

```
In [2]: import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set_style('darkgrid')
```

```
In [6]: #LOAD DATASET
```

```
df=pd.read_csv('forestfires.csv')
df
```

Out[6]:

	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	6.7	0.0	0.00
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00
...
512	4	3	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44
513	2	4	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29
514	7	4	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16
515	1	4	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00
516	6	3	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00

517 rows × 13 columns

In [7]: *#DISPLAY THE FIRST FEW ROWS*

```
df.head(20)
```

Out[7]:

	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	6.7	0.0	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.0
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.0
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.0
5	8	6	aug	sun	92.3	85.3	488.0	14.7	22.2	29	5.4	0.0	0.0
6	8	6	aug	mon	92.3	88.9	495.6	8.5	24.1	27	3.1	0.0	0.0
7	8	6	aug	mon	91.5	145.4	608.2	10.7	8.0	86	2.2	0.0	0.0
8	8	6	sep	tue	91.0	129.5	692.6	7.0	13.1	63	5.4	0.0	0.0
9	7	5	sep	sat	92.5	88.0	698.6	7.1	22.8	40	4.0	0.0	0.0
10	7	5	sep	sat	92.5	88.0	698.6	7.1	17.8	51	7.2	0.0	0.0
11	7	5	sep	sat	92.8	73.2	713.0	22.6	19.3	38	4.0	0.0	0.0
12	6	5	aug	fri	63.5	70.8	665.3	0.8	17.0	72	6.7	0.0	0.0
13	6	5	sep	mon	90.9	126.5	686.5	7.0	21.3	42	2.2	0.0	0.0
14	6	5	sep	wed	92.9	133.3	699.6	9.2	26.4	21	4.5	0.0	0.0
15	6	5	sep	fri	93.3	141.2	713.9	13.9	22.9	44	5.4	0.0	0.0
16	5	5	mar	sat	91.7	35.8	80.8	7.8	15.1	27	5.4	0.0	0.0
17	8	5	oct	mon	84.9	32.8	664.2	3.0	16.7	47	4.9	0.0	0.0
18	6	4	mar	wed	89.2	27.9	70.8	6.3	15.9	35	4.0	0.0	0.0
19	6	4	apr	sat	86.3	27.4	97.1	5.1	9.3	44	4.5	0.0	0.0

In [9]: feature_names = df.columns

```
feature_names
```

Out[9]: Index(['X', 'Y', 'month', 'day', 'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH',
 'wind', 'rain', 'area'],
 dtype='object')

In [12]: *#DATA PRIPRATION AND CLEANING..*

```
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    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
```

In [13]: `df.describe(include=['O'])`

Out[13]:

	month	day
count	517	517
unique	12	7
top	aug	sun
freq	184	95

In [14]: *# Lets create an variable and store its column name*

```
categorical_feature = df.describe(include=['O']).columns
```

```
# Print it in List type ...
```

```
print(list(categorical_feature))
```

```
['month', 'day']
```

```
In [15]: df['month'].value_counts()
```

```
Out[15]: aug      184
         sep      172
         mar       54
         jul       32
         feb       20
         jun       17
         oct       15
         apr        9
         dec        9
         jan        2
         may        2
         nov        1
         Name: month, dtype: int64
```

```
In [16]: df.describe(include=["int", "float"])
```

```
Out[16]:
```

	X	Y	FFMC	DMC	DC	ISI	temp	RH	
count	517.000000	517.000000	517.000000	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	18.889168	44.288201	4.669246
std	2.313778	1.229900	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	2.313778
min	1.000000	2.000000	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000
25%	3.000000	4.000000	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.000000
50%	4.000000	4.000000	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000
75%	7.000000	5.000000	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.000000
max	9.000000	9.000000	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000

```
In [17]: numerical_feature = df.describe(include=["int", "float"]).columns
```

```
# Print it in list type ..
print(list(numerical_feature))
```

```
['X', 'Y', 'FFMC', 'DMC', 'DC', 'ISI', 'temp', 'RH', 'wind', 'rain', 'area']
```

```
In [19]: df['area_km'] = df['area'] / 100
df
```

