In [1]:
```python
# Support vector machines generally deals with the separation between two layers
# there is a linear model,non linear model.
```

FOREST FIRE

In [4]:
```python
import numpy as np
import pandas as pd
from sklearn import preprocessing
from sklearn import metrics
import seaborn as sns
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
from sklearn.decomposition import PCA
from mlxtend.plotting import plot_decision_regions
```

In [5]:
```python
#classify the Size_Categorie using SVM
```

In [6]:
```python
# Let us import the dataset
forest_fire=pd.read_csv("C:\\Users\\nishi\\Desktop\\Assignments\\Support_Vector_M
```

In [7]:
```python
forest_fire
```

Out[7]:

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

517 rows × 31 columns

In [8]:
```python
forest_fire1=forest_fire.copy()
```

In [9]:
```
forest_fire1.head()
```

Out[9]:

	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	...	monthfeb	monthjan	monthjul
0	mar	fri	86.2	26.2	94.3	5.1	8.2	51	6.7	0.0	...	0	0	0
1	oct	tue	90.6	35.4	669.1	6.7	18.0	33	0.9	0.0	...	0	0	0
2	oct	sat	90.6	43.7	686.9	6.7	14.6	33	1.3	0.0	...	0	0	0
3	mar	fri	91.7	33.3	77.5	9.0	8.3	97	4.0	0.2	...	0	0	0
4	mar	sun	89.3	51.3	102.2	9.6	11.4	99	1.8	0.0	...	0	0	0

5 rows × 31 columns

In [10]:
```
forest_fire1.iloc[:,0:11]
```

Out[10]:

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

517 rows × 11 columns

In [11]:
```
forest_fire1.shape
```

Out[11]: (517, 31)

In [12]:
```
forest_fire1.describe()
```

Out[12]:

	FFMC	DMC	DC	ISI	temp	RH	wind	
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.00
mean	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	0.02
std	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	0.29
min	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	0.00
25%	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	0.00
50%	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	0.00
75%	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	0.00
max	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000	6.40

8 rows × 28 columns

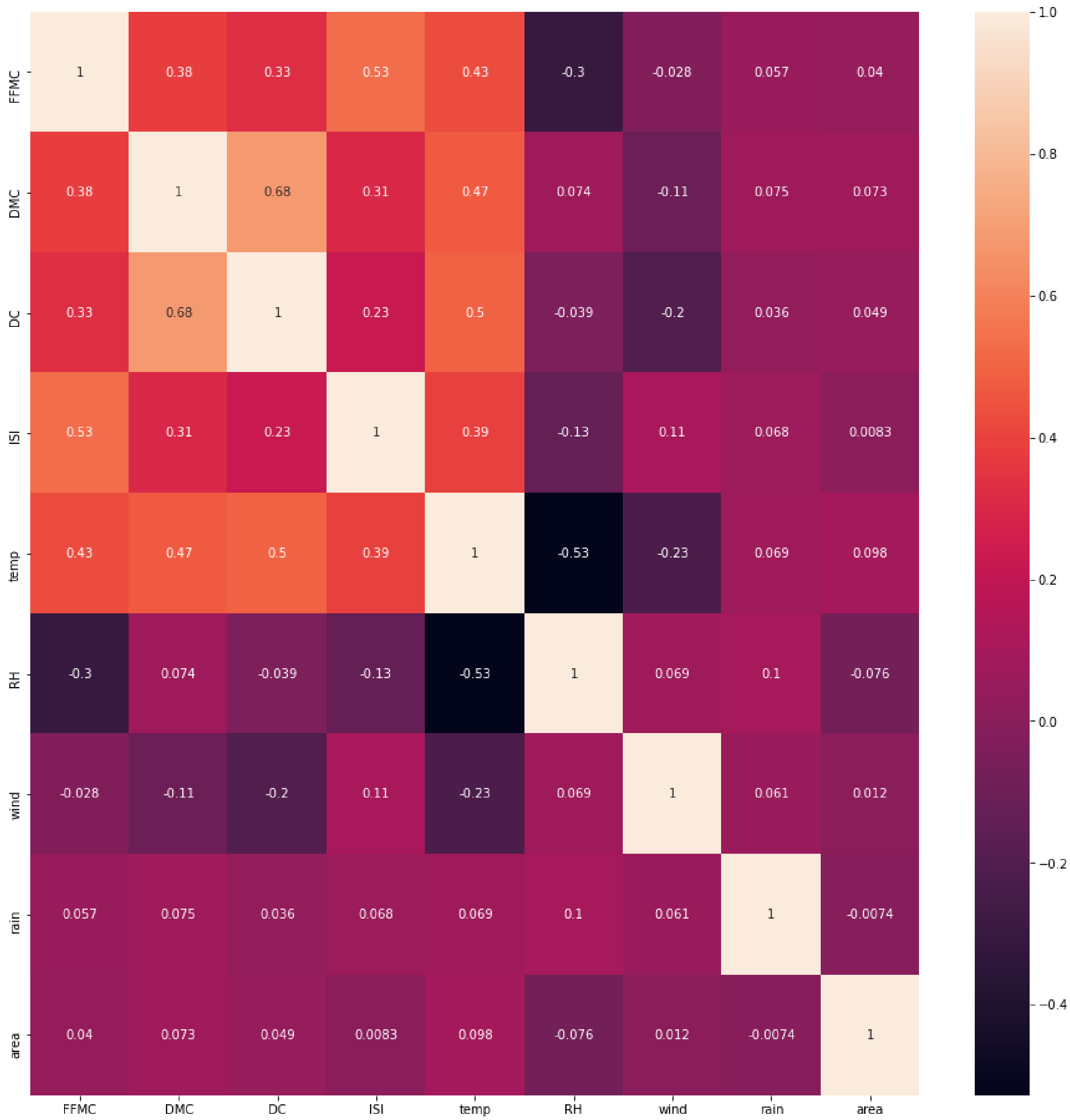
In [13]:
```
forest_fire1.isnull().sum()
```

