Importing Libraries

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings

warnings.filterwarnings('ignore')
```

Loading the dataset

In [2]:

```
df = pd.read_csv("Algerian_forest_fires_dataset_UPDATE.csv", )
```

Checking first 5 rows of the dataset

In [3]:

```
df.head()
```

Out[3]:

													Bejaia Region Dataset
day	**month**	**year**	**Temperature**	**RH**	**Ws**	**Rain**	**FFMC**	**DMC**	**DC**	**ISI**	**BUI**	**FWI**	Classes
01	**06**	**2012**	**29**	**57**	**18**	**0**	**65.7**	**3.4**	**7.6**	**1.3**	**3.4**	**0.5**	not fire
02	**06**	**2012**	**29**	**61**	**13**	**1.3**	**64.4**	**4.1**	**7.6**	**1**	**3.9**	**0.4**	not fire
03	**06**	**2012**	**26**	**82**	**22**	**13.1**	**47.1**	**2.5**	**7.1**	**0.3**	**2.7**	**0.1**	not fire
04	**06**	**2012**	**25**	**89**	**13**	**2.5**	**28.6**	**1.3**	**6.9**	**0**	**1.7**	**0**	not fire

OBSERVATION : Since the dataset contains Bejaia Region data & Sidi-Bel Abbes Region Data we're going to read it separately.

Reading the Bejaia Region data from CSV file.

In [4]:
```
bejaia_df = pd.read_csv('Algerian_forest_fires_dataset_UPDATE.csv',
bejaia_df.head()
```

Out[4]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Clas
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	no
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	no
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	no
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	no
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	no

Addition of Region Column

In [5]:
```
bejaia_df["Region"] = "Bejaia"
bejaia_df
```

Out[5]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	C
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
117	26	9	2012	31	54	11	0.0	82.0	6.0	16.3	2.5	6.2	1.7	
118	27	9	2012	31	66	11	0.0	85.7	8.3	24.9	4.0	9.0	4.1	
119	28	9	2012	32	47	14	0.7	77.5	7.1	8.8	1.8	6.8	0.9	
120	29	9	2012	26	80	16	1.8	47.4	2.9	7.7	0.3	3.0	0.1	
121	30	9	2012	25	78	14	1.4	45.0	1.9	7.5	0.2	2.4	0.1	

In [6]:
```
bejaia_df
```

Out[6]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	C
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
117	26	9	2012	31	54	11	0.0	82.0	6.0	16.3	2.5	6.2	1.7	
118	27	9	2012	31	66	11	0.0	85.7	8.3	24.9	4.0	9.0	4.1	
119	28	9	2012	32	47	14	0.7	77.5	7.1	8.8	1.8	6.8	0.9	
120	29	9	2012	26	80	16	1.8	47.4	2.9	7.7	0.3	3.0	0.1	
121	30	9	2012	25	78	14	1.4	45.0	1.9	7.5	0.2	2.4	0.1	

Reading the Sidi-Bel Abbes Region data from CSV file.

In [7]:
```
abbes_df = pd.read_csv('Algerian_forest_fires_dataset_UPDATE.csv',
abbes_df.head()
```

Out[7]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Clas
0	1	6	2012	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8	0.2	no
1	2	6	2012	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9	0.2	no
2	3	6	2012	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6	0.1	no
3	4	6	2012	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6	1	no
4	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9	no

Addition of Region Column

In [8]:
```
abbes_df['Region'] = "Sidi-Bel Abbes"
abbes_df
```

Out[8]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	C
0	1	6	2012	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8	0.2	
1	2	6	2012	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9	0.2	
2	3	6	2012	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6	0.1	
3	4	6	2012	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6	1	
4	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
117	26	9	2012	30	65	14	0.0	85.4	16.0	44.5	4.5	16.9	6.5	
118	27	9	2012	28	87	15	4.4	41.1	6.5	8	0.1	6.2	0	
119	28	9	2012	27	87	29	0.5	45.9	3.5	7.9	0.4	3.4	0.2	
120	29	9	2012	24	54	18	0.1	79.7	4.3	15.2	1.7	5.1	0.7	
121	30	9	2012	24	64	15	0.2	67.3	3.8	16.5	1.2	4.8	0.5	

Merging these two dataframes into one

In [9]:
```
df = abbes_df.append(bejaia_df).reset_index()
df
```

Out[9]:

	index	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI
0	0	1	6	2012	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8
1	1	2	6	2012	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9
2	2	3	6	2012	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6
3	3	4	6	2012	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6
4	4	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5
...	...	...	...	...	...	...	...	...	...	...	...	...	...
239	117	26	9	2012	31	54	11	0.0	82.0	6.0	16.3	2.5	6.2
240	118	27	9	2012	31	66	11	0.0	85.7	8.3	24.9	4.0	9.0
241	119	28	9	2012	32	47	14	0.7	77.5	7.1	8.8	1.8	6.8
242	120	29	9	2012	26	80	16	1.8	47.4	2.9	7.7	0.3	3.0
243	121	30	9	2012	25	78	14	1.4	45.0	1.9	7.5	0.2	2.4

Shape of the dataframe

In [10]:
```
df.shape
```

Out[10]:
```
(244, 16)
```

Information of the dataframe

In [11]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 16 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   index        244 non-null    int64  
 1   day          244 non-null    int64  
 2   month        244 non-null    int64  
 3   year         244 non-null    int64  
 4   Temperature  244 non-null    int64  
 5    RH          244 non-null    int64  
 6    Ws          244 non-null    int64  
 7   Rain         244 non-null    float64
 8   FFMC         244 non-null    float64
 9   DMC          244 non-null    float64
 10  DC           244 non-null    object 
 11  ISI          244 non-null    float64
 12  BUI          244 non-null    float64
 13  FWI          244 non-null    object 
 14  Classes      243 non-null    object 
 15  Region       244 non-null    object 
dtypes: float64(5), int64(7), object(4)
memory usage: 30.6+ KB
```

OBSERVATION : Features DC & FWI are string objects hence we need to convert them into float values.