Out[19]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	area_km
0	7	5	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	0.00	0.0000
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.00	0.0000
2	7	4	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	0.00	0.0000
3	8	6	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	0.00	0.0000
4	8	6	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	0.00	0.0000
...
512	4	3	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	6.44	0.0644
513	2	4	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	54.29	0.5429
514	7	4	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	11.16	0.1116
515	1	4	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	0.00	0.0000
516	6	3	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	0.00	0.0000

517 rows × 14 columns

```
In [20]: df.sort_values(by="area_km", ascending=False).head()
```

Out[20]:

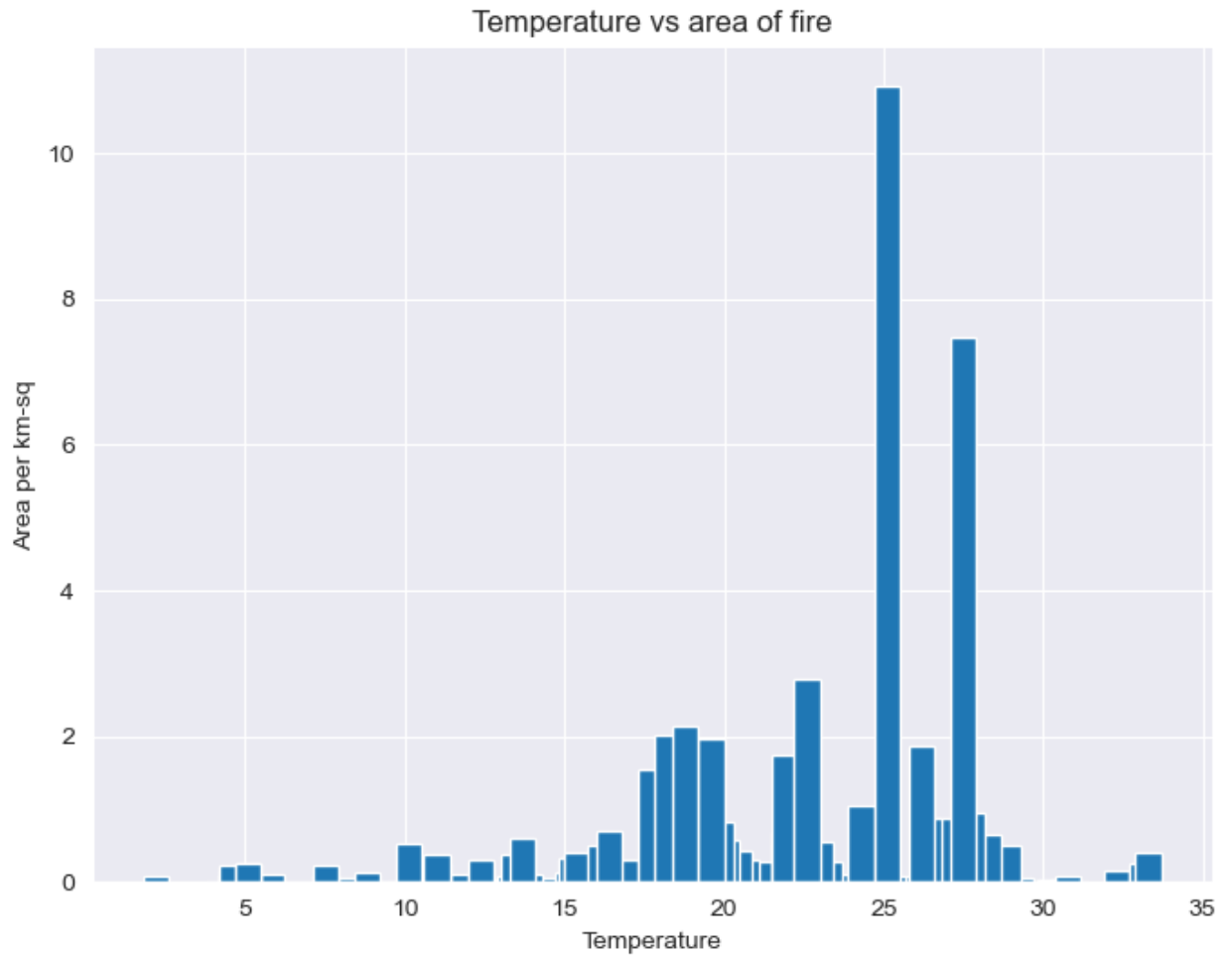
	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area	area_km
238	6	5	sep	sat	92.5	121.1	674.4	8.6	25.1	27	4.0	0.0	1090.84	10.9084
415	8	6	aug	thu	94.8	222.4	698.6	13.9	27.5	27	4.9	0.0	746.28	7.4628
479	7	4	jul	mon	89.2	103.9	431.6	6.4	22.6	57	4.9	0.0	278.53	2.7853
237	1	2	sep	tue	91.0	129.5	692.6	7.0	18.8	40	2.2	0.0	212.88	2.1288
236	2	2	sep	sat	92.5	121.1	674.4	8.6	18.2	46	1.8	0.0	200.94	2.0094

```
In [21]: highest_fire_area = df.sort_values(by="area_km", ascending=True)

plt.figure(figsize=(8, 6))

plt.title("Temperature vs area of fire" )
plt.bar(highest_fire_area['temp'], highest_fire_area['area_km'])

plt.xlabel("Temperature")
plt.ylabel("Area per km-sq")
plt.show()
```



```
In [23]: print(categorical_feature)
```