Out[13]:
```
month            0
day              0
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
monthapr         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 [14]:
```
# Correlation
corr=forest_fire1.iloc[:,0:11].corr()
```

In [15]:
```
plt.figure(figsize=(16,16))
sns.heatmap(corr,annot=True)
```

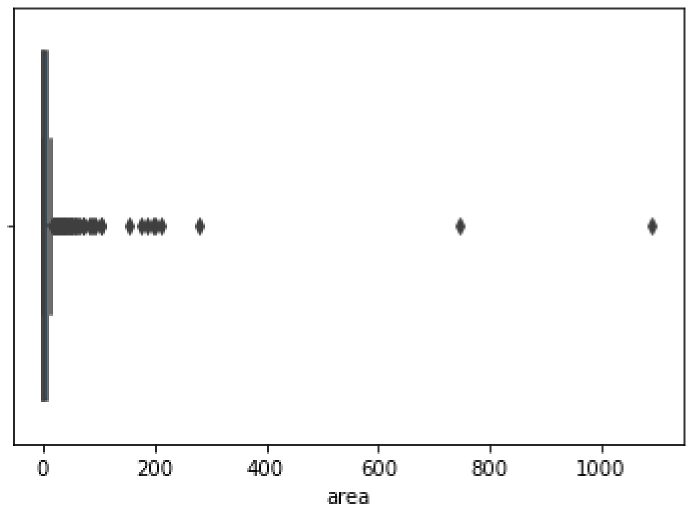
Out[15]: <AxesSubplot:>



Outlier Check

In [16]:
```
outL=sns.boxplot(forest_fire1['area'])
```

```
C:\Users\nishi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWar
ning: Pass the following variable as a keyword arg: x. From version 0.12, the o
nly valid positional argument will be `data`, and passing other arguments witho
ut an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```



We find 3 Outliers in the data.