Feature datatypes

In [12]:

```
df.dtypes
```

Out[12]:

```
index            int64
day              int64
month            int64
year             int64
Temperature      int64
 RH              int64
 Ws              int64
Rain           float64
FFMC           float64
DMC            float64
DC              object
ISI            float64
BUI            float64
FWI             object
Classes         object
Region          object
dtype: object
```

Dropping irrelavant column

```
In [13]: df.drop(columns=['index'], inplace=True)
```

```
In [14]: df.head()
```

Out[14]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Clas
0	1	6	2012	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8	0.2	no
1	2	6	2012	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9	0.2	no
2	3	6	2012	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6	0.1	no
3	4	6	2012	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6	1	no
4	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9	no

Columns in dataframe

```
In [15]: df.columns
```

```
Out[15]: Index(['day', 'month', 'year', 'Temperature', ' RH', ' Ws', 'Rain
         ', 'FFMC',
                'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes  ', 'Region'],
               dtype='object')
```

Checking the null values

In [16]:
```
df.isna().sum()
```

Out[16]:
```
day            0
month          0
year           0
Temperature    0
 RH            0
 Ws            0
Rain           0
FFMC           0
DMC            0
DC             0
ISI            0
BUI            0
FWI            0
Classes        1
Region         0
dtype: int64
```

OBSERVATION : There is one null value present in the Classes feature.

Dropping the row which has missing value

In [17]:
```
df.dropna(inplace=True)
df.isna().sum()
```

Out[17]:
```
day            0
month          0
year           0
Temperature    0
 RH            0
 Ws            0
Rain           0
FFMC           0
DMC            0
DC             0
ISI            0
BUI            0
FWI            0
Classes        0
Region         0
dtype: int64
```

Checking the shape of the df after dropping the row

In [18]:
```
df.shape
```

Out[18]:
```
(243, 15)
```

Changing the name of features

In [19]:
```
df.rename(columns={'Classes  ': "Classes", 'Rain ': "Rain", " Ws":
```

Creation of date column from [day, month, year] features

In [20]:
```
df['date']= df.loc[:, 'day':'year'].apply(lambda x:"/".join(x.astyp
```

In [21]:
```
df
```

Out[21]:

	day	month	year	Temperature	Relative Humidity	Wind-Speed	Rain	FFMC	DMC	DC	ISI	BUI
0	1	6	2012	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8
1	2	6	2012	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9
2	3	6	2012	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6
3	4	6	2012	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6
4	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5
...	...	...	...	...	...	...	...	...	...	...	...	...
239	26	9	2012	31	54	11	0.0	82.0	6.0	16.3	2.5	6.2
240	27	9	2012	31	66	11	0.0	85.7	8.3	24.9	4.0	9.0
241	28	9	2012	32	47	14	0.7	77.5	7.1	8.8	1.8	6.8
242	29	9	2012	26	80	16	1.8	47.4	2.9	7.7	0.3	3.0
243	30	9	2012	25	78	14	1.4	45.0	1.9	7.5	0.2	2.4

Dropping columns [day, month, year] from dataset

In [22]:
```
df.drop(columns=['day', 'month', 'year'], inplace=True)
```

Converting date string to datetime object

In [23]:
```
df['date'] = pd.to_datetime(df['date'])
```

Converting the type for Feature DC

In [24]:
```
df["DC"] = df['DC'].map(lambda x: str(x).replace(" ", "")).astype('
```

Converting the type for Feature FWI

In [25]:
```
df['FWI'] = df['FWI'].astype('float')
```

Checking Values for Classes Feature

In [26]:
```
df['Classes'] = df['Classes'].str.strip()
```

In [27]:
```
df["Classes"].value_counts()
```

Out[27]:
```
fire        137
not fire    106
Name: Classes, dtype: int64
```

```
In [28]: df
```

Out[28]:

	Temperature	Relative Humidity	Wind-Speed	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Reg
0	32	71	12	0.7	57.1	2.5	8.2	0.6	2.8	0.2	not fire	S Ab
1	30	73	13	4.0	55.7	2.7	7.8	0.6	2.9	0.2	not fire	S Ab
2	29	80	14	2.0	48.7	2.2	7.6	0.3	2.6	0.1	not fire	S Ab
3	30	64	14	0.0	79.4	5.2	15.4	2.2	5.6	1.0	not fire	S Ab
4	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9	not fire	S Ab
...	...	...	...	...	...	...	...	...	...	...	...	
239	31	54	11	0.0	82.0	6.0	16.3	2.5	6.2	1.7	not fire	Be
240	31	66	11	0.0	85.7	8.3	24.9	4.0	9.0	4.1	fire	Be
241	32	47	14	0.7	77.5	7.1	8.8	1.8	6.8	0.9	not fire	Be
242	26	80	16	1.8	47.4	2.9	7.7	0.3	3.0	0.1	not fire	Be
243	25	78	14	1.4	45.0	1.9	7.5	0.2	2.4	0.1	not fire	Be

Statistical Summary of the data

In [29]:
```
df.describe()
```

Out[29]:

	Temperature	Relative Humidity	Wind-Speed	Rain	FFMC	DMC	D
count	243.000000	243.000000	243.000000	243.000000	243.000000	243.000000	243.00000
mean	32.152263	62.041152	15.493827	0.762963	77.842387	14.680658	49.4308
std	3.628039	14.828160	2.811385	2.003207	14.349641	12.393040	47.6656
min	22.000000	21.000000	6.000000	0.000000	28.600000	0.700000	6.90000
25%	30.000000	52.500000	14.000000	0.000000	71.850000	5.800000	12.35000
50%	32.000000	63.000000	15.000000	0.000000	83.300000	11.300000	33.10000
75%	35.000000	73.500000	17.000000	0.500000	88.300000	20.800000	69.10000
max	42.000000	90.000000	29.000000	16.800000	96.000000	65.900000	220.40000