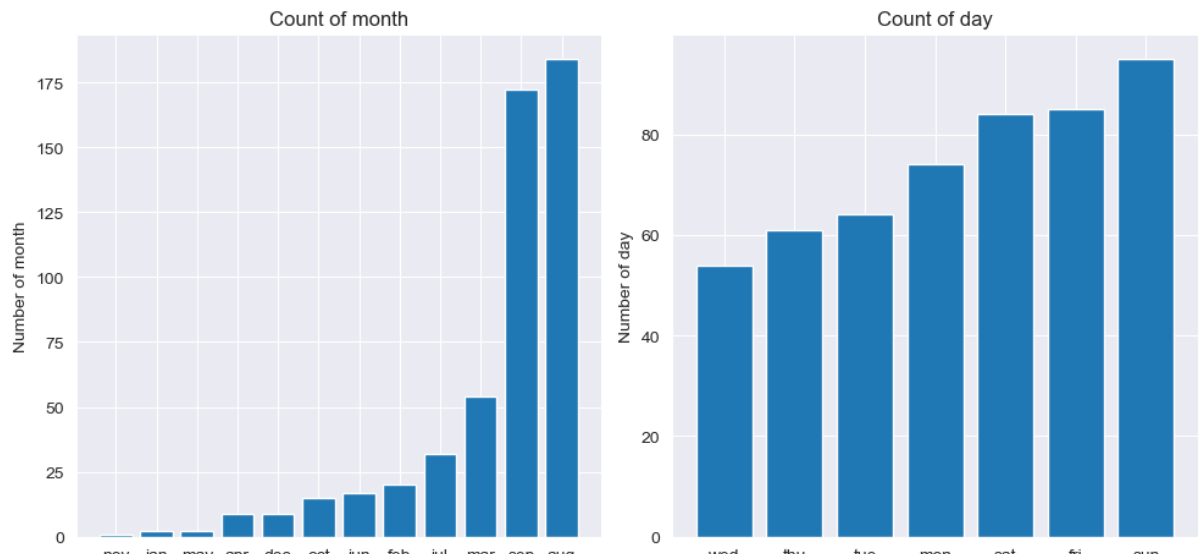
```
plt.figure(figsize=(10, 5))
for idx, column in enumerate(categorical_feature):
    df = df.copy()
    unique = df[column].value_counts(ascending=True);

    plt.subplot(1, 2, idx+1)
    plt.title("Count of " + column)
    plt.bar(unique.index, unique.values);

    plt.xlabel(column)
    plt.ylabel("Number of " + column)

plt.tight_layout()
plt.show()
```

```
Index(['month', 'day'], dtype='object')
```



```
In [25]: ###PairPloat Relation  
  
len(numerical_feature)  
  
# forest_df[numerical_feature]  
df[categorical_feature]
```

