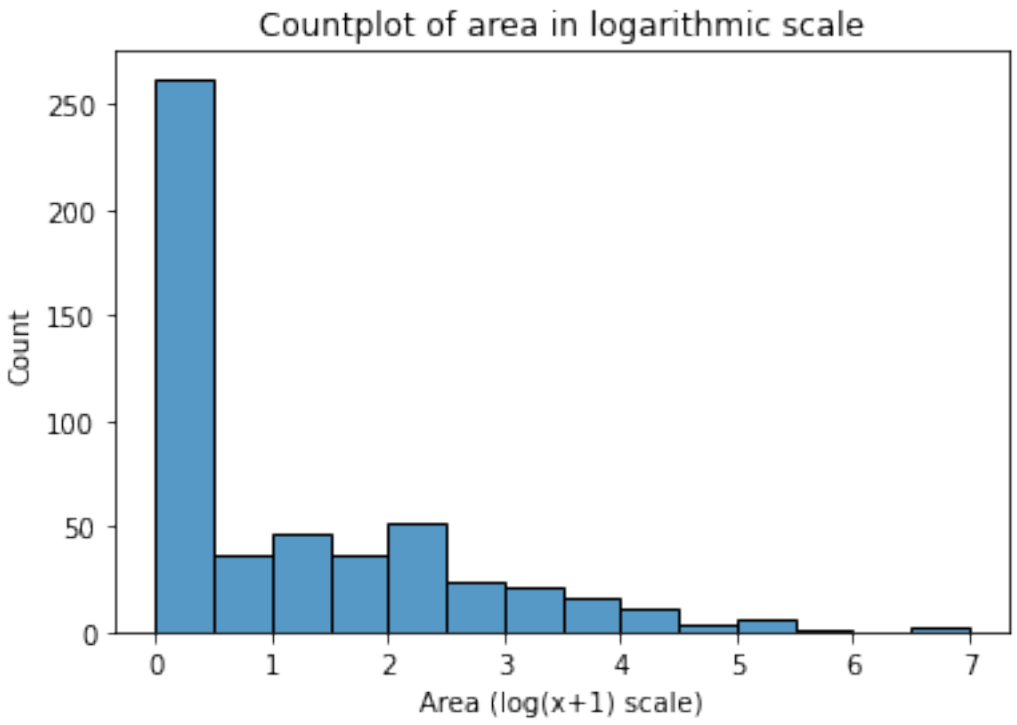
	temp	RH	wind	rain	area	area_log
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000
mean	18.889168	44.288201	4.017602	0.021663	12.847292	1.111026
std	5.806625	16.317469	1.791653	0.295959	63.655818	1.398436
min	2.200000	15.000000	0.400000	0.000000	0.000000	0.000000

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

[6]:
```
temp_data = fires_data.drop(["X", "Y", "area"], axis=1)
corre = temp_data.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corre, annot=True, square=True, fmt='.3f')
plt.title("Correlation heatmap of the data set")
plt.show()
```



[7]:
```
sns.histplot(data=fires_data, x="area_log")
plt.title("Countplot of area in logarithmic scale")
plt.xlabel("Area (log(x+1) scale)")
plt.show()
```



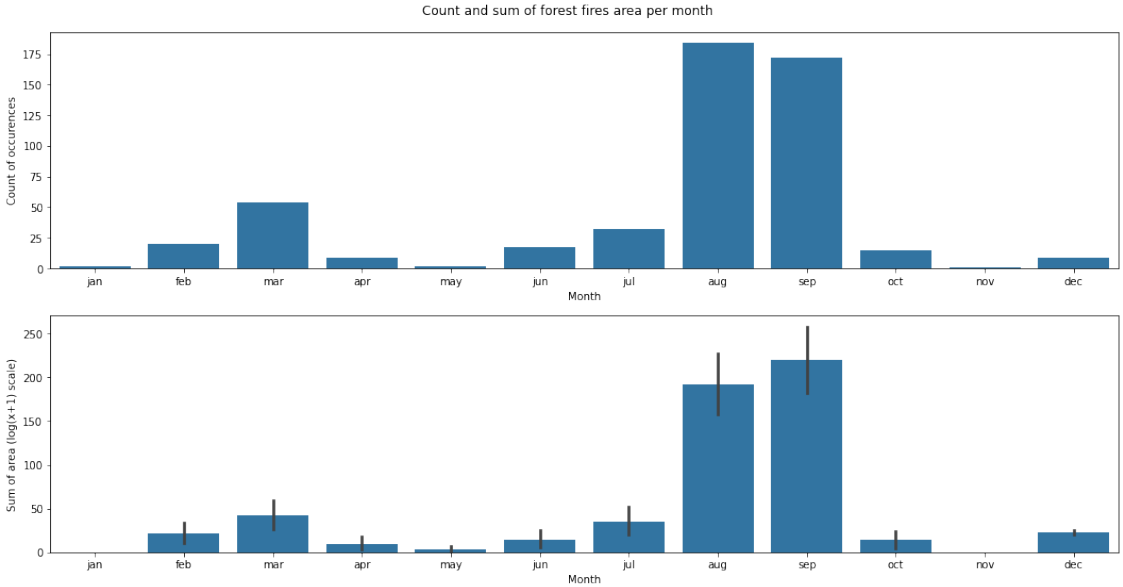
Zauważmy bardzo dużo wartości w pobliżu 0, postanowiłem sprawdzić ile ich jest równe 0.