Making Date as index for the dataframe

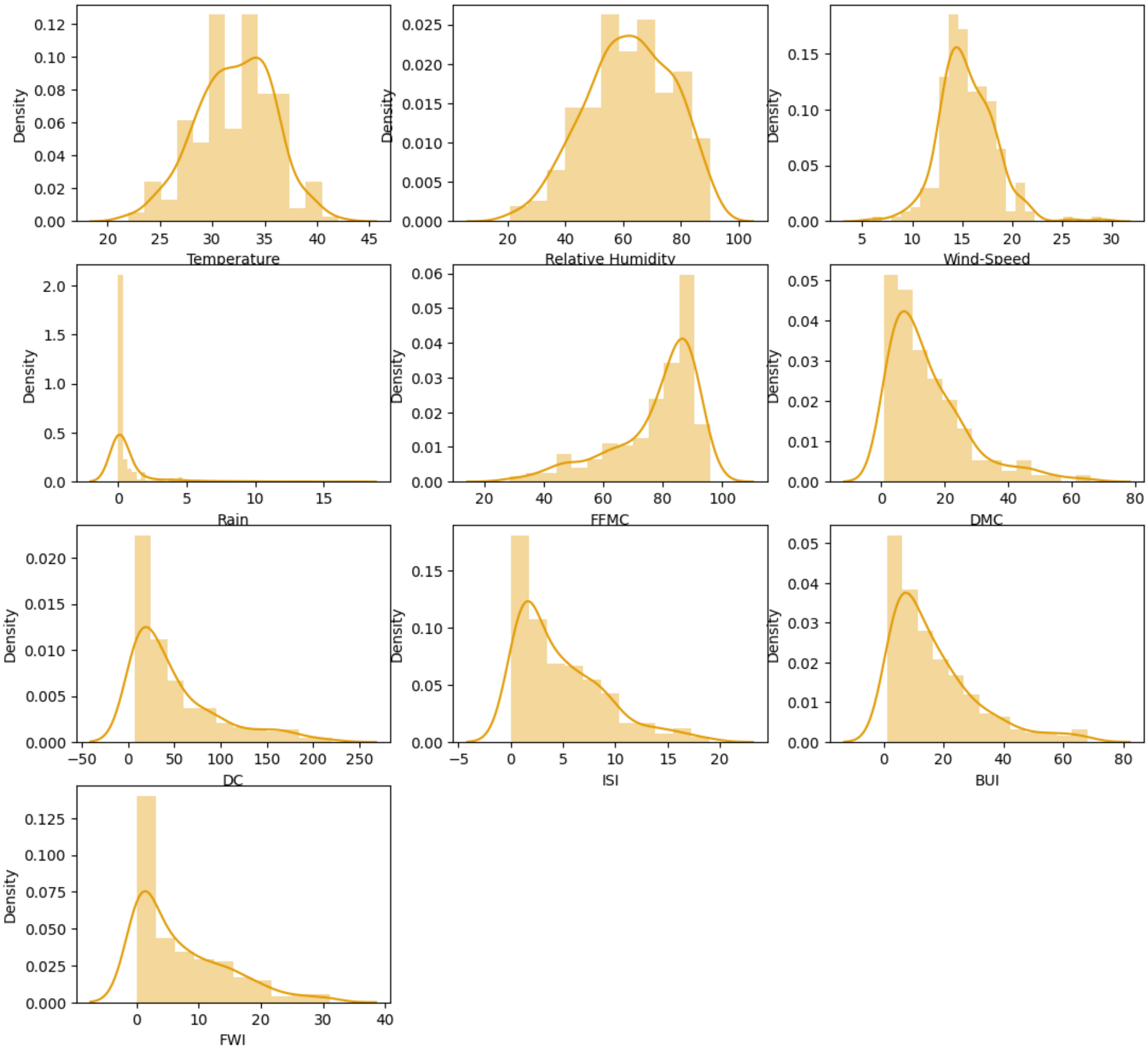
In [30]:
```
df.set_index('date', inplace=True)
```

Univariate Analysis

Distribution plot for all the numerical Values

In [31]:
```
feature_filter = df.columns[(df.dtypes=="int64") | (df.dtypes=='flo
```

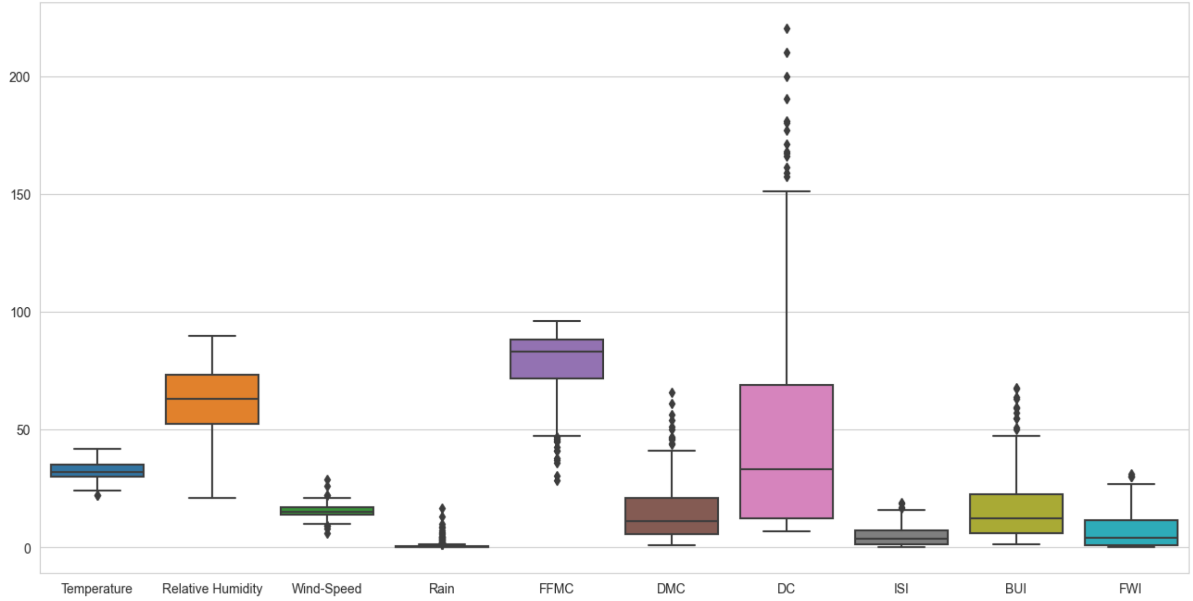
In [32]:
```
pos=1
fig = plt.figure(figsize=(13,25))

for i in feature_filter:
    ax = fig.add_subplot(8,3,pos)
    pos = pos + 1
    sns.distplot(df[i], ax=ax, color="#e39d07")
```