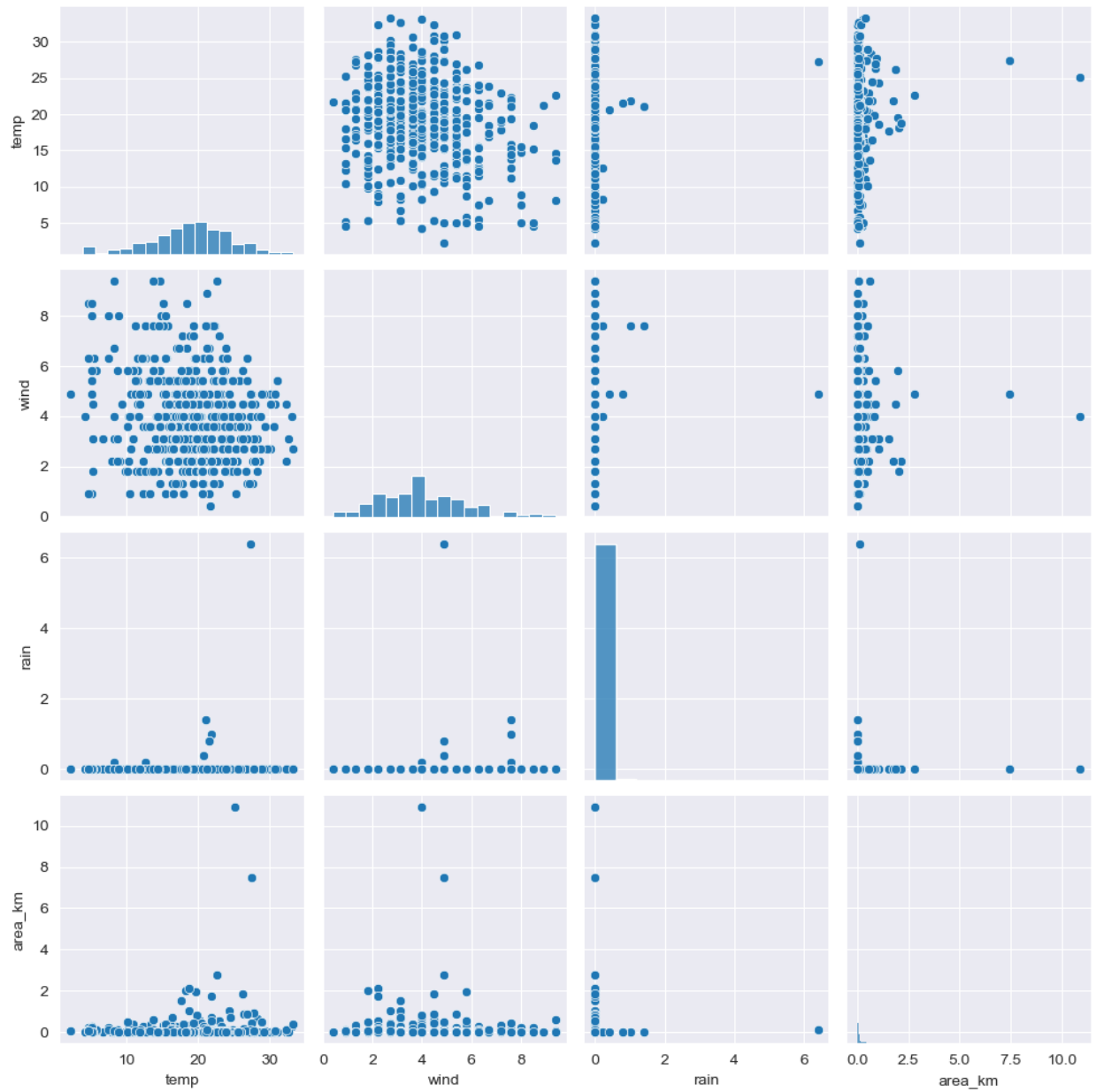
Out[25]:

