

```
In [3]: import pandas as pd
import seaborn as sns
import numpy as np
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Dense, Activation, Layer, Lambda
```

```
In [4]: foefires=pd.read_csv('forestfires.csv')
foefires
```

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	month
0	mar	fri	86.2	26.2	94.3	15.1	8.2	51	6.7	0.0	...	...	0
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	...	0
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	...	0
3	mar	fri	91.7	33.3	77.5	15.0	8.3	97	4.0	0.2	...	...	0
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	...	...	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
512	aug	sun	81.6	56.7	665.6	1.9	27.8	32	2.7	0.0	...	...	0
513	aug	sun	81.6	56.7	665.6	1.9	21.9	71	5.8	0.0	...	...	0
514	aug	sun	81.6	56.7	665.6	1.9	21.2	70	6.7	0.0	...	...	0
515	aug	sat	94.4	146.0	614.7	11.3	25.6	42	4.0	0.0	...	...	0
516	nov	tue	79.5	3.0	106.7	1.1	11.8	31	4.5	0.0	...	...	0

517 rows x 31 columns

```
In [5]: foefires.drop(["month","day"],axis=1,inplace = True)
```

```
In [6]: foefires["size_category"].value_counts()
```

Out[6]: small 178  
large 379  
Name: size\_category, dtype: int64

```
In [7]: foefires.isnull().sum()
```

Out[7]: FFMC: 0  
DMC: 0  
DC: 0  
ISI: 0  
temp: 0  
RH: 0  
wind: 0  
rain: 0  
area: 0  
dayfri: 0  
daymon: 0  
daysat: 0  
daysun: 0  
daythu: 0  
daytue: 0  
daywed: 0  
monthaug: 0  
monthdec: 0  
monthfeb: 0  
monthjan: 0  
monthjul: 0  
monthjun: 0  
monthmar: 0  
monthmay: 0  
monthnov: 0  
monthoct: 0  
monthsep: 0  
size\_category: 0  
dtype: int64