[8]:
```
z = len(fires_data.loc[fires_data["area"] == 0])/len(fires_data["area"])
print(f"Area is equal to 0 in {z*100:.2f}% of rows")
```

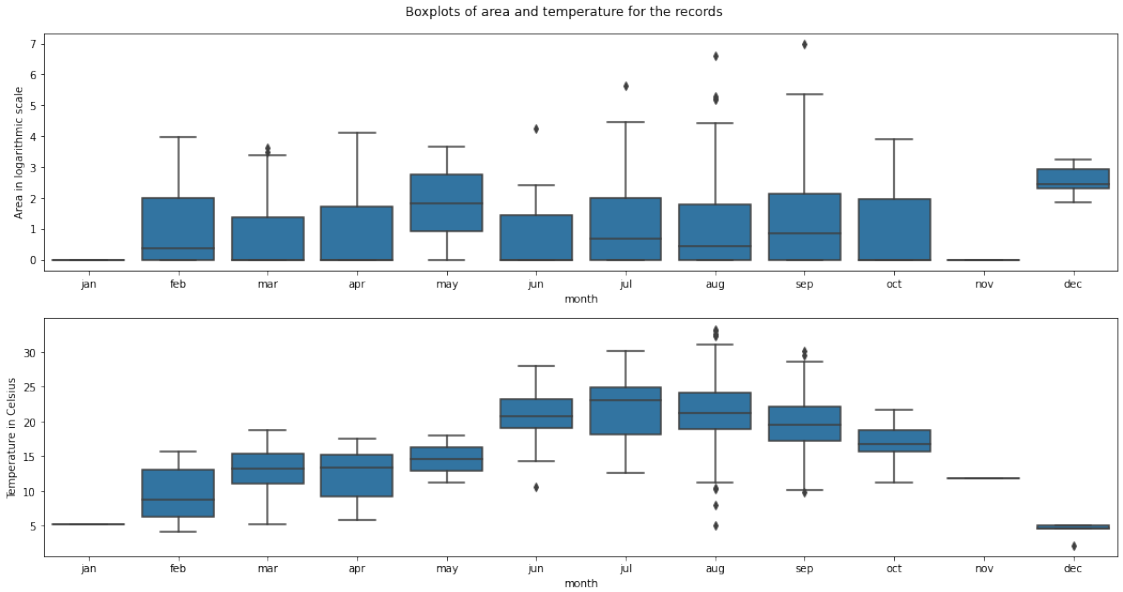
```
Area is equal to 0 in 47.78% of rows
```

Okazuje się że jest to prawie 50% danych. Pytanie czy dane te reprezentują brak pożaru, czy o powierzchni na tyle małej że nie została zmierzona. Jednak w pierwszym przypadku spodziewałbym się danych dość równo rozłożonych przez miesiące. Niestety nie jest to opisane na stronie zbioru danych. Szukając w internecie znalazłem dane że dla małych pożarów powierzchnia została ustalona na 0.

