

In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.preprocessing import LabelEncoder
```

executed in 20ms, finished 16:11:02 2022-01-05

In [3]:

```
df= pd.read_csv("forestfires.csv")
df.head()
```

executed in 310ms, finished 16:11:18 2022-01-05

Out[3]:

	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	

5 rows × 31 columns



In [4]:

df.info()

executed in 802ms, finished 16:11:26 2022-01-05

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 517 entries, 0 to 516

Data columns (total 31 columns):

#	Column	Non-Null Count	Dtype
0	month	517 non-null	object
1	day	517 non-null	object
2	FFMC	517 non-null	float64
3	DMC	517 non-null	float64
4	DC	517 non-null	float64
5	ISI	517 non-null	float64
6	temp	517 non-null	float64
7	RH	517 non-null	int64
8	wind	517 non-null	float64
9	rain	517 non-null	float64
10	area	517 non-null	float64
11	dayfri	517 non-null	int64
12	daymon	517 non-null	int64
13	daysat	517 non-null	int64
14	daysun	517 non-null	int64
15	daythu	517 non-null	int64
16	daytue	517 non-null	int64
17	daywed	517 non-null	int64
18	monthapr	517 non-null	int64
19	monthaug	517 non-null	int64
20	monthdec	517 non-null	int64
21	monthfeb	517 non-null	int64
22	monthjan	517 non-null	int64
23	monthjul	517 non-null	int64
24	monthjun	517 non-null	int64
25	monthmar	517 non-null	int64
26	monthmay	517 non-null	int64
27	monthnov	517 non-null	int64
28	monthoct	517 non-null	int64
29	monthsep	517 non-null	int64
30	size_category	517 non-null	object

dtypes: float64(8), int64(20), object(3)

memory usage: 125.3+ KB

In [4]:

```
df.describe()
```

executed in 162ms, finished 10:22:01 2021-11-27

Out[4]:

	FFMC	DMC	DC	ISI	temp	RH	wind	
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	51
mean	90.644681	110.872340	547.940039	9.021663	18.889168	44.288201	4.017602	
std	5.520111	64.046482	248.066192	4.559477	5.806625	16.317469	1.791653	
min	18.700000	1.100000	7.900000	0.000000	2.200000	15.000000	0.400000	
25%	90.200000	68.600000	437.700000	6.500000	15.500000	33.000000	2.700000	
50%	91.600000	108.300000	664.200000	8.400000	19.300000	42.000000	4.000000	
75%	92.900000	142.400000	713.900000	10.800000	22.800000	53.000000	4.900000	
max	96.200000	291.300000	860.600000	56.100000	33.300000	100.000000	9.400000	

8 rows × 28 columns



In [5]:

```
df.isnull().sum()  
executed in 17ms, finished 10:22:13 2021-11-27
```

Out[5]:

```
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
```

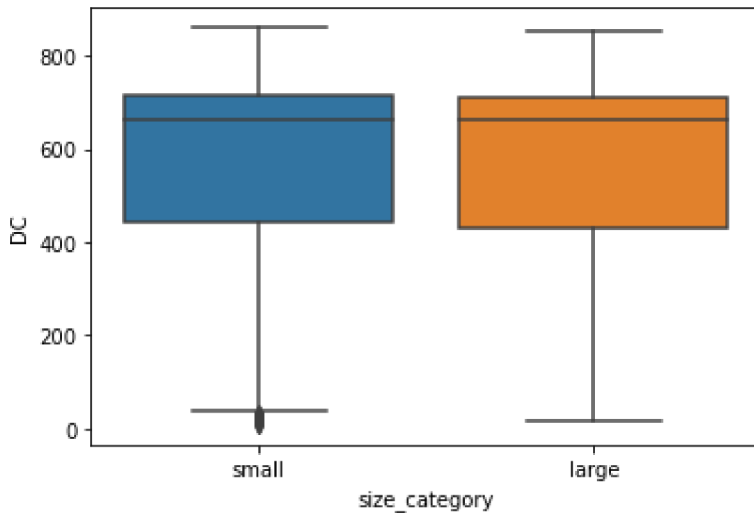
In [6]:

```
x= df.iloc[:, :-1]
y=df.iloc[:, -1]
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
x=x.apply(LabelEncoder().fit_transform)
sns.boxplot(x="size_category", y="DC", data=df)
```

executed in 316ms, finished 10:22:29 2021-11-27

Out[6]:

<AxesSubplot: xlabel='size_category', ylabel='DC'>



In [7]:

```
from sklearn.model_selection import train_test_split
train_x, test_x, train_y, test_y = train_test_split(x, y, test_size = 0.3, random_state=1)
from sklearn.svm import SVC
model_linear = SVC(kernel = "linear")
model_linear.fit(train_x, train_y)
pred_test_linear = model_linear.predict(test_x)

np.mean(pred_test_linear == test_y)
```

executed in 378ms, finished 10:22:44 2021-11-27

Out[7]:

0.9871794871794872

In [8]:

```
model_poly = SVC(kernel = "poly")
model_poly.fit(train_x, train_y)
pred_test_poly = model_poly.predict(test_x)
np.mean(pred_test_poly == test_y)
```

executed in 32ms, finished 10:22:58 2021-11-27

Out[8]:

0.9871794871794872

In [9]:

```
model_rbf = SVC(kernel = "rbf")
model_rbf.fit(train_x,train_y)
pred_test_rbf = model_rbf.predict(test_x)

np.mean(pred_test_rbf==test_y)
```

executed in 47ms, finished 10:23:10 2021-11-27

Out[9]:

0.9743589743589743

In [10]:

```
from sklearn.metrics import classification_report,confusion_matrix
model=SVC()
model.fit(train_x,train_y)
```

executed in 30ms, finished 10:23:23 2021-11-27

Out[10]:

SVC()

In [11]:

```
pred=model.predict(test_x)
print(confusion_matrix(pred,test_y))
print(classification_report(pred,test_y))
```

executed in 61ms, finished 10:23:35 2021-11-27

```
[[ 37  1]
 [ 3 115]]
```

	precision	recall	f1-score	support
large	0.93	0.97	0.95	38
small	0.99	0.97	0.98	118
accuracy			0.97	156
macro avg	0.96	0.97	0.97	156
weighted avg	0.98	0.97	0.97	156

In [12]:

```
np.mean(pred==test_y)*100
```

executed in 14ms, finished 10:23:47 2021-11-27

Out[12]:

97.43589743589743