Box plot distribution for all the numeric features

In [33]:
```
sns.set_style("whitegrid")
plt.figure(figsize=(16, 8))
sns.boxplot(data=df)
```

Out[33]:
```
<AxesSubplot:>
```



Count Matrix of Region wrt Classes

In [34]:
```
pd.crosstab(df['Classes'], df['Region'], normalize=True)*100
```

Out[34]:

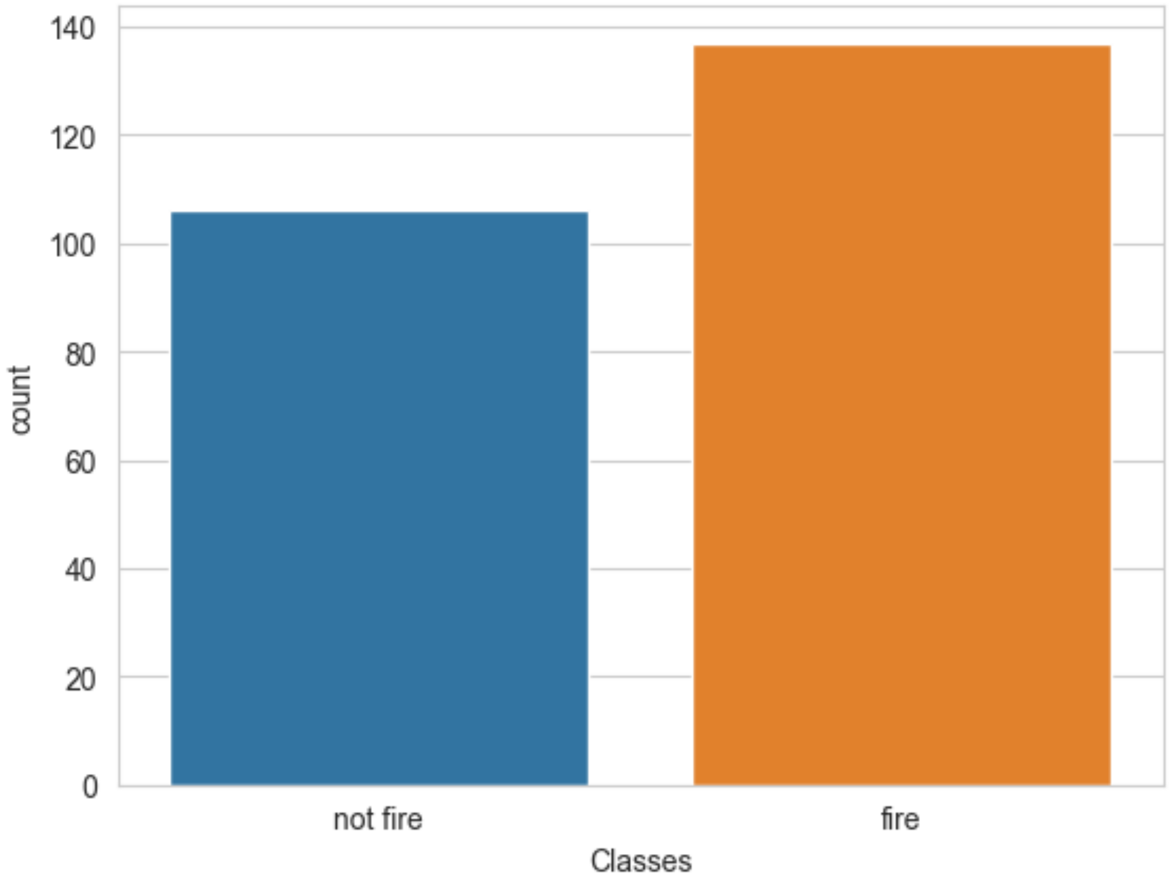
Region	Bejaia	Sidi-Bel Abbes
Classes		
fire	24.279835	32.098765
not fire	25.925926	17.695473

OBSERVATION : There are more fire incidents happened in Sidi-Bel Abbes region as compared to Bejaia region.