[9]:
```
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(18, 9))
sns.countplot(data = fires_data, x = "month", order = month_order, ax=ax1,␣
 ↪color = "tab:blue")
sns.barplot(x="month", y="area_log", order=month_order, data=fires_data,␣
 ↪estimator=sum, ax=ax2, color = "tab:blue")
plt.suptitle("Count and sum of forest fires area per month", y=0.92)
ax1.set_xlabel("Month")
ax1.set_ylabel("Count of occurences")
ax2.set_xlabel("Month")
ax2.set_ylabel("Sum of area (log(x+1) scale)")
plt.show()
```

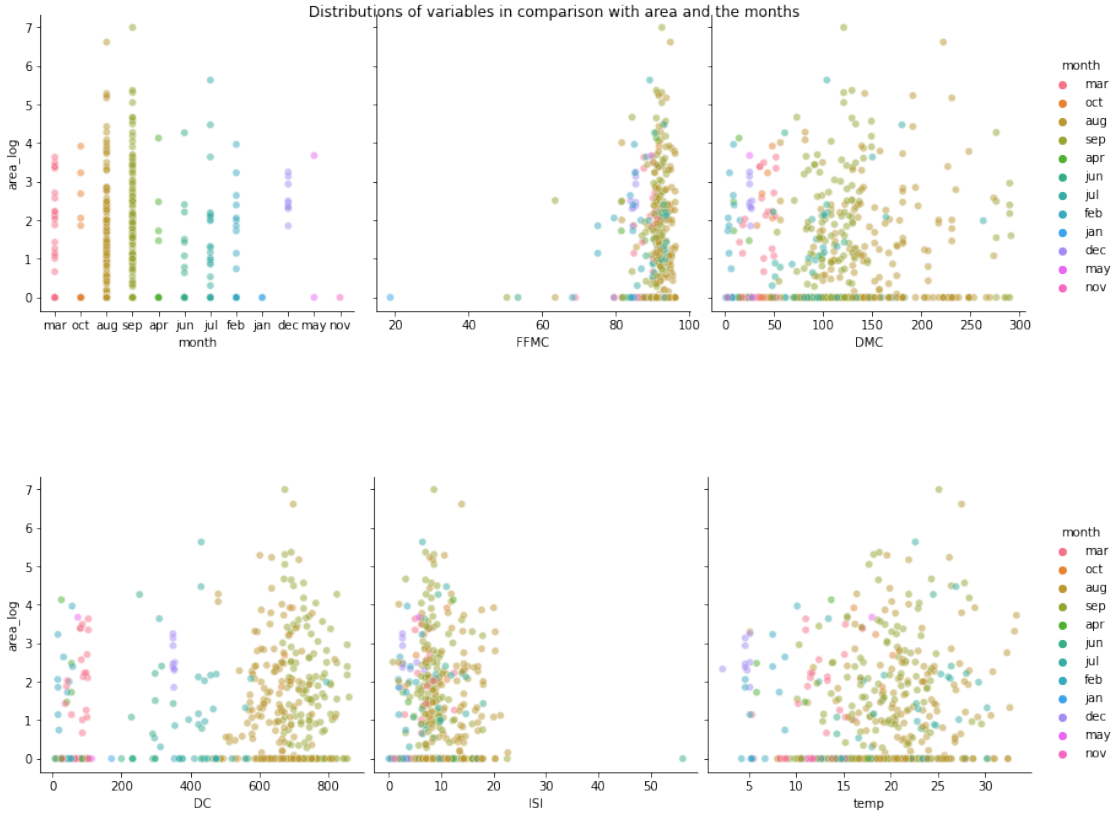


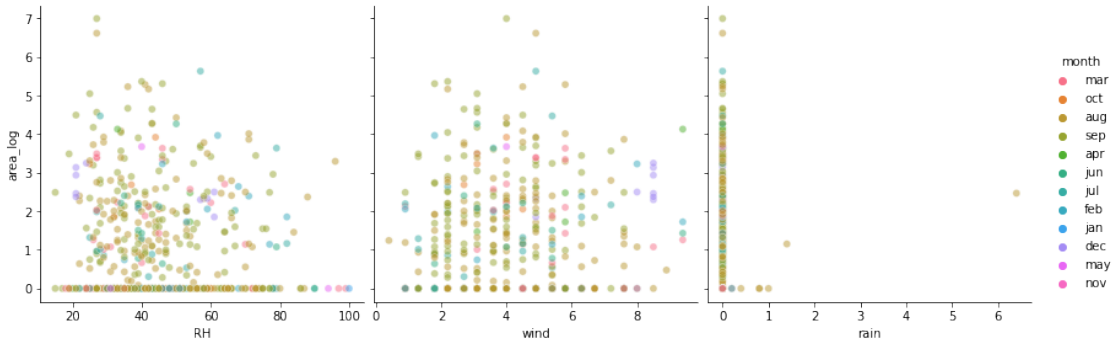
[10]:
```
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(18, 9))
plt.suptitle("Boxplots of area and temperature for the records", y=0.92)
sns.boxplot(x='month', y="area_log", data=fires_data, order=month_order,␣
 ↪ax=ax1, color = "tab:blue")
sns.boxplot(x='month', y="temp", data=fires_data, order=month_order, ax=ax2,␣
 ↪color = "tab:blue")
ax1.set_ylabel("Area in logarithmic scale")
ax2.set_ylabel("Temperature in Celsius")
plt.show()
```



Wygląda na to że istnieje powiązanie między temperaturą danego dnia a powierzchnią pożaru, jednak wydaje się być mniejsza niż bym się spodziewał.