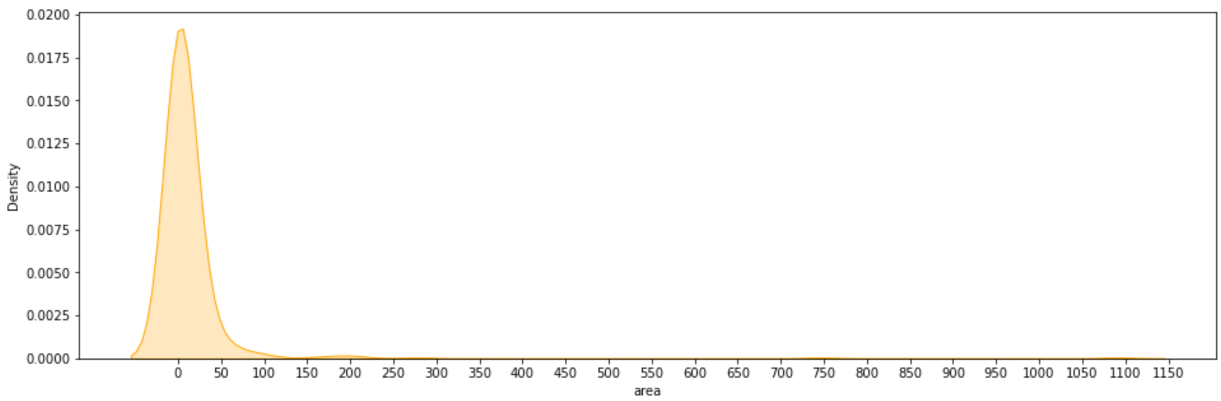
In [17]:
```
plt.rcParams["figure.figsize"] = 9,5
```

In [18]:
```
data=forest_fire1['area']
print(data)
```

```
0       0.00
1       0.00
2       0.00
3       0.00
4       0.00
        ...
512     6.44
513    54.29
514    11.16
515     0.00
516     0.00
Name: area, Length: 517, dtype: float64
```

In [33]:
```
plt.figure(figsize=(16,5))
print("Skew: {}".format(forest_fire1['area'].skew()))
print("Kurtosis: {}".format(forest_fire1['area'].kurtosis()))
outL= sns.kdeplot(forest_fire1['area'],color='orange',shade='True')
plt.xticks([i for i in range(0,1200,50)])
plt.show()
```

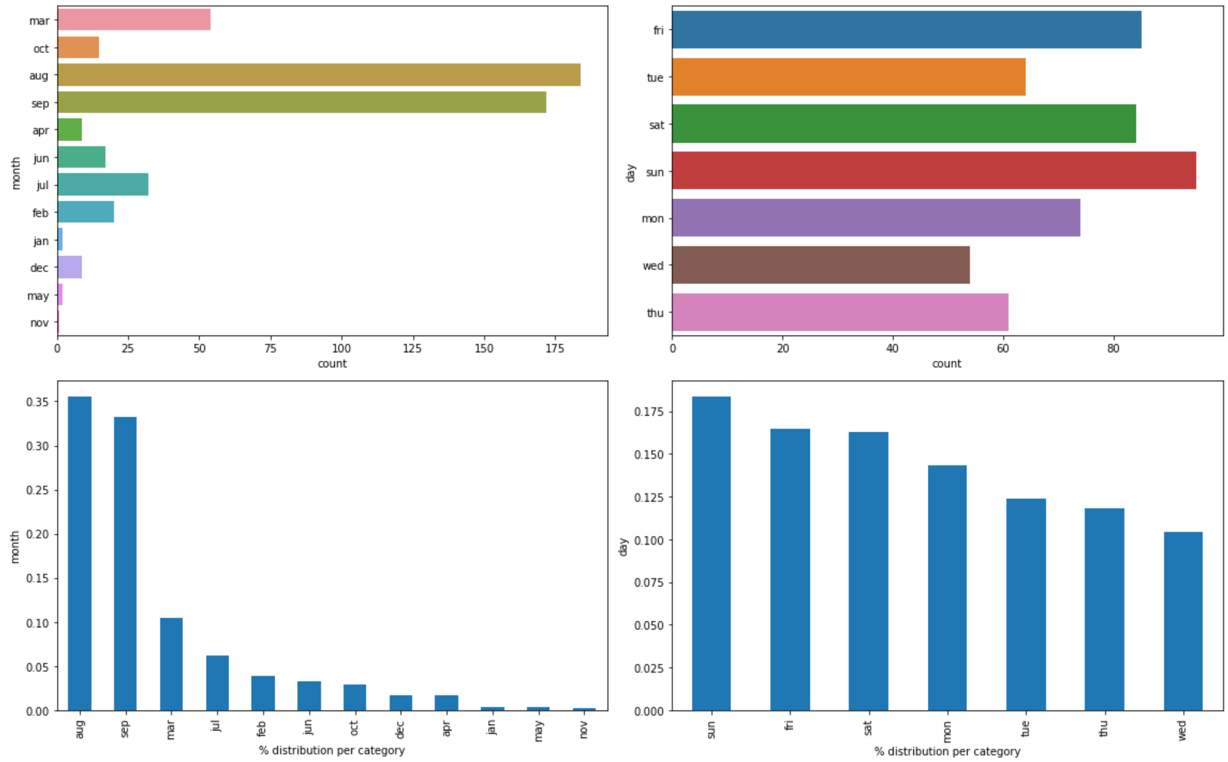
```
Skew: 12.846933533934868
Kurtosis: 194.1407210942299
```