Count plot for Classes in dataset

In [35]:
```
sns.countplot(data=df, x="Classes")
```

Out[35]:
```
<AxesSubplot:xlabel='Classes', ylabel='count'>
```



Bivariate Analysis

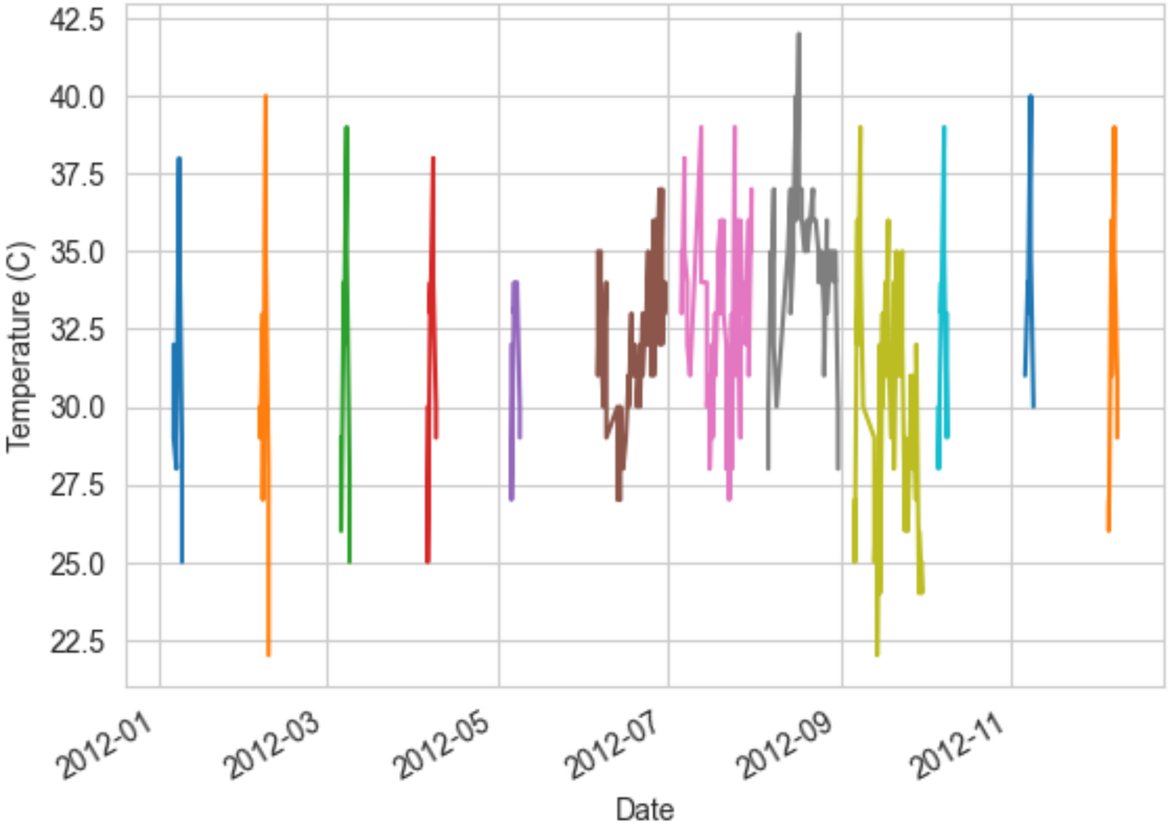
In [36]:
```
df.columns
```

Out[36]:
```
Index(['Temperature', 'Relative Humidity', 'Wind-Speed', 'Rain', '
FFMC', 'DMC',
       'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'Region'],
      dtype='object')
```

Plotting Temperature wrt monthly data

In [37]:
```
temp = df['Temperature'].resample('M')
temp.plot(ylabel = "Temperature (C)", xlabel="Date")
```

Out[37]:
```
date
2012-01-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-02-29    AxesSubplot(0.125,0.2;0.775x0.68)
2012-03-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-04-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-05-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-06-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-07-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-08-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-09-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-10-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-11-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-12-31    AxesSubplot(0.125,0.2;0.775x0.68)
Freq: M, Name: Temperature, dtype: object
```

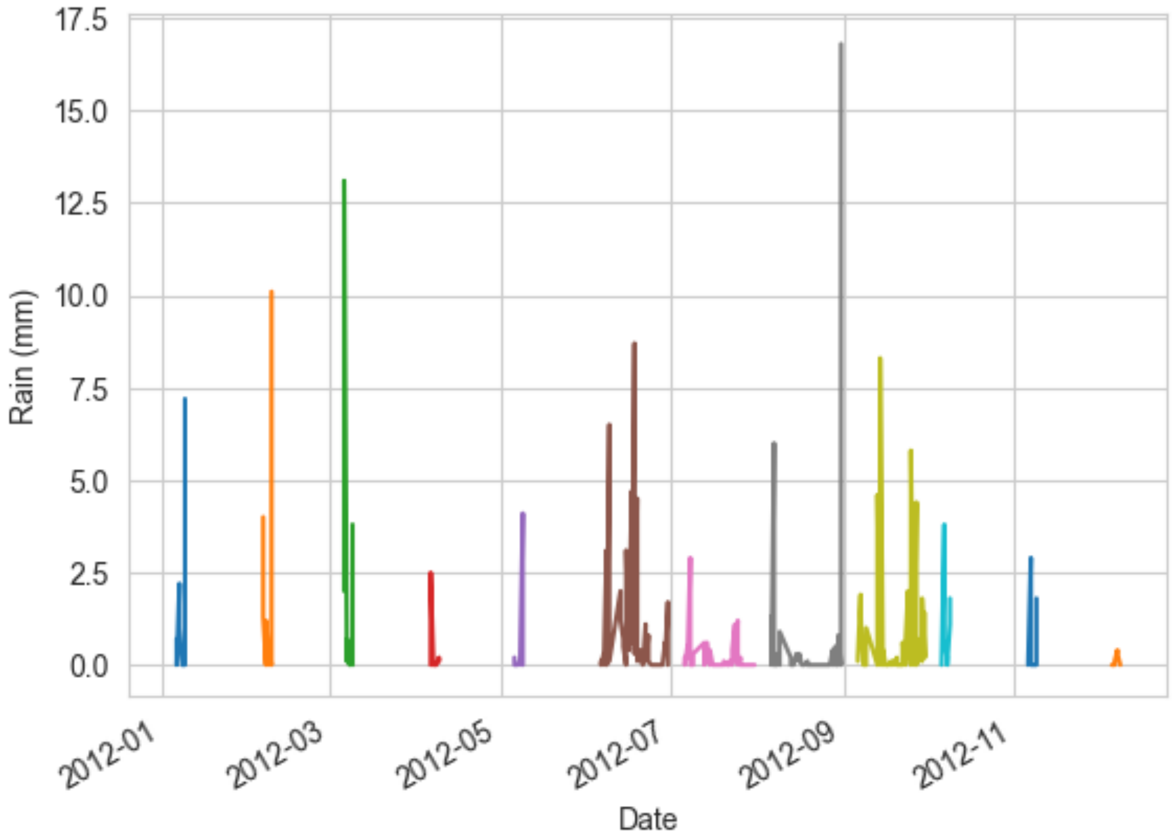


OBSERVATION : As evident from the above graph, high temprature signs are observed between the month of July and October.

Plotting Rain wrt Monthly data

```
In [38]: rain = df['Rain'].resample('M')
         rain.plot(ylabel = "Rain (mm)", xlabel="Date")
```

```
Out[38]: date
         2012-01-31    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-02-29    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-03-31    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-04-30    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-05-31    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-06-30    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-07-31    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-08-31    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-09-30    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-10-31    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-11-30    AxesSubplot(0.125,0.2;0.775x0.68)
         2012-12-31    AxesSubplot(0.125,0.2;0.775x0.68)
         Freq: M, Name: Rain, dtype: object
```