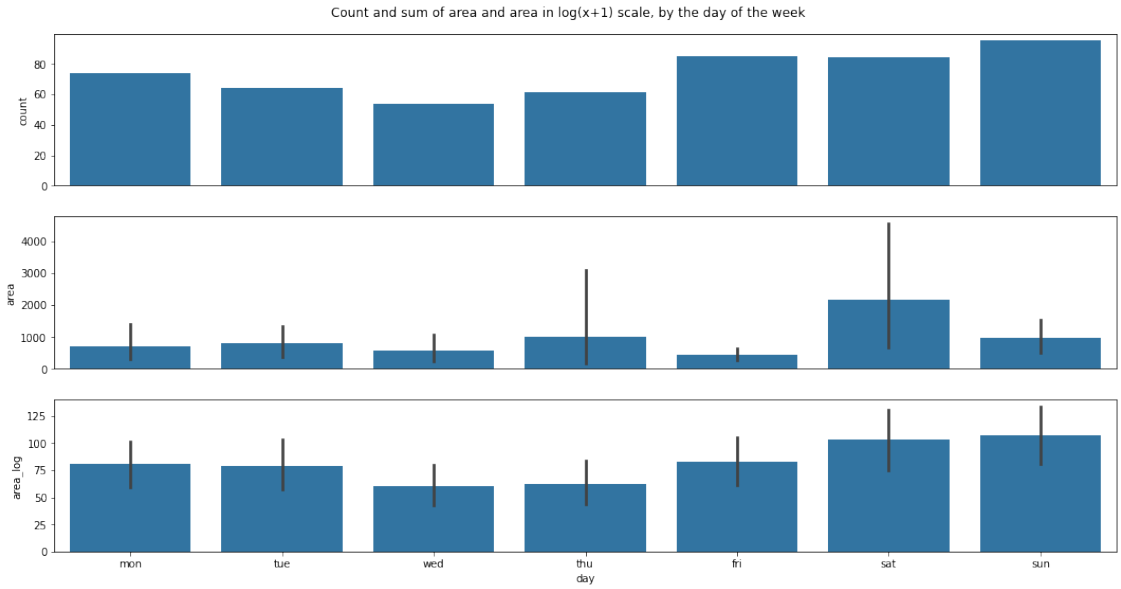
[11]:
```
hue_var = "month"
y_variable = "area_log"
temp_data = fires_data.drop(["X", "Y", "area", "day"], axis = 1)
fst = sns.pairplot(temp_data, y_vars=y_variable, x_vars=temp_data.columns.
 ↪values[:3], hue = hue_var, plot_kws={'alpha': 0.5}, height = 4)
scd = sns.pairplot(temp_data, y_vars=y_variable, x_vars=temp_data.columns.
 ↪values[3:6], hue = hue_var, plot_kws={'alpha': 0.5}, height = 4)
trd = sns.pairplot(temp_data, y_vars=y_variable, x_vars=temp_data.columns.
 ↪values[6:9], hue = hue_var, plot_kws={'alpha': 0.5}, height = 4)
fst.fig.suptitle("Distributions of variables in comparison with area and the␣
 ↪months", y=1.0)
plt.show()
```