	month	day
0	mar	fri
1	oct	tue
2	oct	sat
3	mar	fri
4	mar	sun
...
512	aug	sun
513	aug	sun
514	aug	sun
515	aug	sat
516	nov	tue

517 rows × 2 columns


```
In [27]: sns.set_style('darkgrid')
# Find the relation
# plt.subplot(forest_df)

sns.pairplot(df[["temp", "wind", "rain", "area_km"]])
plt.show()
```

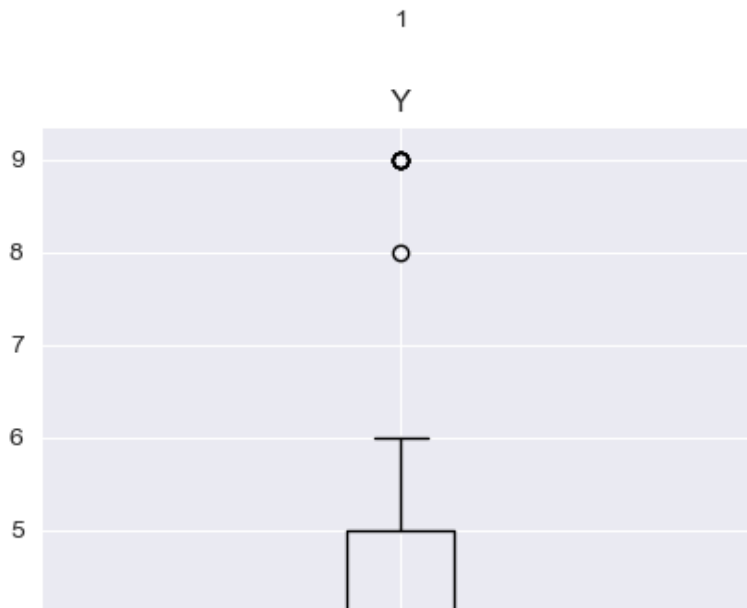


```
In [29]: for idx, col in enumerate(numerical_feature, 1):
plt.figure(figsize=(5,5))

# plt.subplot(len(numerical_feature) // 2, 3, idx)
plt.boxplot(df[col])

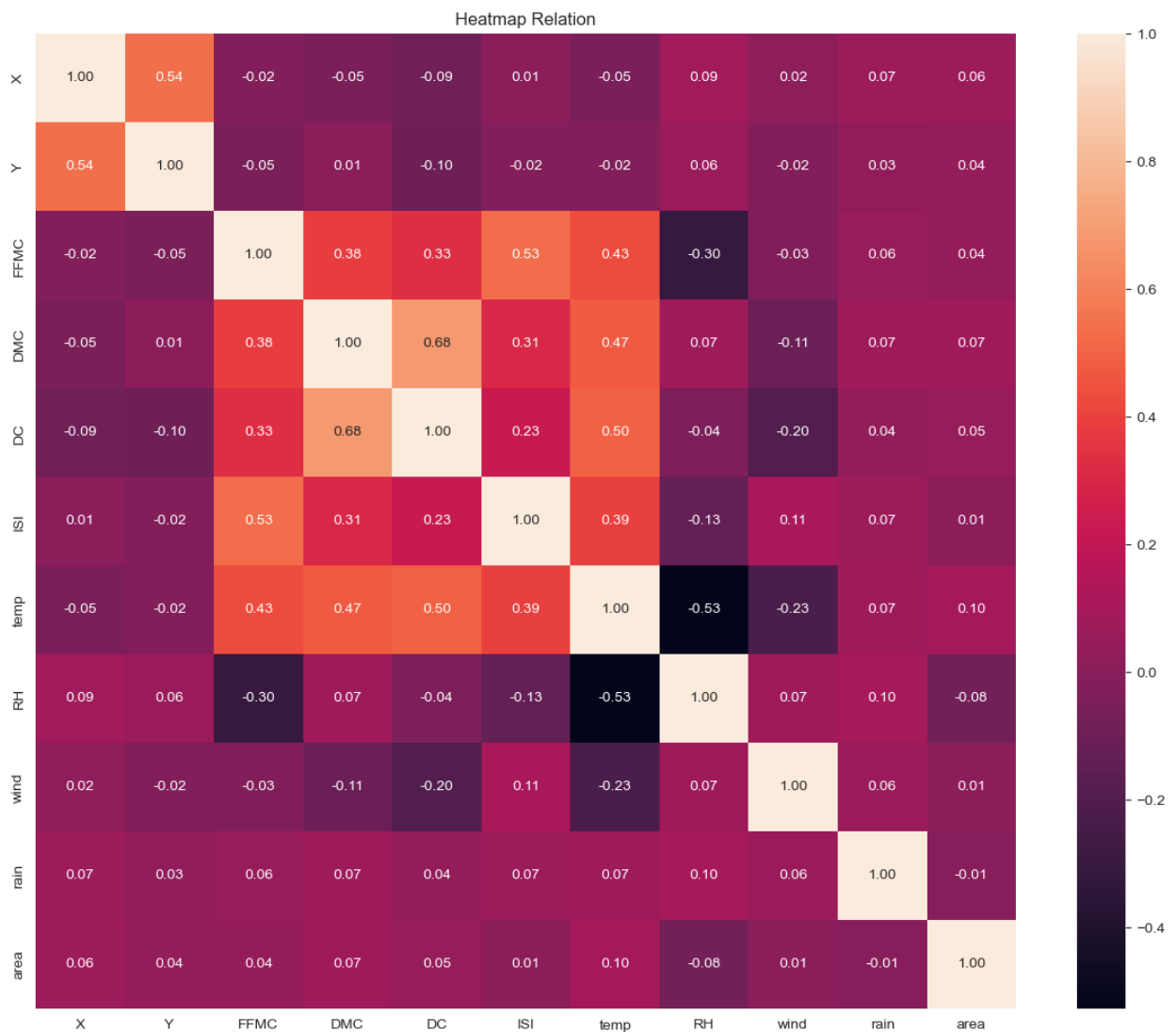
plt.title(col)
# plt.hist(forest_df[col])

plt.tight_layout()
plt.show(plt)
```