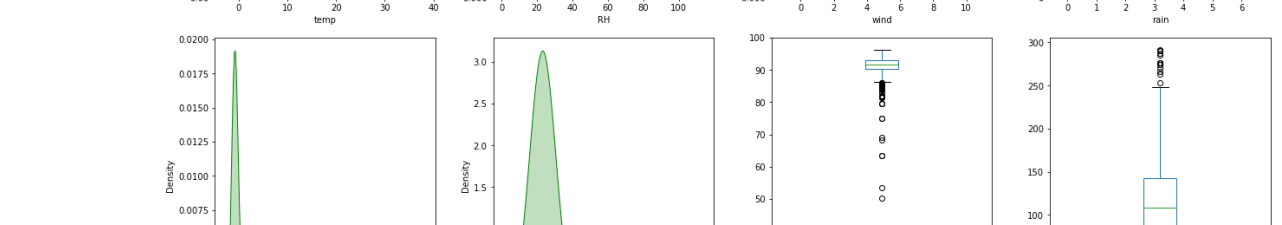
```
In [8]: foefires.describe().T
```

	count	mean	std	min	25%	50%	75%	max
FFMC	517.0	90.644681	5.520111	18.7	90.2	91.60	92.90	96.20
DMC	517.0	110.872340	64.046482	1.1	68.6	108.30	142.40	291.30
DC	517.0	547.940039	248.066192	7.9	437.7	664.20	713.90	860.60
ISI	517.0	9.021603	4.559477	0.0	6.5	8.40	10.80	56.10
temp	517.0	18.889168	5.806625	2.2	15.5	19.30	22.80	33.30
RH	517.0	44.288201	16.317469	15.0	33.0	42.00	53.00	100.00
wind	517.0	4.017602	1.791653	0.4	2.7	4.00	4.90	9.40
rain	517.0	0.021663	0.295959	0.0	0.0	0.00	0.00	6.40
area	517.0	12.847292	63.655818	0.0	0.0	0.52	6.57	1090.84
dayfri	517.0	0.164410	0.371006	0.0	0.0	0.00	0.00	1.00
daymon	517.0	0.143133	0.350548	0.0	0.0	0.00	0.00	1.00
daysat	517.0	0.162476	0.369244	0.0	0.0	0.00	0.00	1.00
daysun	517.0	0.183752	0.387657	0.0	0.0	0.00	0.00	1.00
daythu	517.0	0.117988	0.322907	0.0	0.0	0.00	0.00	1.00
daytue	517.0	0.123791	0.329662	0.0	0.0	0.00	0.00	1.00
daywed	517.0	0.104449	0.306138	0.0	0.0	0.00	0.00	1.00
monthaug	517.0	0.017408	0.130913	0.0	0.0	0.00	0.00	1.00
monthdec	517.0	0.355899	0.479249	0.0	0.0	0.00	1.00	1.00
monthfeb	517.0	0.017408	0.130913	0.0	0.0	0.00	0.00	1.00
monthjan	517.0	0.036685	0.193029	0.0	0.0	0.00	0.00	1.00
monthjun	517.0	0.003868	0.062137	0.0	0.0	0.00	0.00	1.00
monthjul	517.0	0.001896	0.041199	0.0	0.0	0.00	0.00	1.00
monthmar	517.0	0.032892	0.178500	0.0	0.0	0.00	0.00	1.00
monthmay	517.0	0.104449	0.306138	0.0	0.0	0.00	0.00	1.00
monthnov	517.0	0.003868	0.062137	0.0	0.0	0.00	0.00	1.00
monthoct	517.0	0.001934	0.043980	0.0	0.0	0.00	0.00	1.00
monthsep	517.0	0.029014	0.168007	0.0	0.0	0.00	0.00	1.00
monthapr	517.0	0.332689	0.471632	0.0	0.0	0.00	1.00	1.00

## Outlier Check

```
In [9]: import warnings
warnings.filterwarnings('ignore')
```

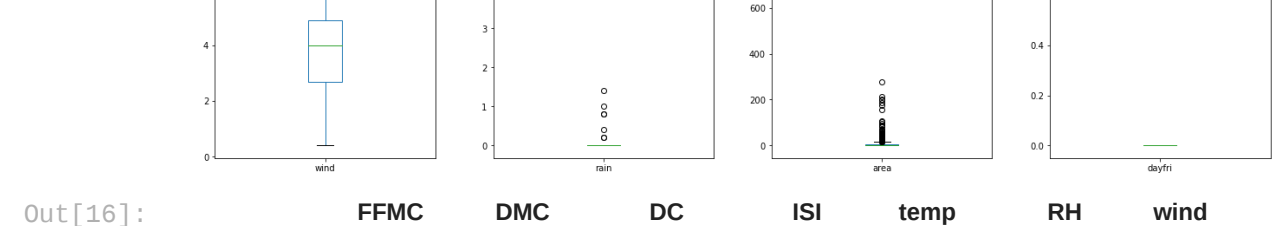
```
In [10]: ax = sns.boxplot(foefires['area'])
```



```
In [11]: plt.rcParams["figure.figsize"] = 9,5
```

```
In [12]: plt.figure(figsize=(16,5))
sns.skewplot(data=foefires['area'],skew())
print('kurtosis: {}'.format(foefires['area'].kurtosis()))
ax = sns.kdeplot(foefires['area'],shade=True,color='g')
plt.xticks([i for i in range(0,1200,50)])
plt.show()
```