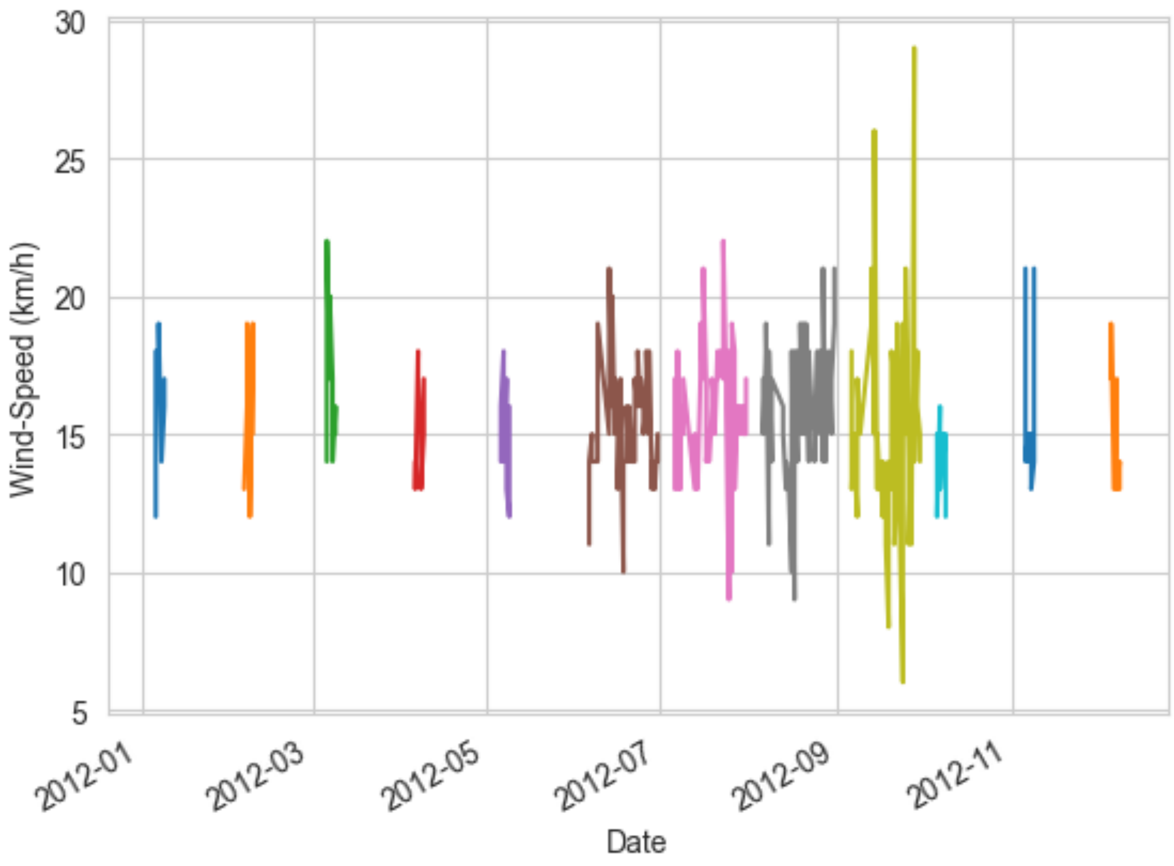


OBSERVATION : High rainfall months were June & September.

Wind-Speed wrt monthly data

In [39]:
```
temp = df['Wind-Speed'].resample('M')
temp.plot(ylabel = "Wind-Speed (km/h)", xlabel="Date")
```

Out[39]:
```
date
2012-01-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-02-29    AxesSubplot(0.125,0.2;0.775x0.68)
2012-03-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-04-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-05-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-06-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-07-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-08-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-09-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-10-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-11-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-12-31    AxesSubplot(0.125,0.2;0.775x0.68)
Freq: M, Name: Wind-Speed, dtype: object
```

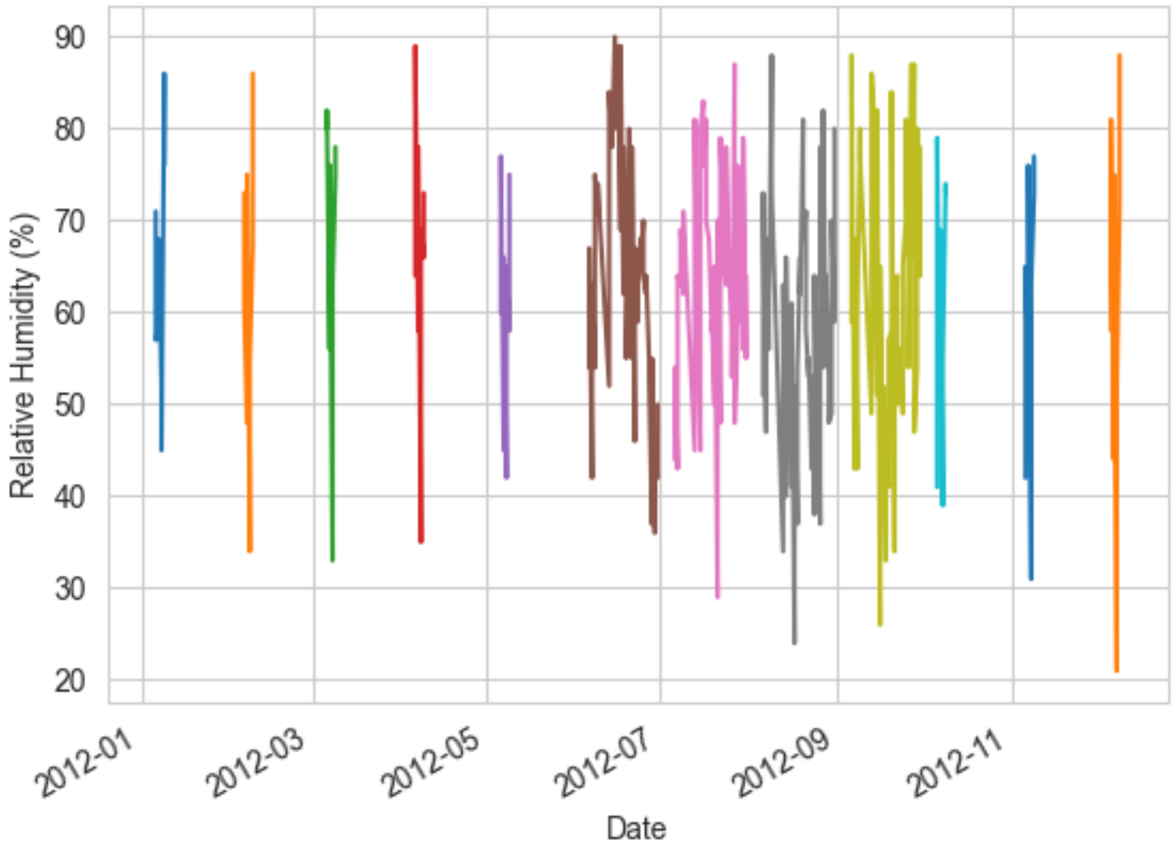


OBSERVATION : Windspeed was high during the month of October.

Relative Humidity wrt monthly data

In [40]:
```
temp = df['Relative Humidity'].resample('M')
temp.plot(ylabel = "Relative Humidity (%)", xlabel="Date")
```

Out[40]:
```
date
2012-01-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-02-29    AxesSubplot(0.125,0.2;0.775x0.68)
2012-03-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-04-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-05-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-06-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-07-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-08-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-09-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-10-31    AxesSubplot(0.125,0.2;0.775x0.68)
2012-11-30    AxesSubplot(0.125,0.2;0.775x0.68)
2012-12-31    AxesSubplot(0.125,0.2;0.775x0.68)
Freq: M, Name: Relative Humidity, dtype: object
```



OBSERVATION : Humidity seems high in the months of August, September & October

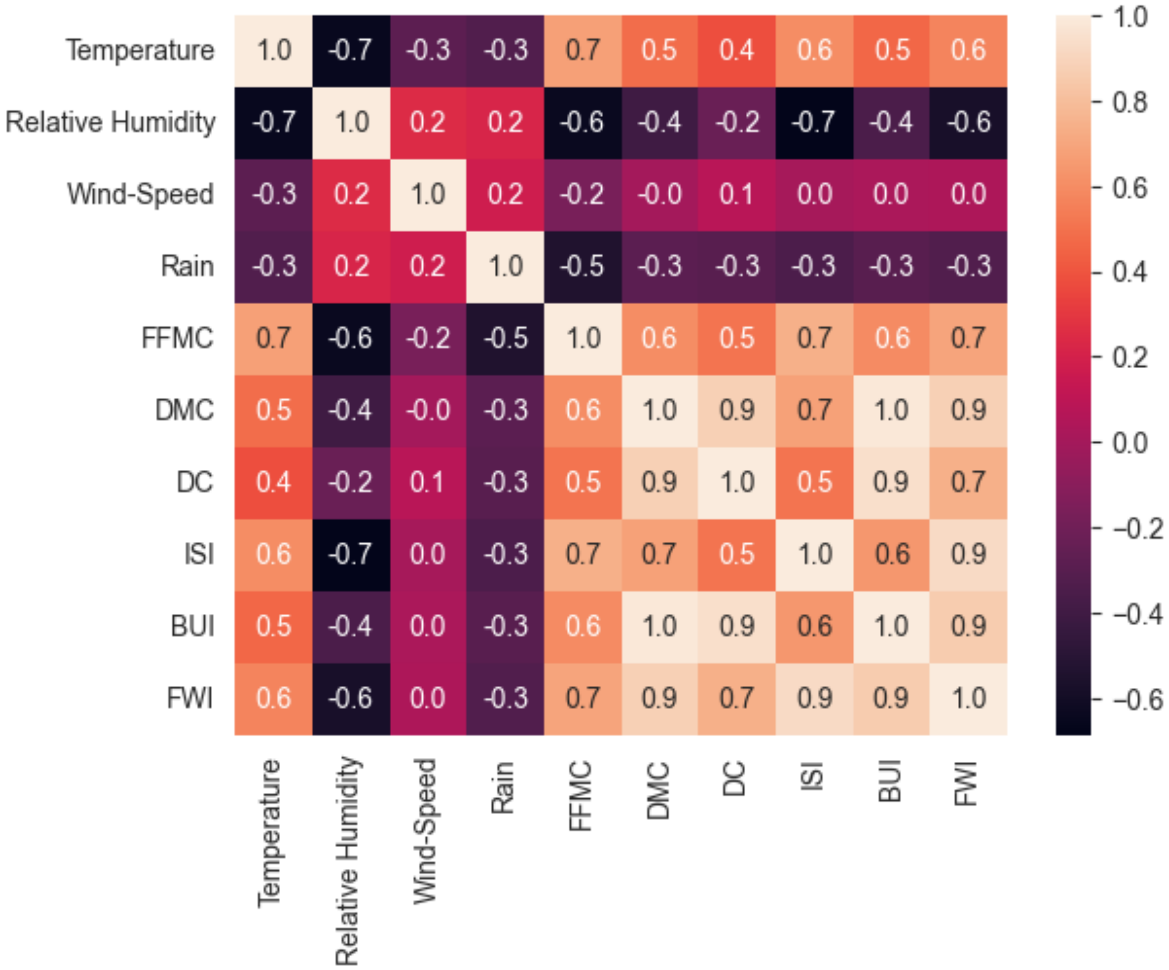
Resetting the index

In [41]:
```
df.reset_index(inplace=True)
```

Correlation Heatmap

```
In [42]: sns.heatmap(df.corr(), annot=True,fmt='.1f')
```

```
Out[42]: <AxesSubplot:>
```