In [31]: `### HEAT MAP ###`

```
plt.figure(figsize=(15, 12))
plt.title("Heatmap Relation")
sns.heatmap(df[numerical_feature].corr(), annot=True, fmt='.2f');
```



In [35]: `### What is the highest rain and give the month and days. ###`

```
highest_rain = df.sort_values(by='rain', ascending=False)[['month', 'day', 'rain']].head(
highest_rain
```

Out[35]:

	month	day	rain
499	aug	tue	6.4
509	aug	fri	1.4
243	aug	sun	1.0
500	aug	tue	0.8
501	aug	tue	0.8

```
In [36]: ##### Give the highest and lowest temperature in month and day wise. #####

highest_temp = df.sort_values(by='temp', ascending=False)[['month', 'day', 'temp']].head
lowest_temp = df.sort_values(by='temp', ascending=True)[['month', 'day', 'temp']].head(

print("Highest Temperature")

print(highest_temp)

print()

print()

print("Lowest Temperature")
print(lowest_temp)
```

Highest Temperature

	month	day	temp
498	aug	tue	33.3
484	aug	sun	33.1
496	aug	mon	32.6
492	aug	fri	32.4
491	aug	thu	32.4

Lowest Temperature

	month	day	temp
280	dec	fri	2.2
282	feb	sun	4.2
279	dec	mon	4.6
278	dec	mon	4.6
277	dec	mon	4.6