Skew: 12.844983253934868  
Kurtosis: 194.1407210942299



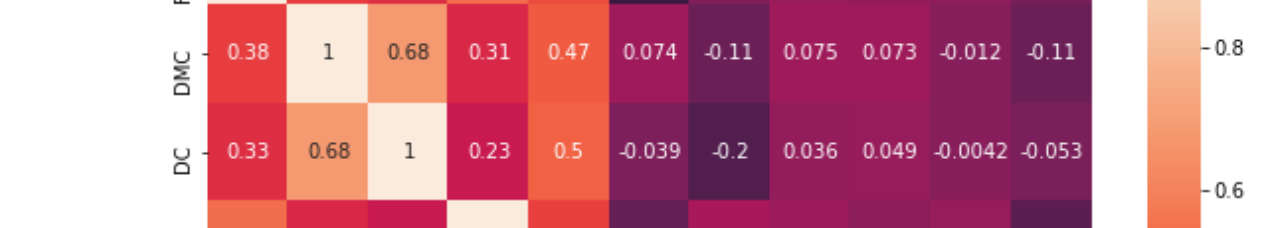
```
In [13]: dfa = foefires[foefires.columns[0:10]]
month_columns = dfa.select_dtypes(include='object').columns.tolist()
```

```
In [14]: plt.figure(figsize=(16,10))
for i,col in enumerate(month_columns):
    plt.subplot(2,2,i)
    sns.countplot(data=dfa,y=col)
    sns.countplot(data=dfa,y=col)
    plt.xticks([i for i in range(0,1200,50)])
    plt.tight_layout()
    plt.show()
```

<Figure size 1152x720 with 0 Axes>

```
In [15]: num_columns = dfa.select_dtypes(exclude='object').columns.tolist()
```

```
In [16]: plt.figure(figsize=(18,40))
for i,col in enumerate(num_columns):
    plt.subplot(8,4,i)
    plt.subplot(8,4,i+10)
    foefires[col].plot.box()
    plt.tight_layout()
    num_data = foefires[num_columns]
    pd.DataFrame(data=[num_data.skew(),num_data.kurtosis()],index=['skewne
```

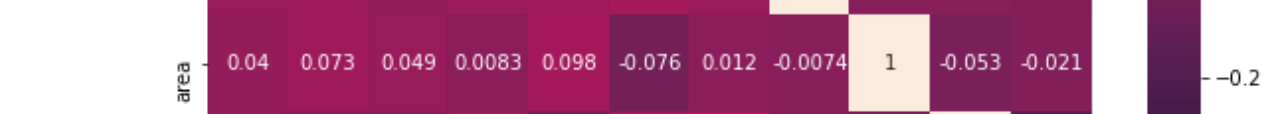


Out[16]: skewness -6.575606 0.547498 -1.100445 2.536325 -0.331172 0.862904 0.571001 19  
kurtosis 67.066041 0.204822 -0.245244 21.458037 0.136166 0.438183 0.054324 421

## Finding Correlation

```
In [17]: corr = foefires[foefires.columns[0:11]].corr()
```

```
In [18]: plt.figure(figsize=(10,10))
sns.heatmap(corr,annot=True)
```



