In [2]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

In [3]:

```
df=pd.read_csv(r"C:\Users\a\Downloads\Algerian_forest_fires_dataset_UPDATE.csv",header=1)
```

In [4]:

```
df.shape
```

Out[4]:

```
(246, 14)
```

In [5]:

```
df
```

Out[5]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Class
0	01	06	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not f
1	02	06	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not f
2	03	06	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not f
3	04	06	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not f
4	05	06	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not f
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
241	26	09	2012	30	65	14	0	85.4	16	44.5	4.5	16.9	6.5	f
242	27	09	2012	28	87	15	4.4	41.1	6.5	8	0.1	6.2	0	not f
243	28	09	2012	27	87	29	0.5	45.9	3.5	7.9	0.4	3.4	0.2	not f
244	29	09	2012	24	54	18	0.1	79.7	4.3	15.2	1.7	5.1	0.7	not f
245	30	09	2012	24	64	15	0.2	67.3	3.8	16.5	1.2	4.8	0.5	not f

246 rows × 14 columns

In [6]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246 entries, 0 to 245
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   day          246 non-null    object
 1   month        245 non-null    object
 2   year         245 non-null    object
 3   Temperature  245 non-null    object
 4    RH          245 non-null    object
 5    Ws          245 non-null    object
 6   Rain         245 non-null    object
 7   FFMC         245 non-null    object
 8   DMC          245 non-null    object
 9   DC           245 non-null    object
 10  ISI          245 non-null    object
 11  BUI          245 non-null    object
 12  FWI          245 non-null    object
 13  Classes      244 non-null    object
dtypes: object(14)
memory usage: 27.0+ KB
```

In [7]:

```
df.isnull().sum()
```

Out[7]:

```
day            0
month          1
year           1
Temperature    1
 RH            1
 Ws            1
Rain           1
FFMC           1
DMC            1
DC             1
ISI            1
BUI            1
FWI            1
Classes        2
dtype: int64
```

In [8]:

```
df.loc[:122,'Region']=1
df.loc[122:,'Region']=2
df[['Region']] = df[['Region']].astype(int)
```

In [9]:

```
df.head(10)
```

Out[9]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
0	01	06	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not fire
1	02	06	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not fire
2	03	06	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire
3	04	06	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not fire
4	05	06	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not fire
5	06	06	2012	31	67	14	0	82.6	5.8	22.2	3.1	7	2.5	fire
6	07	06	2012	33	54	13	0	88.2	9.9	30.5	6.4	10.9	7.2	fire
7	08	06	2012	30	73	15	0	86.6	12.1	38.3	5.6	13.5	7.1	fire
8	09	06	2012	25	88	13	0.2	52.9	7.9	38.8	0.4	10.5	0.3	not fire
9	10	06	2012	28	79	12	0	73.2	9.5	46.3	1.3	12.6	0.9	not fire

In [10]:

```
df.tail(10)
```

Out[10]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Cla
236	21	09	2012	35	34	17	0	92.2	23.6	97.3	13.8	29.4	21.6	
237	22	09	2012	33	64	13	0	88.9	26.1	106.3	7.1	32.4	13.7	
238	23	09	2012	35	56	14	0	89	29.4	115.6	7.5	36	15.2	
239	24	09	2012	26	49	6	2	61.3	11.9	28.1	0.6	11.9	0.4	n
240	25	09	2012	28	70	15	0	79.9	13.8	36.1	2.4	14.1	3	n
241	26	09	2012	30	65	14	0	85.4	16	44.5	4.5	16.9	6.5	
242	27	09	2012	28	87	15	4.4	41.1	6.5	8	0.1	6.2	0	n
243	28	09	2012	27	87	29	0.5	45.9	3.5	7.9	0.4	3.4	0.2	n
244	29	09	2012	24	54	18	0.1	79.7	4.3	15.2	1.7	5.1	0.7	n
245	30	09	2012	24	64	15	0.2	67.3	3.8	16.5	1.2	4.8	0.5	n

In [11]:

```
df =df.dropna().reset_index(drop=True)
df.shape
```

Out[11]:

```
(244, 15)
```

In [12]:

```
df.iloc[[122]]
```

Out[12]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
122	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes

In [13]:

```
df1 = df.drop(122).reset_index(drop=True)
pd.set_option('display.max_rows', None)
df1
```

Out[13]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
0	01	06	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not fire	1
1	02	06	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not fire	1
2	03	06	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire	1
3	04	06	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not fire	1
4	05	06	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not fire	1
5	06	06	2012	31	67	14	0	82.6	5.8	22.2	3.1	7	2.5	fire	1
6	07	06	2012	33	54	13	0	88.2	9.9	30.5	6.4	10.9	7.2	fire	1
7	08	06	2012	30	73	15	0	86.6	12.1	38.3	5.6	13.5	7.1	fire	1
8	09	06	2012	25	88	13	0.2	52.9	7.9	38.8	0.4	10.5	0.3	not fire	1
9	10	06	2012	28	79	12	0	73.2	9.5	46.3	1.3	12.6	0.9	not fire	1

In [14]:

```
df1.shape
```

Out[14]:

```
(243, 15)
```

In [15]:

```
df1.columns
```

Out[15]:

```
Index(['day', 'month', 'year', 'Temperature', ' RH', ' Ws', 'Rain ', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes  ', 'Region'],
      dtype='object')
```

In [16]:

```
df1.columns = df1.columns.str.strip()
df1.columns
```

Out[16]:

```
Index(['day', 'month', 'year', 'Temperature', 'RH', 'Ws', 'Rain', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'Region'],
      dtype='object')
```

In [17]:

```
df1[['month', 'day', 'year', 'Temperature','RH', 'Ws']] = df1[['month', 'day', 'year', 'Tem
```

In [18]:

```
objects = [features for features in df1.columns if df1[features].dtypes=='O']
for i in objects:
    if i != 'Classes':
        df1[i] = df1[i].astype(float)
```

In [19]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 243 entries, 0 to 242
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   day          243 non-null    int32  
 1   month        243 non-null    int32  
 2   year         243 non-null    int32  
 3   Temperature  243 non-null    int32  
 4   RH           243 non-null    int32  
 5   Ws           243 non-null    int32  
 6   Rain         243 non-null    float64
 7   FFMC         243 non-null    float64
 8   DMC          243 non-null    float64
 9   DC           243 non-null    float64
 10  ISI          243 non-null    float64
 11  BUI          243 non-null    float64
 12  FWI          243 non-null    float64
 13  Classes      243 non-null    object 
 14  Region       243 non-null    int32  
dtypes: float64(7), int32(7), object(1)
memory usage: 22.0+ KB
```

In [20]:

```
df1.describe()
```

Out[20]:

	day	month	year	Temperature	RH	Ws	Rain	I
count	243.000000	243.000000	243.0	243.000000	243.000000	243.000000	243.000000	243.00
mean	15.761317	7.502058	2012.0	32.152263	62.041152	15.493827	0.762963	77.84
std	8.842552	1.114793	0.0	3.628039	14.828160	2.811385	2.003207	14.34
min	1.000000	6.000000	2012.0	22.000000	21.000000	6.000000	0.000000	28.60
25%	8.000000	7.000000	2012.0	30.000000	52.500000	14.000000	0.000000	71.85
50%	16.000000	8.000000	2012.0	32.000000	63.000000	15.000000	0.000000	83.30
75%	23.000000	8.000000	2012.0	35.000000	73.500000	17.000000	0.500000	88.30
max	31.000000	9.000000	2012.0	42.000000	90.000000	29.000000	16.800000	96.00

In [21]:

```
df1["Classes"].value_counts()
```

Out[21]:

```
fire             131
not fire         101
fire               4
fire               2
not fire           2
not fire           1
not fire           1
not fire           1
Name: Classes, dtype: int64
```

In [22]:

```
df1.Classes = df1.Classes.str.strip()
```

In [23]:

```
df1["Classes"].value_counts()
```

Out[23]:

```
fire        137
not fire    106
Name: Classes, dtype: int64
```

In [24]:

```
df1[:122]
```

Out[24]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	not fire	1
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	not fire	1
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire	1
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	not fire	1
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	not fire	1
5	6	6	2012	31	67	14	0.0	82.6	5.8	22.2	3.1	7.0	2.5	fire	1
6	7	6	2012	33	54	13	0.0	88.2	9.9	30.5	6.4	10.9	7.2	fire	1
7	8	6	2012	30	73	15	0.0	86.6	12.1	38.3	5.6	13.5	7.1	fire	1
8	9	6	2012	25	88	13	0.2	52.9	7.9	38.8	0.4	10.5	0.3	not fire	1
9	10	6	2012	28	79	12	0.0	73.2	9.5	46.3	1.3	12.6	0.9	not fire	1

In [25]:

```
df1[122:]
```

Out[25]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
122	1	6	2012	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8	0.2	not fire	2
123	2	6	2012	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9	0.2	not fire	2
124	3	6	2012	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6	0.1	not fire	2
125	4	6	2012	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6	1.0	not fire	2
126	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9	not fire	2
127	6	6	2012	35	54	11	0.1	83.7	8.4	26.3	3.1	9.3	3.1	fire	2
128	7	6	2012	35	44	17	0.2	85.6	9.9	28.9	5.4	10.7	6.0	fire	2
129	8	6	2012	28	51	17	1.3	71.4	7.7	7.4	1.5	7.3	0.8	not fire	2
130	9	6	2012	27	59	18	0.1	78.1	8.5	14.7	2.4	8.3	1.9	not fire	2
131	10	6	2012	30	41	15	0.0	89.4	13.3	22.5	8.4	13.1	10.0	fire	2

In [26]:

```
df1.corr()
```

Out[26]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC
day	1.000000	-0.000369	NaN	0.097227	-0.076034	0.047812	-0.112523	0.224956
month	-0.000369	1.000000	NaN	-0.056781	-0.041252	-0.039880	0.034822	0.017030
year	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Temperature	0.097227	-0.056781	NaN	1.000000	-0.651400	-0.284510	-0.326492	0.676568
RH	-0.076034	-0.041252	NaN	-0.651400	1.000000	0.244048	0.222356	-0.644873
Ws	0.047812	-0.039880	NaN	-0.284510	0.244048	1.000000	0.171506	-0.166548
Rain	-0.112523	0.034822	NaN	-0.326492	0.222356	0.171506	1.000000	-0.543906
FFMC	0.224956	0.017030	NaN	0.676568	-0.644873	-0.166548	-0.543906	1.000000
DMC	0.491514	0.067943	NaN	0.485687	-0.408519	-0.000721	-0.288773	0.603608
DC	0.527952	0.126511	NaN	0.376284	-0.226941	0.079135	-0.298023	0.507397
ISI	0.180543	0.065608	NaN	0.603871	-0.686667	0.008532	-0.347484	0.740007
BUI	0.517117	0.085073	NaN	0.459789	-0.353841	0.031438	-0.299852	0.592011
FWI	0.350781	0.082639	NaN	0.566670	-0.580957	0.032368	-0.324422	0.691132
Region	0.000821	0.001857	NaN	0.269555	-0.402682	-0.181160	-0.040013	0.222241

In [27]:

```
df1.drop(columns=['year'],axis=1,inplace=True)
```

In [28]:

```
df1.duplicated().sum()
```

Out[28]:

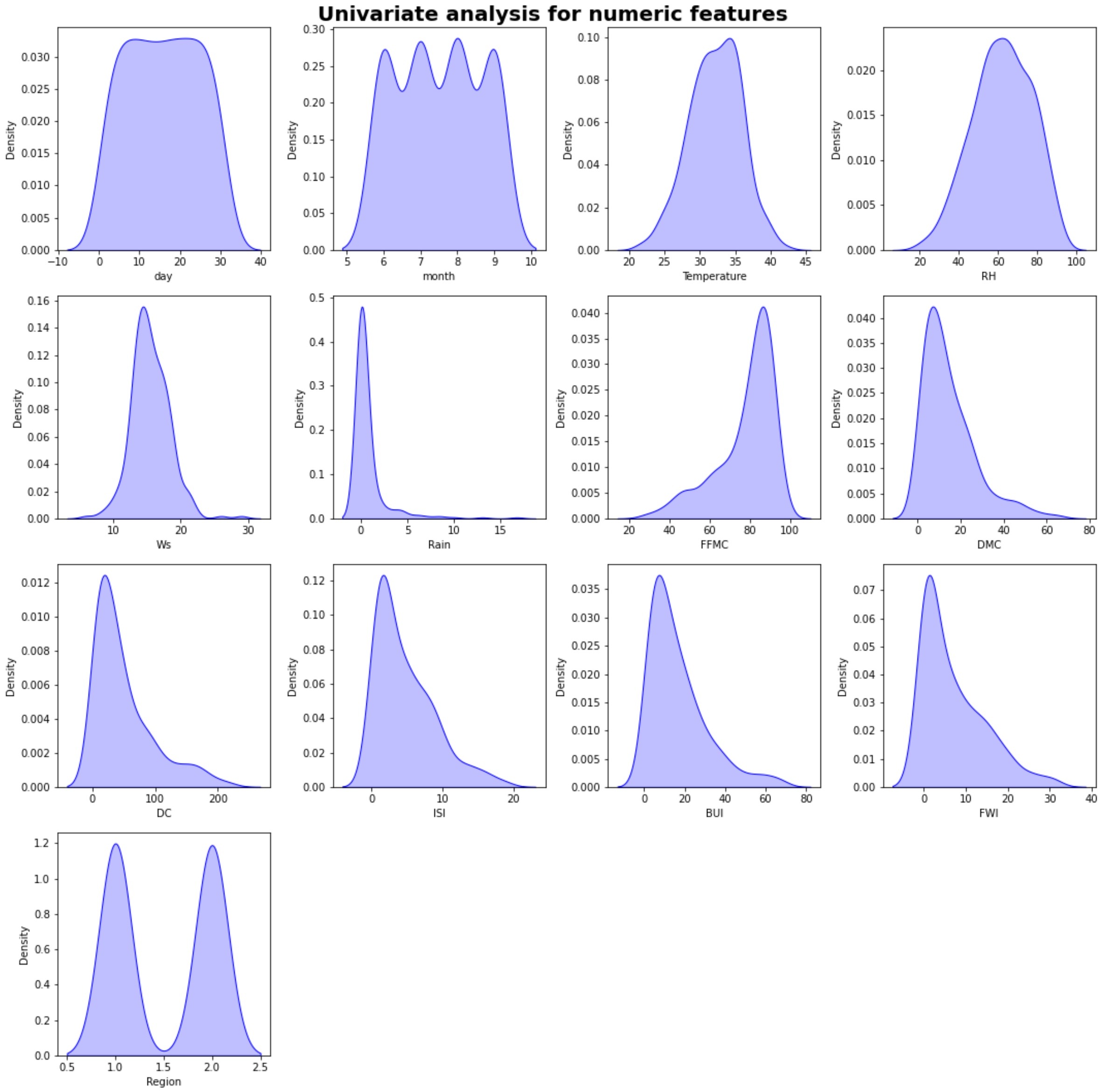
```
0
```

In [29]:

```
numeric_col=[col for col in df1.columns if df1[col].dtype!='O']
categorical_col=[col for col in df1.columns if df1[col].dtype=='O']
```

In [30]:

```
plt.figure(figsize=(15,15))
plt.suptitle('Univariate analysis for numeric features',fontsize=20,fontweight='bold')
for i in range(0,len(numeric_col)):
    plt.subplot(4,4,i+1)
    sns.kdeplot(x=df1[numeric_col[i]],shade=True,color='b')
    plt.xlabel(numeric_col[i])
    plt.tight_layout()
```



In [31]:

```
df1['Classes'].unique()
```

Out[31]:

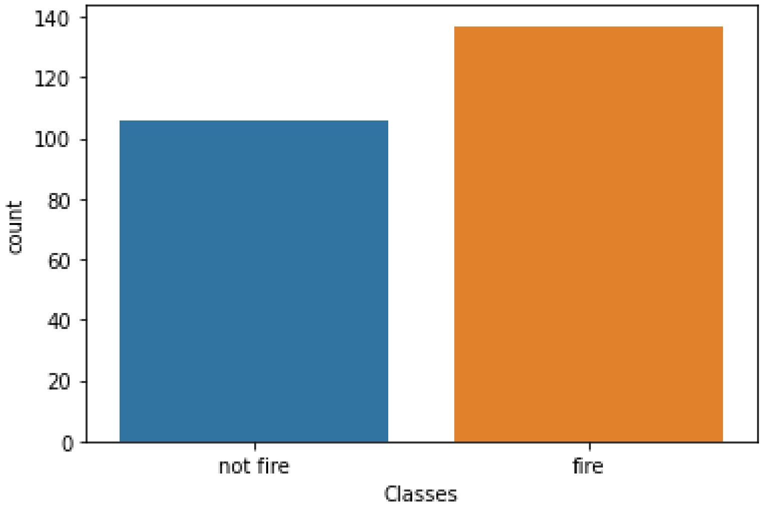
```
array(['not fire', 'fire'], dtype=object)
```

In [32]:

```
sns.countplot(df1['Classes'])
```

Out[32]:

```
<AxesSubplot:xlabel='Classes', ylabel='count'>
```

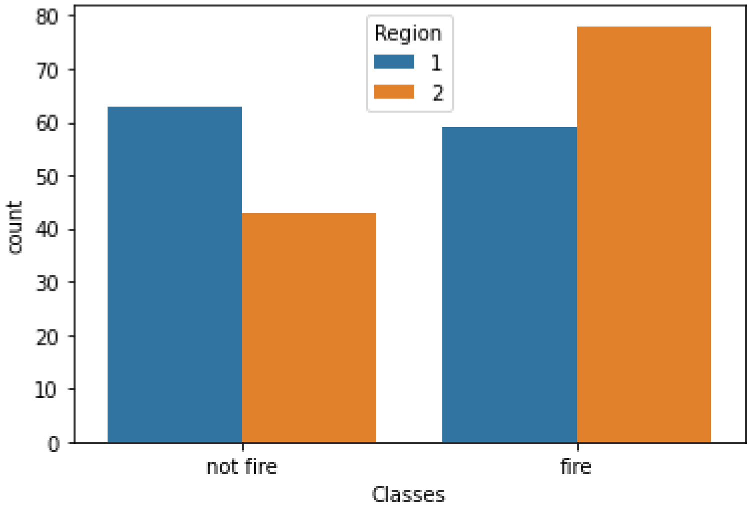


In [33]:

```
sns.countplot(df1['Classes'],hue=df1['Region'])
```

Out[33]:

```
<AxesSubplot:xlabel='Classes', ylabel='count'>
```

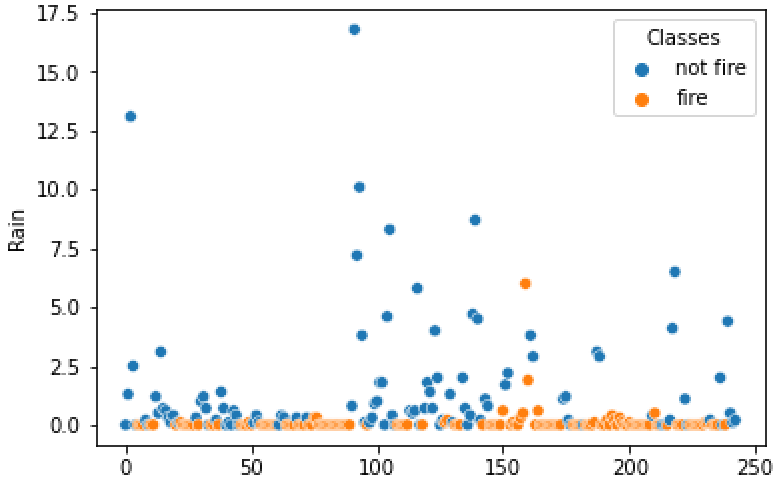


In [34]:

```
sns.scatterplot(x=df1.index,y='Rain',data=df1,hue=df1['Classes'])
```

Out[34]:

```
<AxesSubplot:ylabel='Rain'>
```

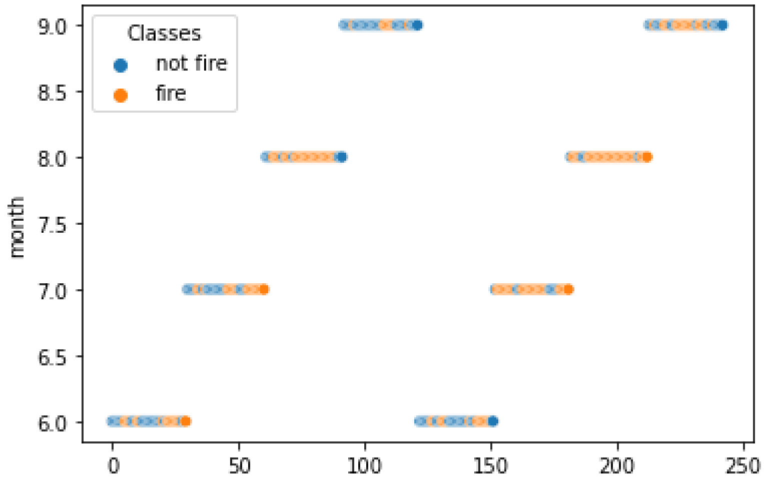


In [35]:

```
sns.scatterplot(x=df1.index,y='month',data=df1,hue=df1['Classes'])
```

Out[35]:

```
<AxesSubplot:ylabel='month'>
```

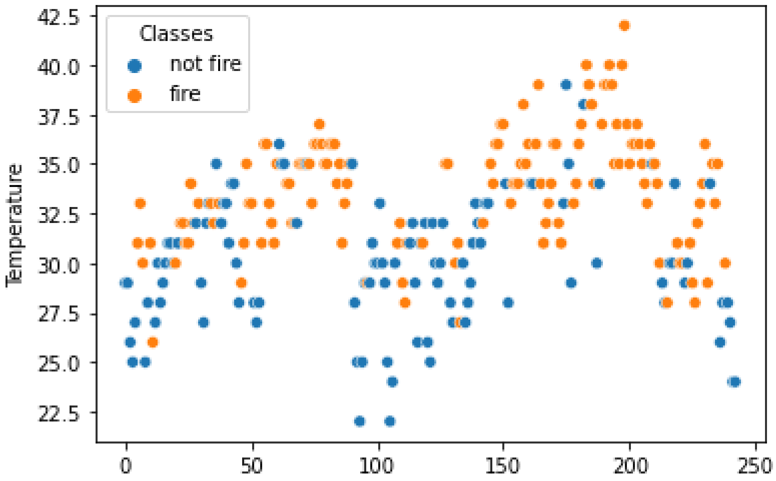


In [36]:

```
sns.scatterplot(x=df1.index,y='Temperature',data=df1,hue=df1['Classes'])
```

Out[36]:

```
<AxesSubplot:ylabel='Temperature'>
```

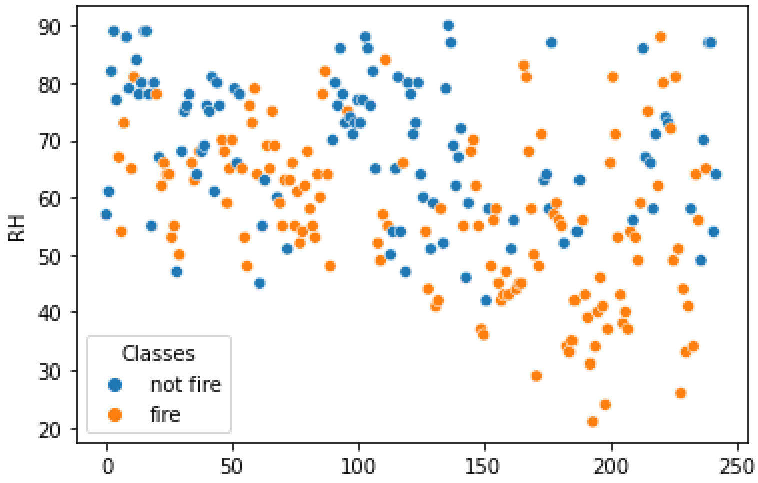


In [37]:

```
sns.scatterplot(x=df1.index,y='RH',data=df1,hue=df1['Classes'])
```

Out[37]:

```
<AxesSubplot:ylabel='RH'>
```

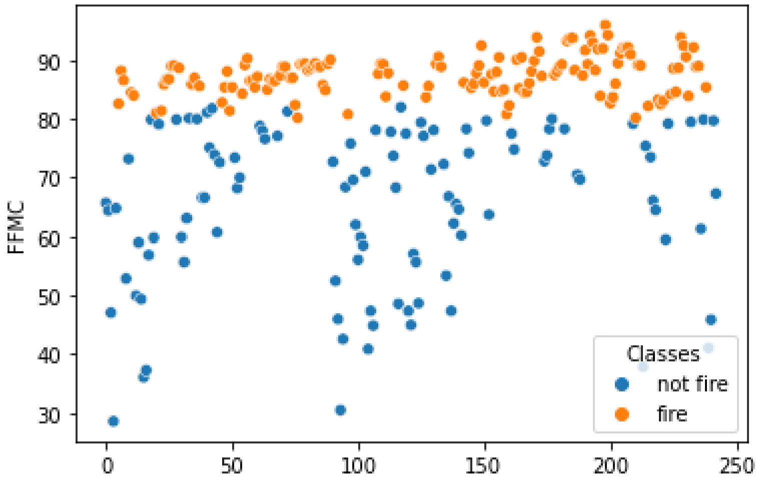


In [38]:

```
sns.scatterplot(x=df1.index,y='FFMC',data=df1,hue=df1['Classes'])
```

Out[38]:

```
<AxesSubplot:ylabel='FFMC'>
```

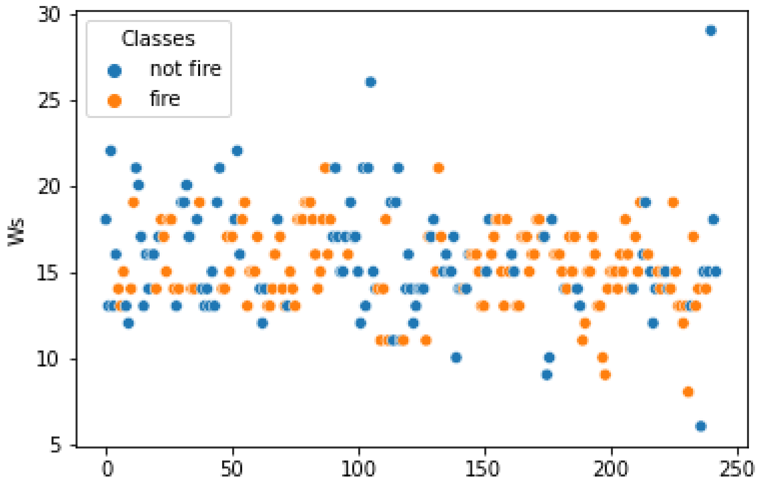


In [39]:

```
sns.scatterplot(x=df1.index,y='Ws',data=df1,hue=df1['Classes'])
```

Out[39]:

```
<AxesSubplot:ylabel='Ws'>
```

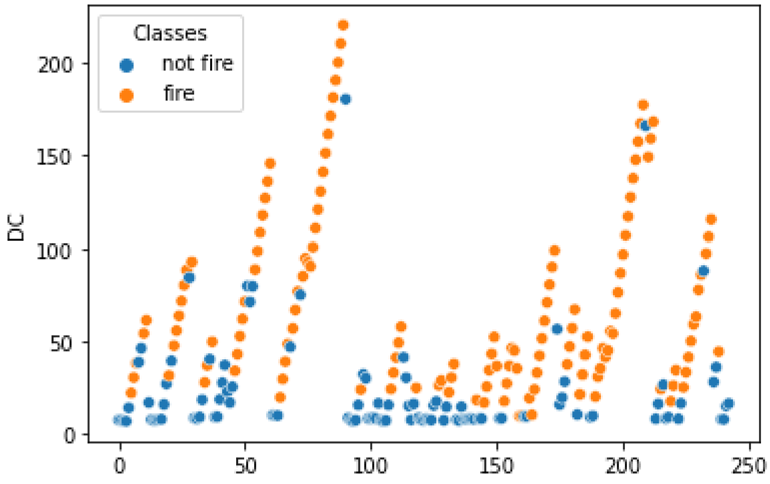


In [40]:

```
sns.scatterplot(x=df1.index,y='DC',data=df1,hue=df1['Classes'])
```

Out[40]:

```
<AxesSubplot:ylabel='DC'>
```

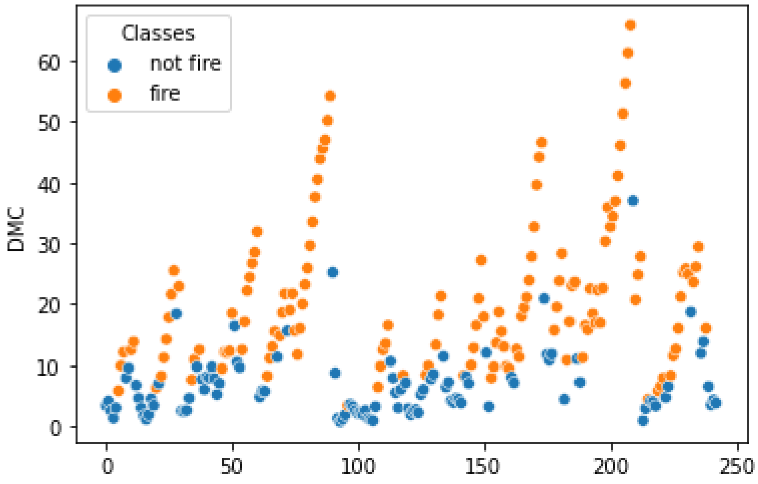


In [41]:

```
sns.scatterplot(x=df1.index,y='DMC',data=df1,hue=df1['Classes'])
```

Out[41]:

```
<AxesSubplot:ylabel='DMC'>
```

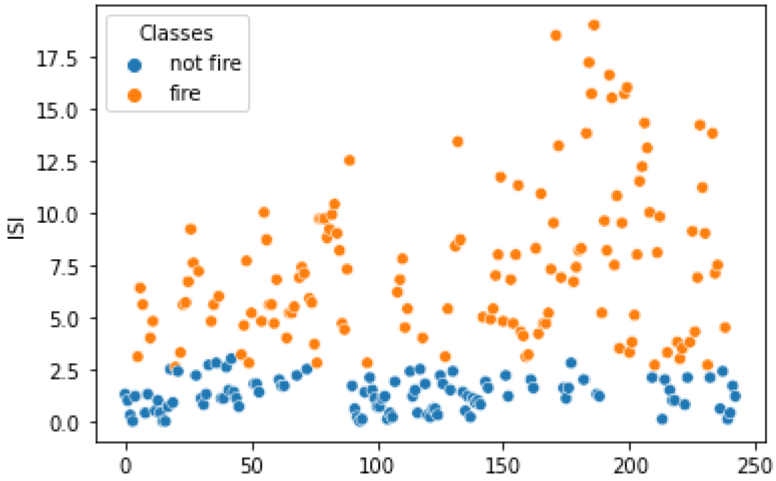


In [42]:

```
sns.scatterplot(x=df1.index,y='ISI',data=df1,hue=df1['Classes'])
```

Out[42]:

```
<AxesSubplot:ylabel='ISI'>
```

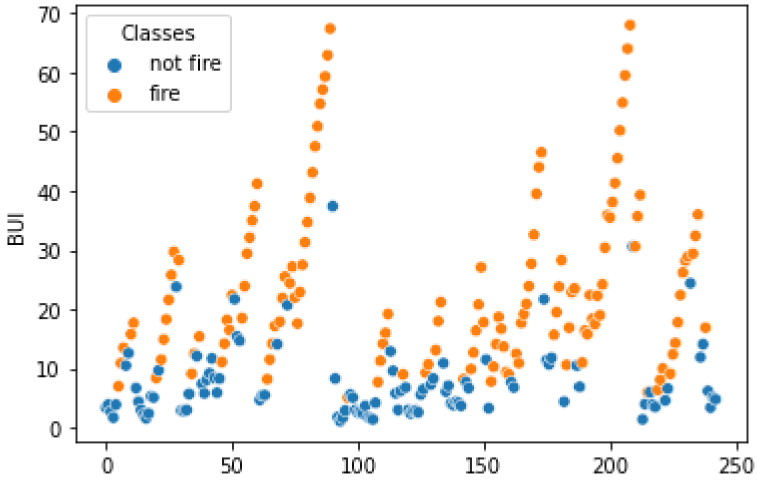


In [43]:

```
sns.scatterplot(x=df1.index,y='BUI',data=df1,hue=df1['Classes'])
```

Out[43]:

```
<AxesSubplot:ylabel='BUI'>
```

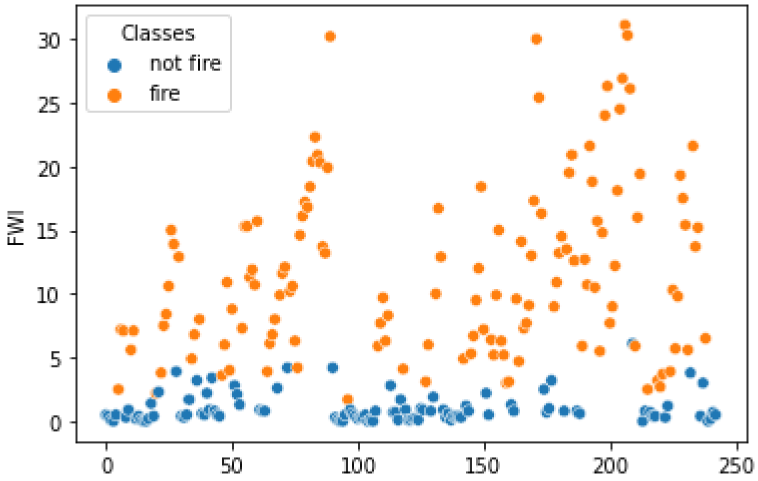


In [44]:

```
sns.scatterplot(x=df1.index,y='FWI',data=df1,hue=df1['Classes'])
```

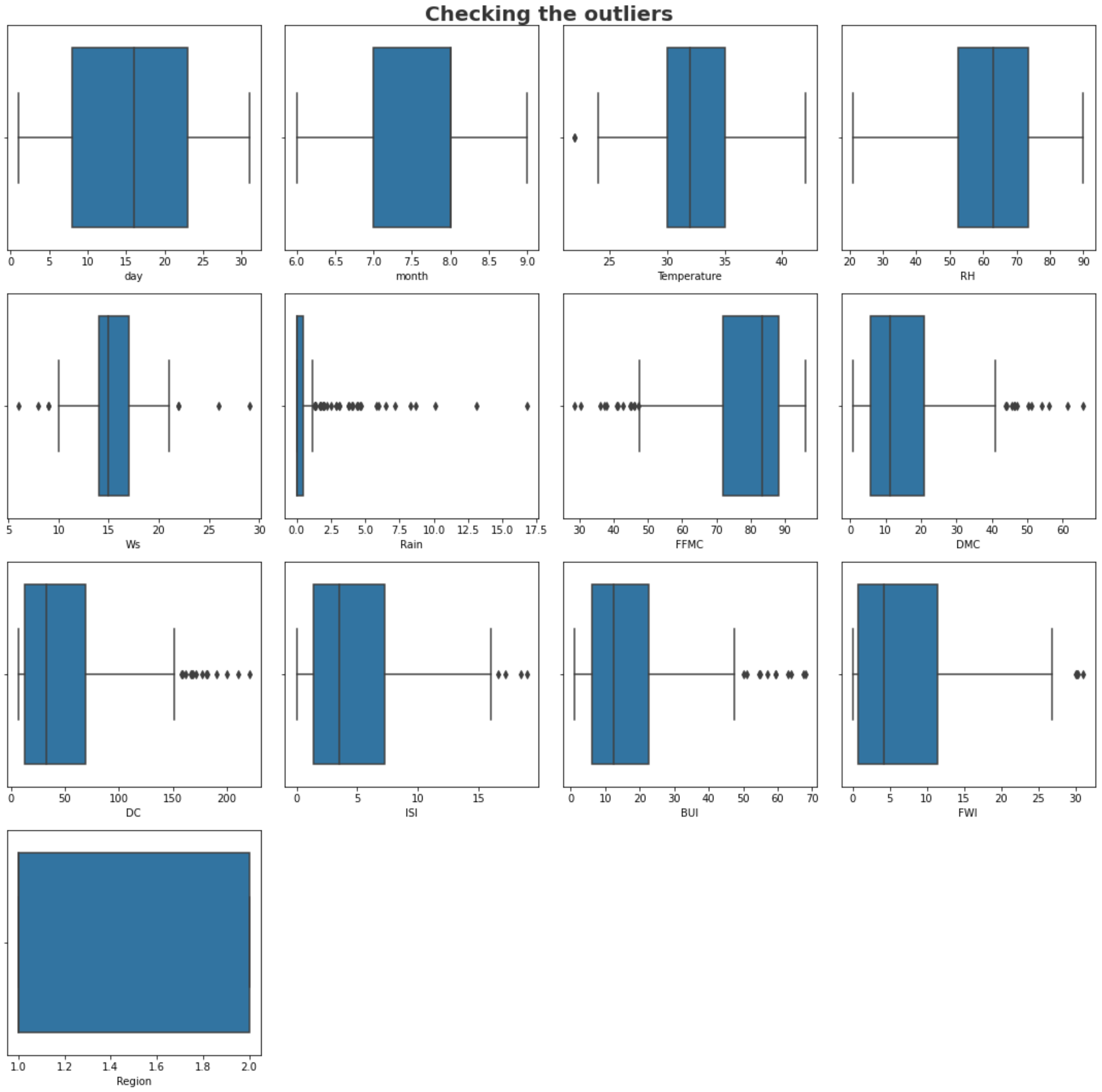
Out[44]:

```
<AxesSubplot:ylabel='FWI'>
```



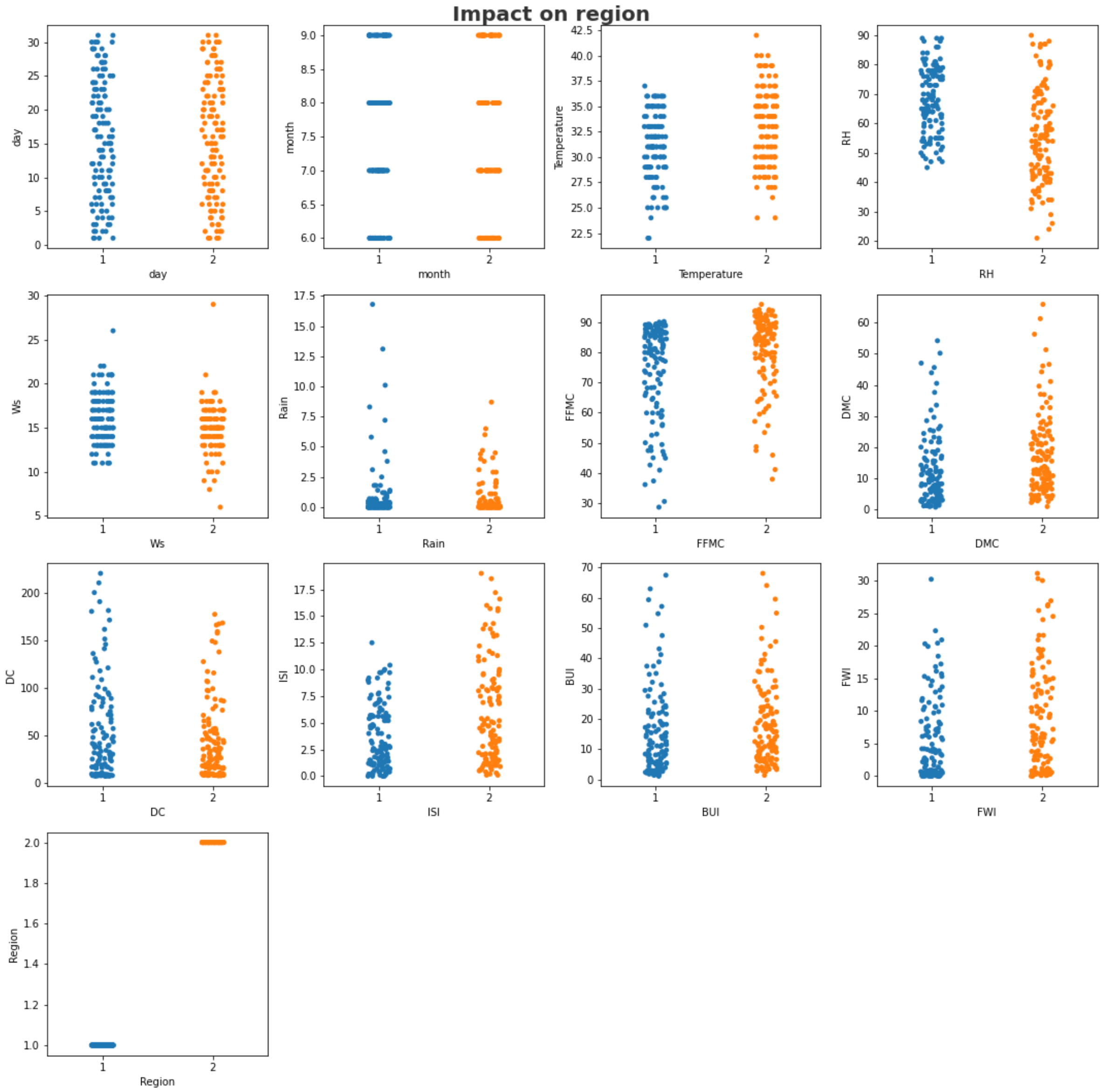
In [45]:

```
plt.figure(figsize=(15,15))
plt.suptitle('Checking the outliers',fontsize=20,fontweight='bold',alpha=0.8)
for i in range(0,len(numeric_col)):
    plt.subplot(4,4,i+1)
    sns.boxplot(x=df1[numeric_col[i]],hue=df1['Classes'])
    plt.xlabel(numeric_col[i])
    plt.tight_layout()
```



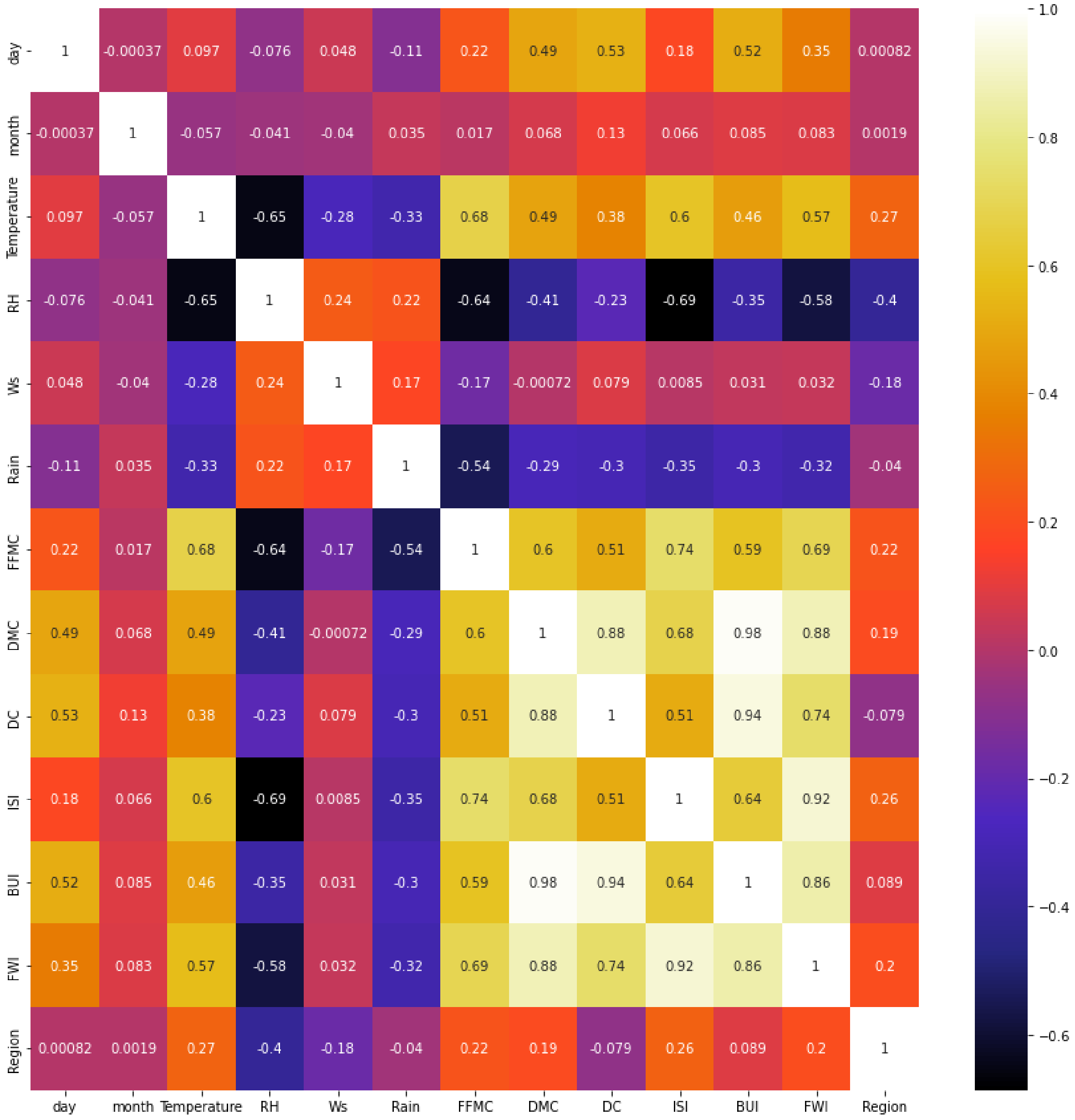
In [46]:

```
plt.figure(figsize=(15,15))
plt.suptitle('Impact on region',fontsize=20,fontweight='bold',alpha=0.8)
for i in range(0,len(numeric_col)):
    plt.subplot(4,4,i+1)
    sns.stripplot(x='Region',y=numeric_col[i],data=df1)
    plt.xlabel(numeric_col[i])
    plt.tight_layout()
```



In [47]:

```
plt.figure(figsize=(15,15))
sns.heatmap(df1.corr(),cmap='CMRmap',annot=True)
plt.show()
```

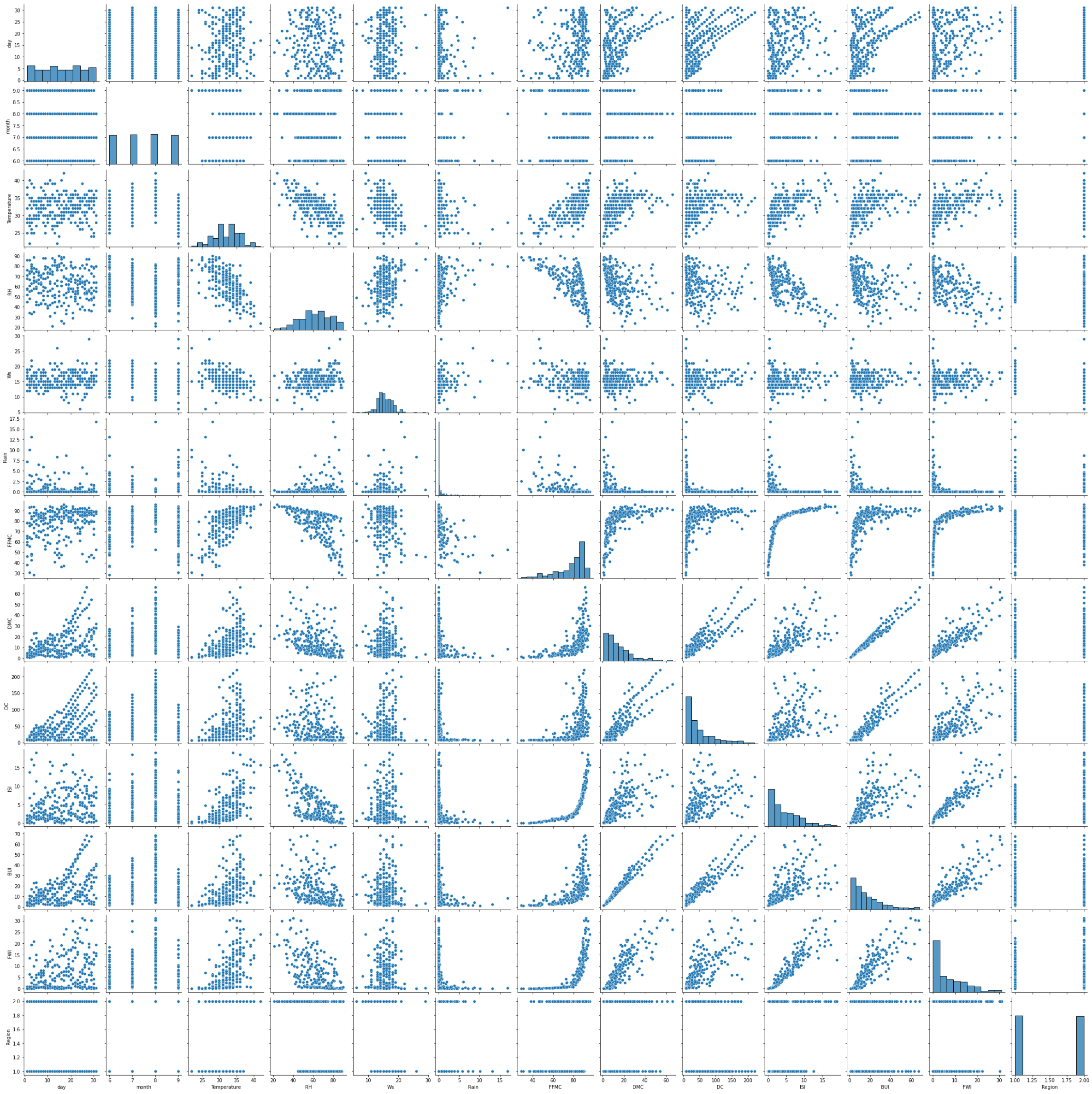


In [48]:

```
sns.pairplot(data=df1)
```

Out[48]:

```
<seaborn.axisgrid.PairGrid at 0x16983dea790>
```



In [49]:

```
df1.drop(columns=['BUI'],axis=1,inplace=True)
```

In [50]:

```
df1
```

Out[50]:

	day	month	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	FWI	Classes	Region
0	1	6	29	57	18	0.0	65.7	3.4	7.6	1.3	0.5	not fire	1
1	2	6	29	61	13	1.3	64.4	4.1	7.6	1.0	0.4	not fire	1
2	3	6	26	82	22	13.1	47.1	2.5	7.1	0.3	0.1	not fire	1
3	4	6	25	89	13	2.5	28.6	1.3	6.9	0.0	0.0	not fire	1
4	5	6	27	77	16	0.0	64.8	3.0	14.2	1.2	0.5	not fire	1
5	6	6	31	67	14	0.0	82.6	5.8	22.2	3.1	2.5	fire	1
6	7	6	33	54	13	0.0	88.2	9.9	30.5	6.4	7.2	fire	1
7	8	6	30	73	15	0.0	86.6	12.1	38.3	5.6	7.1	fire	1
8	9	6	25	88	13	0.2	52.9	7.9	38.8	0.4	0.3	not fire	1
9	10	6	28	79	12	0.0	73.2	9.5	46.3	1.3	0.9	not fire	1

In [51]:

```
df1.drop(columns=['FWI'],axis=1,inplace=True)
```

In [52]:

```
df1['Classes']= np.where(df1['Classes']== 'not fire',0,1)
df1.head(10)
```

Out[52]:

	day	month	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	Classes	Region
0	1	6	29	57	18	0.0	65.7	3.4	7.6	1.3	0	1
1	2	6	29	61	13	1.3	64.4	4.1	7.6	1.0	0	1
2	3	6	26	82	22	13.1	47.1	2.5	7.1	0.3	0	1
3	4	6	25	89	13	2.5	28.6	1.3	6.9	0.0	0	1
4	5	6	27	77	16	0.0	64.8	3.0	14.2	1.2	0	1
5	6	6	31	67	14	0.0	82.6	5.8	22.2	3.1	1	1
6	7	6	33	54	13	0.0	88.2	9.9	30.5	6.4	1	1
7	8	6	30	73	15	0.0	86.6	12.1	38.3	5.6	1	1
8	9	6	25	88	13	0.2	52.9	7.9	38.8	0.4	0	1
9	10	6	28	79	12	0.0	73.2	9.5	46.3	1.3	0	1

In [53]:

```
df1.tail(10)
```

Out[53]:

	day	month	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	Classes	Region
233	21	9	35	34	17	0.0	92.2	23.6	97.3	13.8	1	2
234	22	9	33	64	13	0.0	88.9	26.1	106.3	7.1	1	2
235	23	9	35	56	14	0.0	89.0	29.4	115.6	7.5	1	2
236	24	9	26	49	6	2.0	61.3	11.9	28.1	0.6	0	2
237	25	9	28	70	15	0.0	79.9	13.8	36.1	2.4	0	2
238	26	9	30	65	14	0.0	85.4	16.0	44.5	4.5	1	2
239	27	9	28	87	15	4.4	41.1	6.5	8.0	0.1	0	2
240	28	9	27	87	29	0.5	45.9	3.5	7.9	0.4	0	2
241	29	9	24	54	18	0.1	79.7	4.3	15.2	1.7	0	2
242	30	9	24	64	15	0.2	67.3	3.8	16.5	1.2	0	2

In [54]:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.metrics import accuracy_score,confusion_matrix
```

In [55]:

```
x=df1.drop(columns=['Classes'])
```

In [56]:

```
y=df1['Classes']
```

In [57]:

```
vif=pd.DataFrame()
vif['vif']=[variance_inflation_factor(x,i) for i in range(x.shape[1])]
vif['Features']=x.columns
```

In [58]:

```
vif
```

Out[58]:

	vif	Features
0	6.070685	day
1	40.717203	month
2	107.492884	Temperature
3	33.644586	RH
4	35.210384	Ws
5	1.710823	Rain
6	98.824528	FFMC
7	20.207994	DMC
8	14.124657	DC
9	9.125718	ISI
10	15.044862	Region

In [59]:

```
x.drop(columns=['Temperature'],axis=1,inplace=True)
```

In [60]:

```
vif=pd.DataFrame()
vif['vif']=[variance_inflation_factor(x,i) for i in range(x.shape[1])]
vif['Features']=x.columns
```

In [61]:

```
vif
```

Out[61]:

	vif	Features
0	6.061376	day
1	39.295956	month
2	32.170676	RH
3	35.166048	Ws
4	1.634709	Rain
5	50.761641	FFMC
6	20.197410	DMC
7	14.121837	DC
8	9.120342	ISI
9	14.243279	Region

In [62]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=42)
```

In [63]:

```
from sklearn.preprocessing import StandardScaler
```

In [64]:

```
scaler=StandardScaler()
```

In [65]:

```
x_train=scaler.fit_transform(x_train)
```

In [66]:

```
x_test=scaler.transform(x_test)
```

In [67]:

```
logistic=LogisticRegression()
```

In [68]:

```
logistic.fit(x_train,y_train)
```

Out[68]:

```
▾ LogisticRegression
LogisticRegression()
```

In [69]:

```
logistic_pred=logistic.predict(x_test)
```

In [70]:

```
accuracy=accuracy_score(y_test,logistic_pred)
```

In [71]:

```
accuracy
```

Out[71]:

```
0.958904109589041
```

In [72]:

```
con_mat=confusion_matrix(y_test,logistic_pred)
```

In [73]:

```
con_mat
```

Out[73]:

```
array([[27,  2],
       [ 1, 43]], dtype=int64)
```

In [74]:

```
true_positive=con_mat[0][0]
false_positive=con_mat[0][1]
false_negative=con_mat[1][0]
true_negative=con_mat[1][1]
```

In [75]:

```
precision=true_positive/(true_positive+false_positive)
```

In [76]:

```
precision
```

Out[76]:

```
0.9310344827586207
```

In [77]:

```
recall=true_positive/(true_positive+false_negative)
```

In [78]:

```
recall
```

Out[78]:

```
0.9642857142857143
```

In [79]:

```
f1_score=2*(precision*recall)/(precision+recall)
```

In [80]:

```
f1_score
```

Out[80]:

```
0.9473684210526316
```

Handling imbalance data usig Smote

In [81]:

```
from imblearn.over_sampling import SMOTE
```

In [95]:

```
!pip install imbalanced-learn
```

```
Collecting imbalanced-learn
  Downloading imbalanced_learn-0.9.1-py3-none-any.whl (199 kB)

WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
    WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3
\lib\site-packages)
ERROR: Could not install packages due to an OSError: [WinError 5] Access is
denied: 'C:\\Users\\a\\anaconda3\\Lib\\site-packages\\~klearn\\linear_model
\\_cd_fast.cp39-win_amd64.pyd'
Consider using the `--user` option or check the permissions.

WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)


Collecting scikit-learn>=1.1.0
  Downloading scikit_learn-1.1.3-cp39-cp39-win_amd64.whl (7.6 MB)
Requirement already satisfied: joblib>=1.0.0 in c:\users\a\anaconda3\lib\sit
e-packages (from imbalanced-learn) (1.1.0)
Requirement already satisfied: scipy>=1.3.2 in c:\users\a\anaconda3\lib\site
-packages (from imbalanced-learn) (1.7.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\a\anaconda3
\lib\site-packages (from imbalanced-learn) (2.2.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\a\anaconda3\lib\sit
e-packages (from imbalanced-learn) (1.20.3)
Installing collected packages: scikit-learn, imbalanced-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 0.24.2
    Uninstalling scikit-learn-0.24.2:
      Successfully uninstalled scikit-learn-0.24.2
```

In [97]:

```
!pip install imblearn
```

```
Collecting imblearn
  Downloading imblearn-0.0-py2.py3-none-any.whl (1.9 kB)
Collecting imbalanced-learn
  Using cached imbalanced_learn-0.9.1-py3-none-any.whl (199 kB)
Requirement already satisfied: scikit-learn>=1.1.0 in c:\users\a\anaconda3\l
ib\site-packages (from imbalanced-learn->imblearn) (1.1.3)
Requirement already satisfied: scipy>=1.3.2 in c:\users\a\anaconda3\lib\site
-packages (from imbalanced-learn->imblearn) (1.7.1)
Requirement already satisfied: joblib>=1.0.0 in c:\users\a\anaconda3\lib\sit
e-packages (from imbalanced-learn->imblearn) (1.1.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\a\anaconda3\lib\sit
e-packages (from imbalanced-learn->imblearn) (1.20.3)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\a\anaconda3
\lib\site-packages (from imbalanced-learn->imblearn) (2.2.0)
Installing collected packages: imbalanced-learn, imblearn
Successfully installed imbalanced-learn-0.9.1 imblearn-0.0
```

```
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
WARNING: Ignoring invalid distribution -tatsmodels (c:\users\a\anaconda3\lib
\site-packages)
```

In [82]:

```
smt=SMOTE(sampling_strategy='minority',random_state=42)
```

In [83]:

```
x_smt,y_smt=smt.fit_resample(x,y)
```

In [84]:

```
y_smt.value_counts()
```

Out[84]:

```
0    137
1    137
Name: Classes, dtype: int64
```

In [85]:

```
x_train_smt,x_test_smt,y_train_smt,y_test_smt=train_test_split(x_smt,y_smt)
```

In [86]:

```
logis=LogisticRegression()
```

In [87]:

```
logis.fit(x_train_smt,y_train_smt)
```

Out[87]:

```
▾ LogisticRegression
LogisticRegression()
```

In [91]:

```
logis_pred=logis.predict(x_test_smt)
```

In [92]:

```
y_test_smt
```

Out[92]:

```
246    0
223    0
269    0
30     0
55     1
259    0
225    1
138    0
22     1
232    0
8      0
100    0
143    0
221    1
214    0
164    1
62     0
3      0
13     0
241    0
87     1
109    1
69     1
181    1
117    0
265    0
202    1
213    0
217    0
224    1
209    0
182    0
211    1
83     1
97     0
25     1
102    0
21     0
84     1
147    1
134    0
267    0
238    1
237    0
103    0
236    0
180    1
47     1
6      1
33     0
46     1
58     1
228    1
24     1
16     0
```

```
56     1
40     0
101    0
52     0
60     1
41     0
235    1
136    0
188    0
129    0
26     1
150    1
183    1
249    0
Name: Classes, dtype: int32
```

In [93]:

```
accuracy_smt=accuracy_score(y_test_smt,logis_pred)
accuracy
```

Out[93]:

```
0.958904109589041
```

In [95]:

```
con_mat_smt=confusion_matrix(y_test_smt,logis_pred)
con_mat_smt
```

Out[95]:

```
array([[39,  0],
       [ 0, 30]], dtype=int64)
```

In [96]:

```
true_positive=con_mat_smt[0][0]
false_positive=con_mat_smt[0][1]
false_negative=con_mat_smt[1][0]
true_negative=con_mat_smt[1][1]
```

In [97]:

```
precision_smt=true_positive/(true_positive+false_positive)
precision_smt
```

Out[97]:

```
1.0
```

In [98]:

```
recall=true_positive/(true_positive+false_negative)
recall
```

Out[98]:

```
1.0
```

In [100]:

```
f1_score=2*(precision_smt*recall)/(precision_smt+recall)
f1_score
```

Out[100]:

```
1.0
```