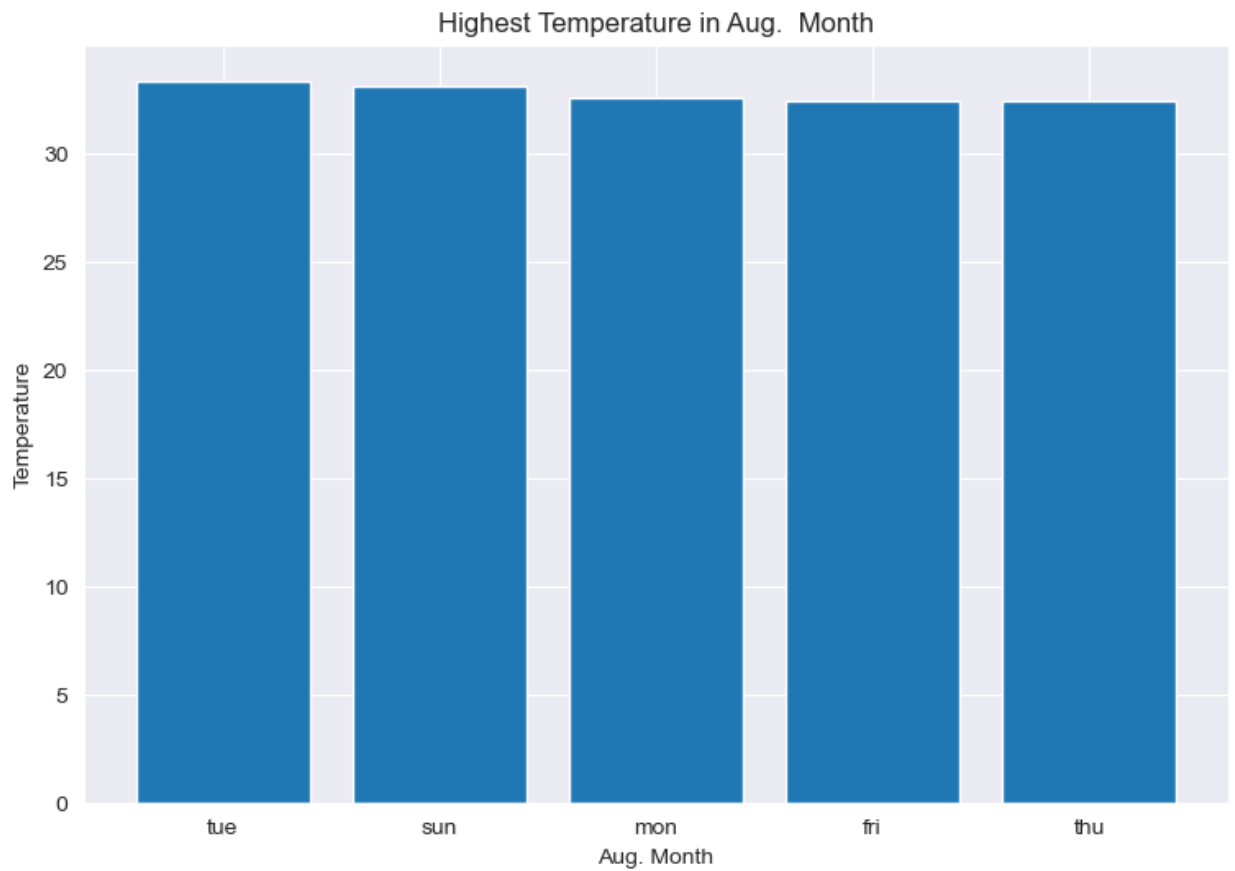
```
In [37]: plt.figure(figsize=(9, 6))

plt.title("Highest Temperature in Aug. Month")

plt.bar(highest_temp['day'], highest_temp['temp'])

plt.xlabel("Day")
plt.xlabel("Aug. Month")

plt.ylabel("Temperature")
plt.show()
```



```