Out[18]: <AxesSubplot>

FFMC 1 0.38 0.33 0.53 0.43 -0.3 -0.028 0.057 0.04 0.019 -0.059  
DMC 0.38 1 0.68 0.31 0.47 0.074 -0.11 0.075 0.073 -0.012 -0.11  
DC 0.33 0.68 1 0.23 0.5 -0.039 -0.2 0.036 0.049 -0.042 -0.053  
ISI -0.53 0.31 0.23 1 0.39 -0.13 0.11 0.068 0.0083 0.047 -0.16  
temp 0.43 0.47 0.5 0.39 1 -0.53 -0.23 0.069 0.098 -0.072 -0.14  
RH -0.3 0.074 -0.039 -0.13 -0.53 1 0.069 0.1 -0.076 0.065 0.0094  
wind -0.028 -0.11 -0.2 0.11 -0.23 0.069 1 0.061 0.012 0.12 -0.064  
rain 0.057 0.075 0.036 0.068 0.069 0.1 0.061 1 -0.0074 -0.0043 -0.03  
area 0.04 -0.073 0.049 0.0083 0.098 -0.076 0.012 -0.0074 1 -0.053 -0.021  
dayfri -0.019 -0.012 -0.042 0.047 -0.072 0.065 0.12 -0.0043 -0.053 1 -0.18  
daymon -0.055 -0.11 -0.053 -0.16 -0.14 0.0094 -0.064 -0.03 -0.021 -0.18 1  
daysat 0.057 0.075 0.036 0.068 0.069 0.1 0.061 1 -0.0074 -0.0043 -0.03  
daysun 0.04 -0.073 0.049 0.0083 0.098 -0.076 0.012 -0.0074 1 -0.053 -0.021  
daythu 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
daytue 0.055 0.11 0.053 0.16 0.14 0.0094 0.064 0.03 0.021 0.18 1  
daywed 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthaug 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthdec 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthfeb 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthjan 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthjul 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthjun 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthmar 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthmay 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthnov 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthoct 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthsep 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18  
monthapr 0.019 0.012 0.042 0.047 0.072 0.065 0.12 0.0043 0.053 1 -0.18

## Neural Network model

```
In [19]: mapping = {'small': 1, 'large': 2}
```

```
In [20]: foefires = foefires.replace(mapping)
```

```
In [21]: x = np.array(foefires.iloc[:,0:28])
y = np.array(foefires.iloc[:,28])
```

```
In [22]: def norm_func(i):
x = (i-i.min())/(i.max()-i.min())
return (x)
```

```
In [23]: X_norm = norm_func(X)
```

```
In [24]: x_train,x_test,y_train,y_test= train_test_split(X_norm,y, test_size=0.
```

```
In [25]: model = Sequential()
model.add(Dense(12, input_dim=28, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