The plot above is skewed to the right and has a high kurtosis value. We observe that most of the forest fire area lies in less than 150 hectares.

In [34]:
```
dfa = forest_fire1[forest_fire1.columns[0:10]]
month_column = dfa.select_dtypes(include='object').columns.tolist()
```

In [35]:
```
plt.figure(figsize=(16,10))
for i,col in enumerate(month_column,1):
    plt.subplot(2,2,i)
    sns.countplot(data=dfa,y=col)
    plt.subplot(2,2,i+2)
    forest_fire1[col].value_counts(normalize=True).plot.bar()
    plt.ylabel(col)
    plt.xlabel('% distribution per category')
plt.tight_layout()
plt.show()
```

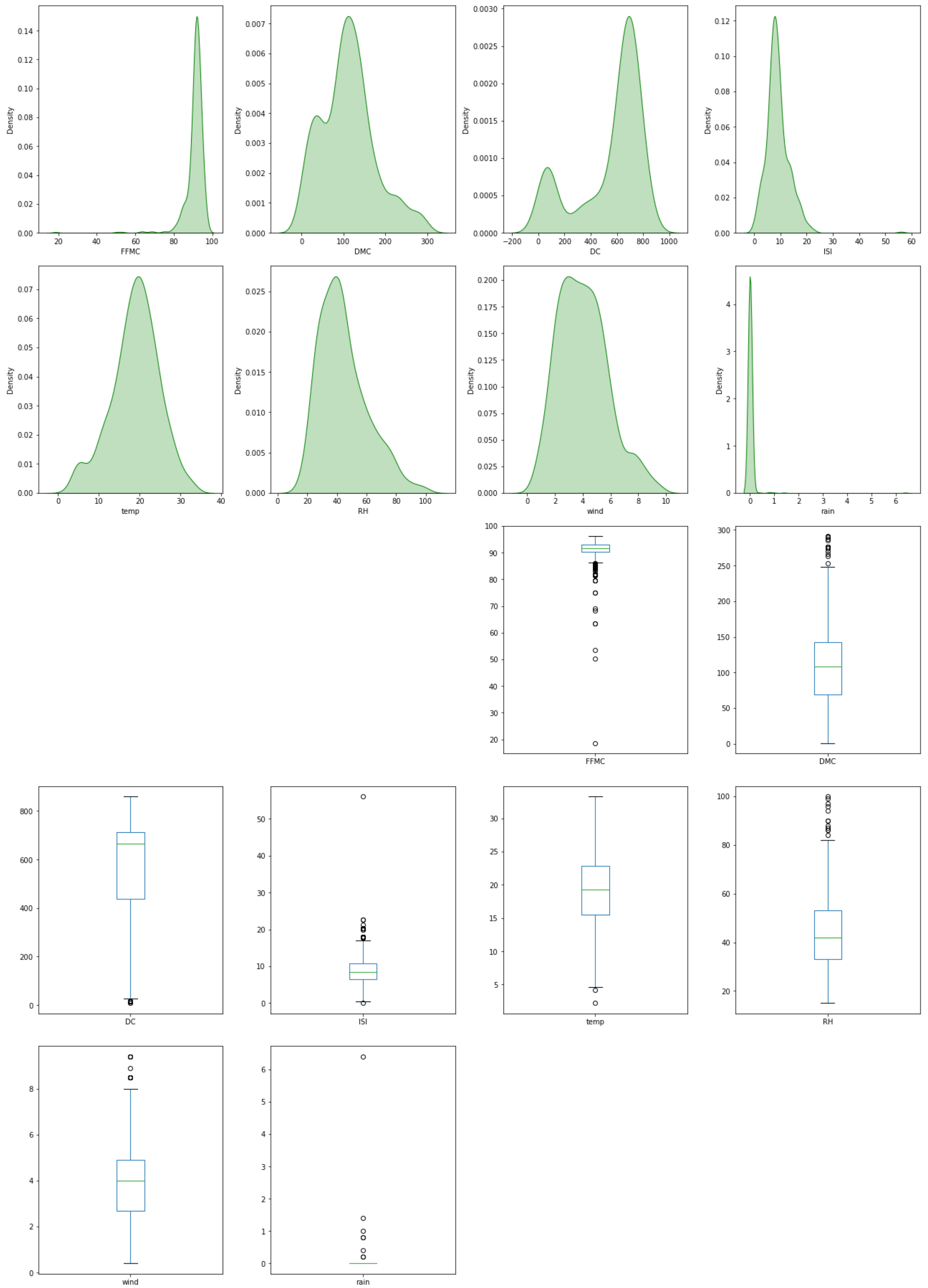


We can conclude that majority of the fires occur in the month of august and september. When we talk about the days the major cases occurr on friday,saturday and sunday.

In [36]:
```
num_columns = dfa.select_dtypes(exclude='object').columns.tolist()
```

In [37]:

```
plt.figure(figsize=(18,40))
for i,col in enumerate(num_columns,1):
    plt.subplot(8,4,i)
    sns.kdeplot(forest_fire[col],color='g',shade=True)
    plt.subplot(8,4,i+10)
    forest_fire[col].plot.box()
plt.tight_layout()
plt.show()
num_data = forest_fire[num_columns]
pd.DataFrame(data=[num_data.skew(),num_data.kurtosis()],index=['skewness','kurtos
```



Out[37]:

	FFMC	DMC	DC	ISI	temp	RH	wind	rain
skewness	-6.575606	0.547498	-1.100445	2.536325	-0.331172	0.862904	0.571001	19.816344