In [38]: plt.figure(figsize=(9, 6))

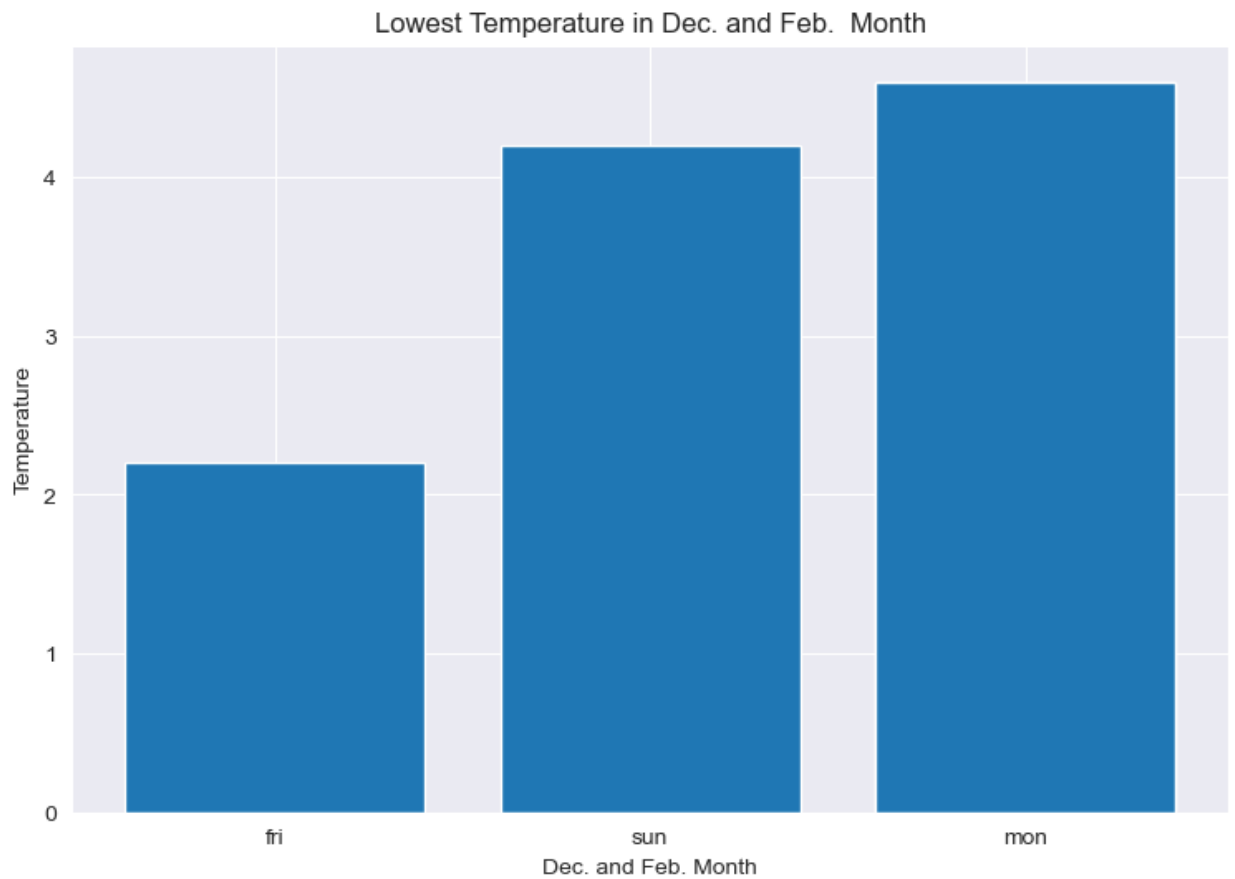
plt.title("Lowest Temperature in Dec. and Feb. Month")

plt.bar(lowest_temp['day'], lowest_temp['temp'])

plt.xlabel("Day")
plt.xlabel("Dec. and Feb. Month")

plt.ylabel("Temperature")

plt.show()
```



In []: