

```
In [2]: import numpy as np
import pandas as pd
import matplotlib as plt
import seaborn as sns
%matplotlib inline
# set seaborn dark grid
sns.set()
```

```
In [3]: # Create a variable for the data to be worked on.
forest_fire = 'C:/Users/USER/Desktop/The-Data-Visualization-Workshop-master/The-Data-Visualization-Workshop-master/forest_fire.csv'
```

```
In [4]: # Read the data using the pandas library
data_forest = pd.read_csv(forest_fire)
```

```
In [5]: # View the size of the dataset
data_forest.shape
```

Out[5]: (517, 13)

```
In [6]: # View the first two rows of the forest data
data_forest[0:2]
```

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	0.0
1	7	4	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	0.0

```
In [7]: # View the first ten rows of the forest data
data_forest[0:11]
```

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

```
In [8]: # Description of the forest data
data_forest.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 [9]: # Summary of the data
data_forest.describe()
```

Out[9]:

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
count	517.000000	517.000000	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	4.017602	0.021663	12.847292
std	2.313778	1.229900	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.295959	63.655818
min	1.000000	2.000000	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	0.000000	0.000000
25%	3.000000	4.000000	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	0.000000	0.000000
50%	4.000000	4.000000	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	0.000000	0.520000
75%	7.000000	5.000000	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	0.000000	6.570000
max	9.000000	9.000000	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000	6.400000	1090.840000

```
In [10]: # Checking if there are any null values in the forest data
data_forest.isnull().any()
```

Out[10]:

X	False
Y	False
month	False
day	False
FFMC	False
DMC	False
DC	False
ISI	False
temp	False
RH	False
wind	False
rain	False
area	False
dtype:	bool

```
In [11]: area_forest = data_forest[data_forest['area']>0]
area_forest
```

Out[11]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
138	9	9	jul	tue	85.8	48.3	313.4	3.9	18.0	42	2.7	0.0	0.36
139	1	4	sep	tue	91.0	129.5	692.6	7.0	21.7	38	2.2	0.0	0.43
140	2	5	sep	mon	90.9	126.5	686.5	7.0	21.9	39	1.8	0.0	0.47
141	1	2	aug	wed	95.5	99.9	513.3	13.2	23.3	31	4.5	0.0	0.55
142	8	6	aug	fri	90.1	108.0	529.8	12.5	21.2	51	8.9	0.0	0.61
...	...	...	...	...	...	...	...	...	...	...	...	...	...
509	5	4	aug	fri	91.0	166.9	752.6	7.1	21.1	71	7.6	1.4	2.17
510	6	5	aug	fri	91.0	166.9	752.6	7.1	18.2	62	5.4	0.0	0.43
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

270 rows × 13 columns

```
In [12]: area_forest['area'].mean()
```

Out[12]: 24.60018518518518

```
In [13]: # Max value of the sorted areas > 0
area_forest['area'].max()
```

Out[13]: 1090.84

```
In [14]: # Min value of the sorted areas > 0
area_forest['area'].min()
```

Out[14]: 0.09

```
In [15]: # Standard deviation of the sorted areas > 0
area_forest['area'].std()
```

Out[15]: 86.50163460412125

```
In [16]: # Median value of the sorted areas > 0
area_forest['area'].median()
```

Out[16]: 6.37

```
In [17]: # Sort the filtered dataset using the area column and print the last 20 entries using
# the tail method to see how many very large values

area_forest.sort_values(by=["area"]).tail(20)
```

Out[17]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
469	6	3	apr	sun	91.0	14.6	25.6	12.3	13.7	33	9.4	0.0	61.13
228	4	6	sep	sun	93.5	149.3	728.6	8.1	28.3	26	3.1	0.0	64.10
473	9	4	jun	sat	90.5	61.1	252.6	9.4	24.5	50	3.1	0.0	70.32
392	1	3	sep	sun	91.0	276.3	825.1	7.1	21.9	43	4.0	0.0	70.76
229	8	6	aug	sat	92.2	81.8	480.8	11.9	16.4	43	4.0	0.0	71.30
457	1	4	aug	wed	91.7	191.4	635.9	7.8	19.9	50	4.0	0.0	82.75
293	7	6	jul	tue	93.1	180.4	430.8	11.0	26.9	28	5.4	0.0	86.45
230	4	4	sep	wed	92.9	133.3	699.6	9.2	26.4	21	4.5	0.0	88.49
231	1	5	sep	sun	93.5	149.3	728.6	8.1	27.8	27	3.1	0.0	95.18
232	6	4	sep	tue	91.0	129.5	692.6	7.0	18.7	43	2.7	0.0	103.39
233	9	4	sep	tue	84.4	73.4	671.9	3.2	24.3	36	3.1	0.0	105.66
234	4	5	sep	sat	92.5	121.1	674.4	8.6	17.7	25	3.1	0.0	154.88
377	2	2	aug	sat	93.7	231.1	715.1	8.4	21.9	42	2.2	0.0	174.63
420	8	8	aug	wed	91.7	191.4	635.9	7.8	26.2	36	4.5	0.0	185.76
235	8	6	aug	sun	91.4	142.4	601.4	10.6	19.6	41	5.8	0.0	196.48
236	2	2	sep	sat	92.5	121.1	674.4	8.6	18.2	46	1.8	0.0	200.94
237	1	2	sep	tue	91.0	129.5	692.6	7.0	18.8	40	2.2	0.0	212.88
479	7	4	jul	mon	89.2	103.9	431.6	6.4	22.6	57	4.9	0.0	278.53
415	8	6	aug	thu	94.8	222.4	698.6	13.9	27.5	27	4.9	0.0	746.28
238	6	5	sep	sep	92.5	121.1	674.4	8.6	25.1	27	4.0	0.0	1090.84

```
In [20]: # List all the months in the datasets in order to compare the values of temp.
months = area_forest["month"].unique()
months
```

Out[20]: array(['jul', 'sep', 'aug', 'mar', 'jun', 'apr', 'oct', 'feb', 'dec', 'may'], dtype=object)

```
In [22]: area_forest[area_forest['month'] == 'mar'].shape[0]
```

Out[22]: 19

```
In [24]: # filter our dataset for rows containing the given month, and calculate the mean temperature. Print a statement
# number of fires, the mean temperature, and the month
for month in months:
    forest = area_forest[area_forest["month"] == month]
    fires_in_month = month_forest.shape[0]
    avg_tmp_in_month = int(month_forest["temp"].mean())
    print(str(fires_in_month) + " fires in " + month + \
          + " with a mean temperature of ~" + \
          + str(avg_tmp_in_month) + "°C")
```

```
18 fires in jul with a mean temperature of ~22°C
97 fires in sep with a mean temperature of ~19°C
99 fires in aug with a mean temperature of ~22°C
19 fires in mar with a mean temperature of ~12°C
8 fires in jun with a mean temperature of ~21°C
4 fires in apr with a mean temperature of ~9°C
5 fires in oct with a mean temperature of ~18°C
10 fires in feb with a mean temperature of ~7°C
9 fires in dec with a mean temperature of ~4°C
1 fires in may with a mean temperature of ~18°C
```

```
In [ ]:
```