```
In [26]: model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['
```

```
In [29]: model.fit(x_train, y_train, epochs=150, validation_data=None)
```

Epoch 1/150  
13/13 [=====] - 0s 2ms/step - loss: 0.6732 - accuracy: 0.6852  
Epoch 2/150  
13/13 [=====] - 0s 2ms/step - loss: 0.6323 - accuracy: 0.7312  
Epoch 3/150  
13/13 [=====] - 0s 2ms/step - loss: 0.5846 - accuracy: 0.7312  
Epoch 4/150  
13/13 [=====] - 0s 2ms/step - loss: 0.5280 - accuracy: 0.7312  
Epoch 5/150  
13/13 [=====] - 0s 1ms/step - loss: 0.4634 - accuracy: 0.7312  
Epoch 6/150  
13/13 [=====] - 0s 2ms/step - loss: 0.3898 - accuracy: 0.7312  
Epoch 7/150  
13/13 [=====] - 0s 2ms/step - loss: 0.3877 - accuracy: 0.7312  
Epoch 8/150  
13/13 [=====] - 0s 1ms/step - loss: 0.2172 - accuracy: 0.7312  
Epoch 9/150  
13/13 [=====] - 0s 2ms/step - loss: 0.1180 - accuracy: 0.7312  
Epoch 10/150  
13/13 [=====] - 0s 2ms/step - loss: 0.0180 - accuracy: 0.7312  
Epoch 11/150  
13/13 [=====] - 0s 2ms/step - loss: -0.1011 - accuracy: 0.7312  
Epoch 12/150  
13/13 [=====] - 0s 2ms/step - loss: -0.2168 - accuracy: 0.7312  
Epoch 13/150  
13/13 [=====] - 0s 2ms/step - loss: -0.3324 - accuracy: 0.7312  
Epoch 14/150  
13/13 [=====] - 0s 2ms/step - loss: -0.4594 - accuracy: 0.7312  
Epoch 15/150  
13/13 [=====] - 0s 2ms/step - loss: -0.5807 - accuracy: 0.7312  
Epoch 16/150  
13/13 [=====] - 0s 2ms/step - loss: -0.7096 - accuracy: 0.7312  
Epoch 17/150  
13/13 [=====] - 0s 2ms/step - loss: -0.8388 - accuracy: 0.7312  
Epoch 18/150  
13/13 [=====] - 0s 3ms/step - loss: -0.9744 - accuracy: 0.7312  
Epoch 19/150  
13/13 [=====] - 0s 2ms/step - loss: -1.1233 - accuracy: 0.7312  
Epoch 20/150  
13/13 [=====] - 0s 2ms/step - loss: -1.2751 - accuracy: 0.7312  
Epoch 21/150  
13/13 [=====] - 0s 3ms/step - loss: -1.4452 - accuracy: 0.7312  
Epoch 22/150  
13/13 [=====] - 0s 2ms/step - loss: -1.6452 - accuracy: 0.7312  
Epoch 23/150  
13/13 [=====] - 0s 2ms/step - loss: -1.8726 - accuracy: 0.7312  
Epoch 24/150  
13/13 [=====] - 0s 3ms/step - loss: -2.1268 - accuracy: 0.7312  
Epoch 25/150  
13/13 [=====] - 0s 2ms/step - loss: -2.4140 - accuracy: 0.7312  
Epoch 26/150  
13/13 [=====] - 0s 2ms/step - loss: -2.7312 - accuracy: 0.7312  
Epoch 27/150  
13/13 [=====] - 0s 2ms/step - loss: -3.0988 - accuracy: 0.7312  
Epoch 28/150  
13/13 [=====] - 0s 3ms/step - loss: -3.4819 - accuracy: 0.7312  
Epoch 29/150  
13/13 [=====] - 0s 3ms/step - loss: -3.9135 - accuracy: 0.7312  
Epoch 30/150  
13/13 [=====] - 0s 2ms/step - loss: -4.4100 - accuracy: 0.7312  
Epoch 31/150  
13/13 [=====] - 0s 2ms/step - loss: -4.9349 - accuracy: 0.7312  
Epoch 32/150  
13/13 [=====] - 0s 2ms/step - loss: -5.5090 - accuracy: 0.7312  
Epoch 33/150  
13/13 [=====] - 0s 2ms/step - loss: -6.1223 - accuracy: 0.7312  
Epoch 34/150  
13/13 [=====] - 0s 2ms/step - loss: -6.8027 - accuracy: 0.7312  
Epoch 35/150  
13/13 [=====] - 0s 2ms/step - loss: -7.5210 - accuracy: 0.7312  
Epoch 36/150  
13/13 [=====] - 0s 2ms/step - loss: -8.3033 - accuracy: 0.7312  
Epoch 37/150  
13/13 [=====] - 0s 2ms/step - loss: -9.1439 - accuracy: 0.7312  
Epoch 38/150  