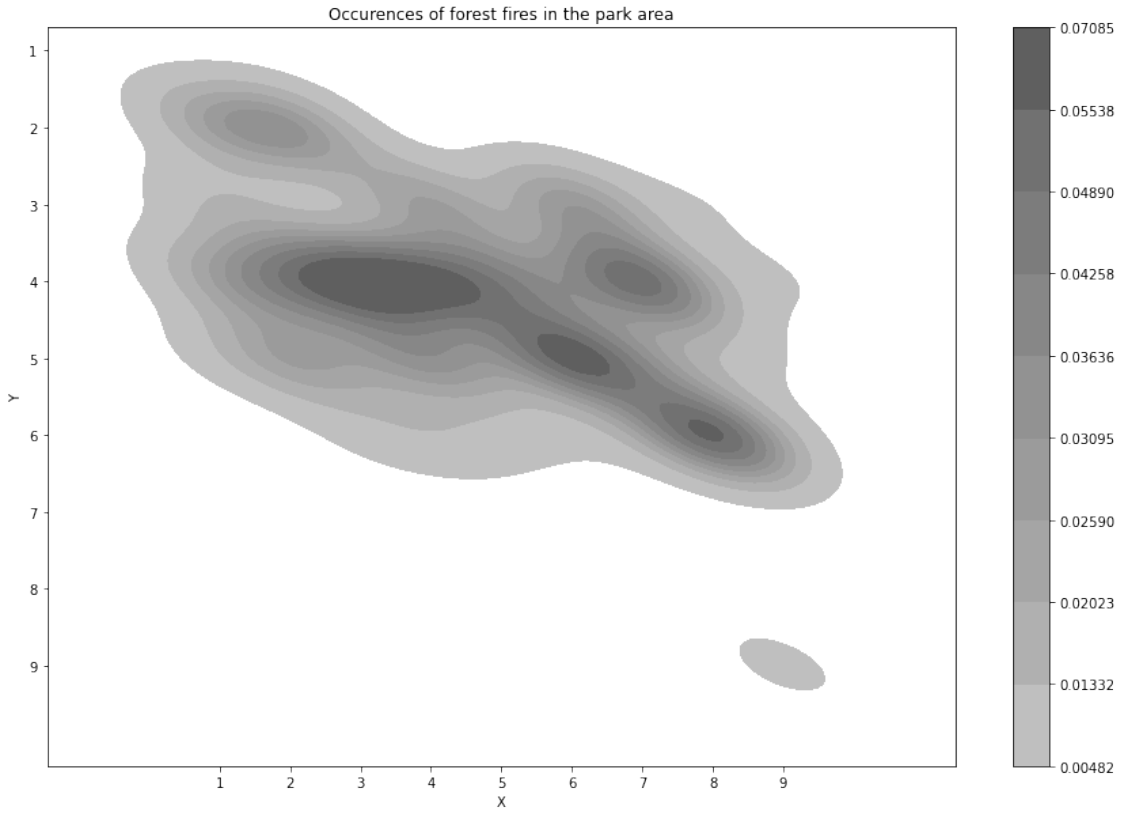
Jedyny klarowny rozdział miesięcy jest w wykresie area_log/DC. Według https://www.nwcg.gov/publications/pms437/cffdrs/fire-weather-index-system, DC to "The Drought Code" który wyznaczony jest na podstawie wysuszania się głębokich warstw gleby. Sensowne wydaje się powiązanie tego z miesiącami oraz z większymi pożarami.

[12]:
```
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(18, 9), sharex=True)
sns.countplot(data = fires_data, x = "day", order = day_order, ax=ax1, color =␣
 ↪"tab:blue")
sns.barplot(x="day", y="area", order=day_order, data=fires_data, estimator=sum,␣
 ↪ax=ax2, color = "tab:blue")
sns.barplot(x="day", y="area_log", order=day_order, data=fires_data,␣
 ↪estimator=sum, ax=ax3, color = "tab:blue")
ax1.tick_params(
    axis='x',
    which='both',
    bottom=False,
    labelbottom=False)
ax2.tick_params(
    axis='x',
    which='both',
    bottom=False,
    labelbottom=False)
ax1.set(xlabel='')
ax2.set(xlabel='')
plt.suptitle("Count and sum of area and area in log(x+1) scale, by the day of␣
 ↪the week", y = 0.92)
plt.show()
```

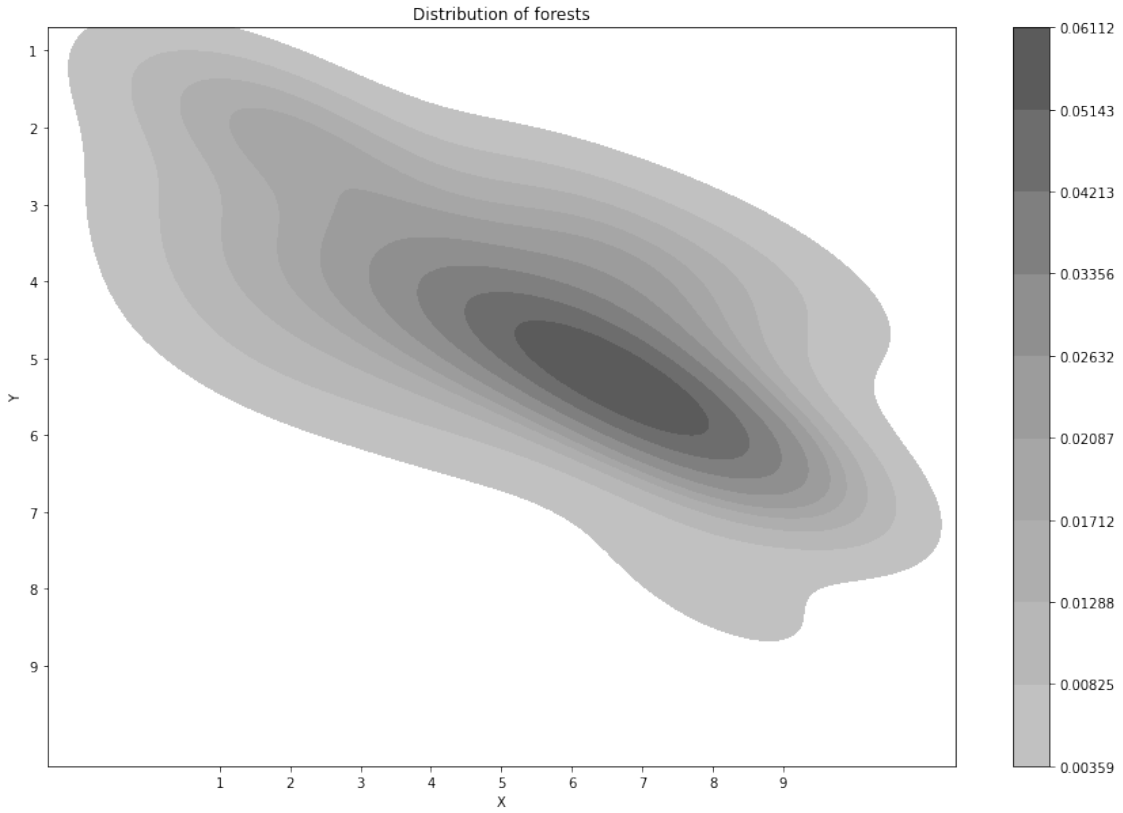


W weekend zdarza się najwięcej pożarów, tak samo pokrywają one największą powierzchnię. Zapewne związane jest to z uczęszczaniem społeczeństwa w wycieczkach do parku, które z oczywistych powodów wydarzają się w weekendy.

[13]:
```
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
sns.kdeplot(data = fires_data, x = "X", y = "Y", color="black", fill = True,␣
 ↪cbar = True, ax = ax)
ax.invert_yaxis()
plt.title("Occurences of forest fires in the park area")
ticks = [1,2,3,4,5,6,7,8,9]
plt.xticks(ticks)
plt.yticks(ticks)
plt.show()
```



[14]:
```
fig, ax = plt.subplots(1, 1, figsize=(15, 10))
sns.kdeplot(data = fires_data, x = "X", y = "Y", color="black", fill = True,␣
 ↪weights = "area", cbar = True, ax = ax)
ax.invert_yaxis()
plt.title("Distribution of forests")
ticks = [1,2,3,4,5,6,7,8,9]
plt.xticks(ticks)
plt.yticks(ticks)
plt.show()
```



Mając te dwa wykresy oraz mape parku możemy sprawdzić które obszary doświadczają najwięcej i największe pożary. Mogą to być obszary najczęściej uczęszczane przez ludzi, najdalsze od zbiorników wodnych, itd. Brakuje nam danych by to dokładniej określić.

0.4 Mapa Parku

Konwertowanie do pdf'a niestety usuwa mapę.

0.5 Pandas-Profiling

[15]:
```
profile = ProfileReport(fires_data, title='Pandas Profiling Report',␣
↪explorative=True)
```

[16]:
```
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>
```

Narzędzia automatycznej eksploracji danych mogą być użyteczne, jednak jako dodatek lub wstęp do porządnej analizy danych. Oczywiście wszystko w nich wykonuje się automatycznie, więc nie ma miejsca na skupianie się na odpowiednich zmiennych, wiedzy o danej dziedzinie która jest znacząca dla analizy, itd. Może przyspieszyć prostą lecz żmudną część robienia histogramów, .info/.describe.