kurtosis	67.066041	0.204822	-0.245244	21.458037	0.136166	0.438183	0.054324	421.295964

SVM

In [38]:
```
X = forest_fire1.iloc[:,2:30]
y = forest_fire1.iloc[:,30]
```

In [39]:
```
mapping = {'small': 1, 'large': 2}
```

In [40]:
```
y = y.replace(mapping)
```

In [41]:
```
x_train,x_test,y_train,y_test = train_test_split(X,y,test_size = 0.20, stratify =
```

Linear

In [45]:
```
model_linear = SVC(kernel = "linear")
model_linear.fit(x_train,y_train)
pred_test_linear = model_linear.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_linear))
```

```
Accuracy: 0.9711538461538461
```

Poly

In [46]:
```
model_poly = SVC(kernel = "poly")
model_poly.fit(x_train,y_train)
pred_test_poly = model_poly.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_poly))
```

```
Accuracy: 0.7403846153846154
```

RBF

In [48]:
```
model_rbf = SVC(kernel = "rbf")
model_rbf.fit(x_train,y_train)
pred_test_rbf = model_rbf.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_rbf))
```

```
Accuracy: 0.7403846153846154
```

Sigmoid

In [49]:
```
model_sigmoid = SVC(kernel = "sigmoid")
model_sigmoid.fit(x_train,y_train)
pred_test_sigmoid = model_sigmoid.predict(x_test)
print("Accuracy:",metrics.accuracy_score(y_test, pred_test_sigmoid))
```

```
Accuracy: 0.7019230769230769
```

CONCLUSION

```
The linear model gives us the best accuracy compared to poly,rbf and sigmoid
model.
```