13/13 [=====] - 0s 2ms/step - loss: -10.0267 - accuracy: 0.7312  
Epoch 39/150  
13/13 [=====] - 0s 2ms/step - loss: -10.9942 - accuracy: 0.7312  
Epoch 40/150  
13/13 [=====] - 0s 3ms/step - loss: -11.9952 - accuracy: 0.7312  
Epoch 41/150  
13/13 [=====] - 0s 2ms/step - loss: -13.0745 - accuracy: 0.7312  
Epoch 42/150  
13/13 [=====] - 0s 3ms/step - loss: -14.2161 - accuracy: 0.7312  
Epoch 43/150  
13/13 [=====] - 0s 3ms/step - loss: -15.4152 - accuracy: 0.7312  
Epoch 44/150  
13/13 [=====] - 0s 2ms/step - loss: -16.6663 - accuracy: 0.7312  
Epoch 45/150  
13/13 [=====] - 0s 2ms/step - loss: -17.9745 - accuracy: 0.7312  
Epoch 46/150  
13/13 [=====] - 0s 2ms/step - loss: -19.4228 - accuracy: 0.7312  
Epoch 47/150  
13/13 [=====] - 0s 2ms/step - loss: -20.8379 - accuracy: 0.7312  
Epoch 48/150  
13/13 [=====] - 0s 3ms/step - loss: -22.4229 - accuracy: 0.7312  
Epoch 49/150  
13/13 [=====] - 0s 2ms/step - loss: -24.0208 - accuracy: 0.7312  
Epoch 50/150  
13/13 [=====] - 0s 2ms/step - loss: -25.7092 - accuracy: 0.7312  
Epoch 51/150  
13/13 [=====] - 0s 2ms/step - loss: -27.4197 - accuracy: 0.7312  
Epoch 52/150  
13/13 [=====] - 0s 2ms/step - loss: -29.2493 - accuracy: 0.7312  
Epoch 53/150  
13/13 [=====] - 0s 1ms/step - loss: -31.1683 - accuracy: 0.7312  
Epoch 54/150  
13/13 [=====] - 0s 3ms/step - loss: -33.1014 - accuracy: 0.7312  
Epoch 55/150  
13/13 [=====] - 0s 3ms/step - loss: -35.1778 - accuracy: 0.7312  
Epoch 56/150  
13/13 [=====] - 0s 3ms/step - loss: -37.3148 - accuracy: 0.7312  
Epoch 57/150  
13/13 [=====] - 0s 2ms/step - loss: -39.4930 - accuracy: 0.7312  
Epoch 58/150  
13/13 [=====] - 0s 2ms/step - loss: -41.7847 - accuracy: 0.7312  
Epoch 59/150  
13/13 [=====] - 0s 2ms/step - loss: -44.1379 - accuracy: 0.7312  
Epoch 60/150  
13/13 [=====] - 0s 2ms/step - loss: -46.5779 - accuracy: 0.7312  
Epoch 61/150  
13/13 [=====] - 0s 2ms/step - loss: -49.1249 - accuracy: 0.7312  
Epoch 62/150  
13/13 [=====] - 0s 2ms/step - loss: -51.6768 - accuracy: 0.7312  
Epoch 63/150  
13/13 [=====] - 0s 3ms/step - loss: -54.4158 - accuracy: 0.7312  
Epoch 64/150  
13/13 [=====] - 0s 2ms/step - loss: -57.1217 - accuracy: 0.7312  
Epoch 65/150  
13/13 [=====] - 0s 2ms/step - loss: -60.1031 - accuracy: 0.7312  
Epoch 66/150  
13/13 [=====] - 0s 2ms/step - loss: -62.9902 - accuracy: 0.7312  
Epoch 67/150  
13/13 [=====] - 0s 2ms/step - loss: -66.0352 - accuracy: 0.7312  
Epoch 68/150  
13/13 [=====] - 0s 2ms/step - loss: -69.1567 - accuracy: 0.7312  
Epoch 69/150  
13/13 [=====] - 0s 2ms/step - loss: -72.3776 - accuracy: 0.7312  
Epoch 70/150  
13/13 [=====] - 0s 2ms/step - loss: -75.6341 - accuracy: 0.7312  
Epoch 71/150  
13/13 [=====] - 0s 2ms/step - loss: -79.0171 - accuracy: 0.7312  
Epoch 72/150  
13/13 [=====] - 0s 2ms/step - loss: -82.5053 - accuracy: 0.7312  
Epoch 73/150  
13/13 [=====] - 0s 3ms/step - loss: -86.1241 - accuracy: 0.7312  
Epoch 74/150  
13/13 [=====] - 0s 2ms/step - loss: -89.7829 - accuracy: 0.7312  
Epoch 75/150  
13/13 [=====] - 0s 2ms/step - loss: -93.5639 - accuracy: 0.7312  
Epoch 76/150  
13/13 [=====] - 0s 2ms/step - loss: -97.3429 - accuracy: 0.7312  
Epoch