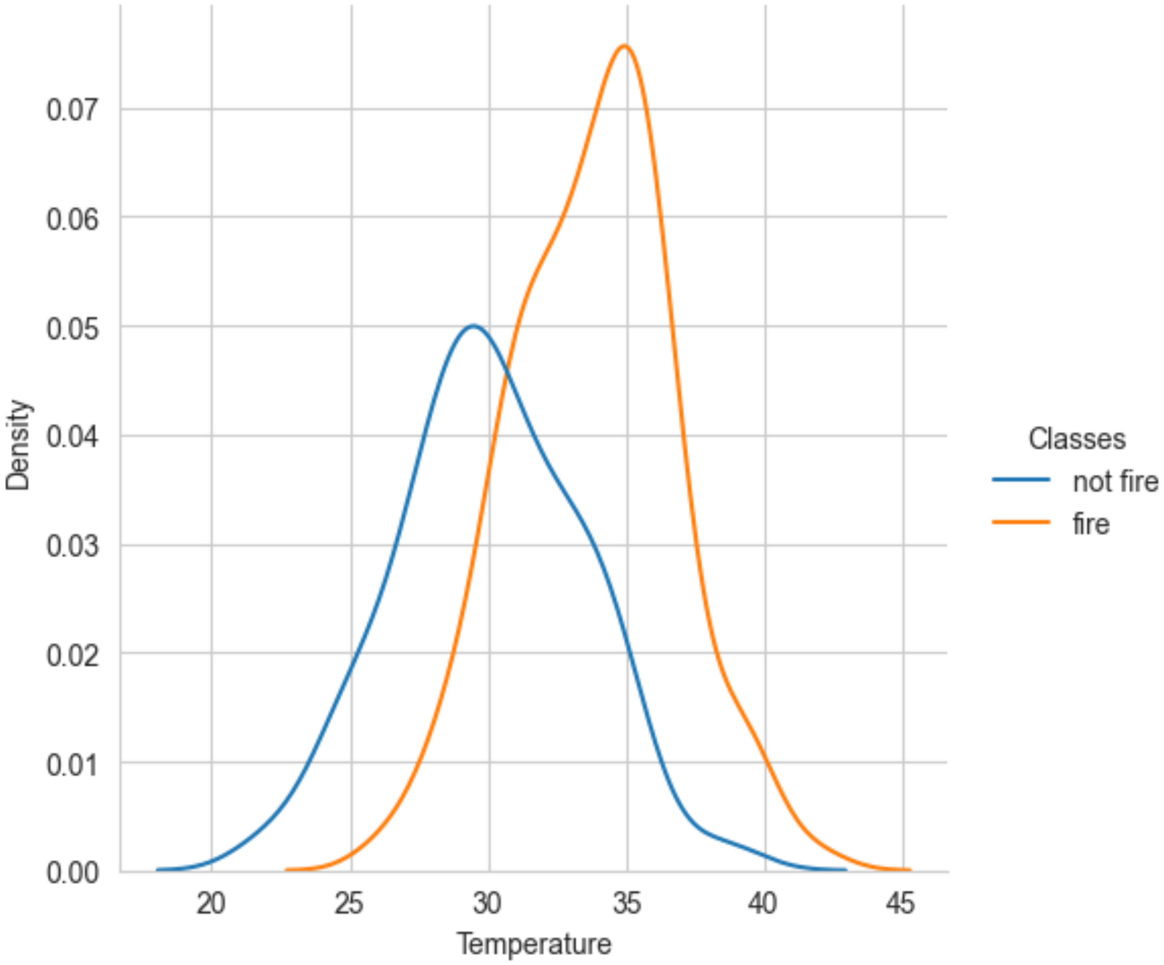


OBSERVATION : DC & DMC, FWI & DMC , BUI & DC , FWI & ISI AND FWI & BUI are highly correlated.

Temperature Distribution

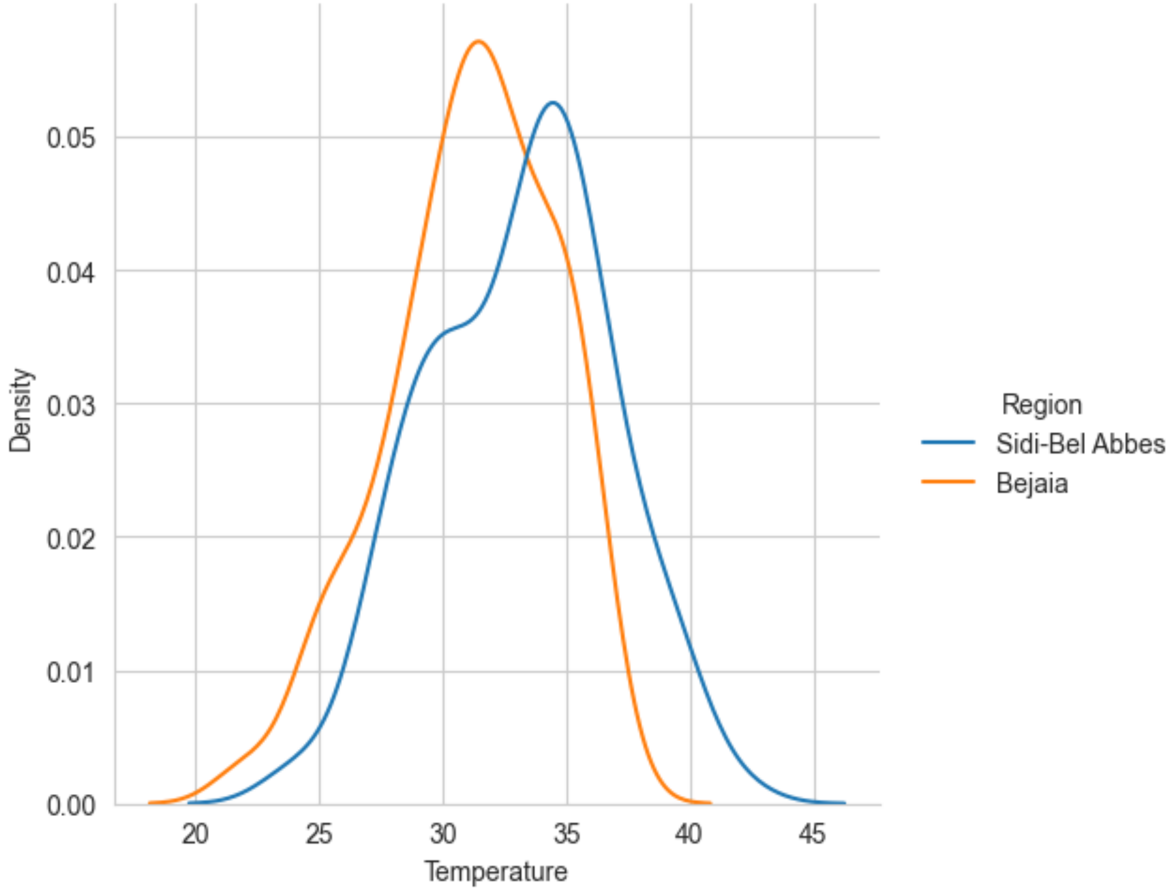
```
In [43]: sns.set_style("whitegrid")
         sns.displot(data=df, x='Temperature', hue="Classes", kind="kde")
```

```
Out[43]: <seaborn.axisgrid.FacetGrid at 0x175214220>
```



In [44]:
```
sns.set_style("whitegrid")
sns.displot(data=df, x='Temperature', hue="Region", kind="kde")
```

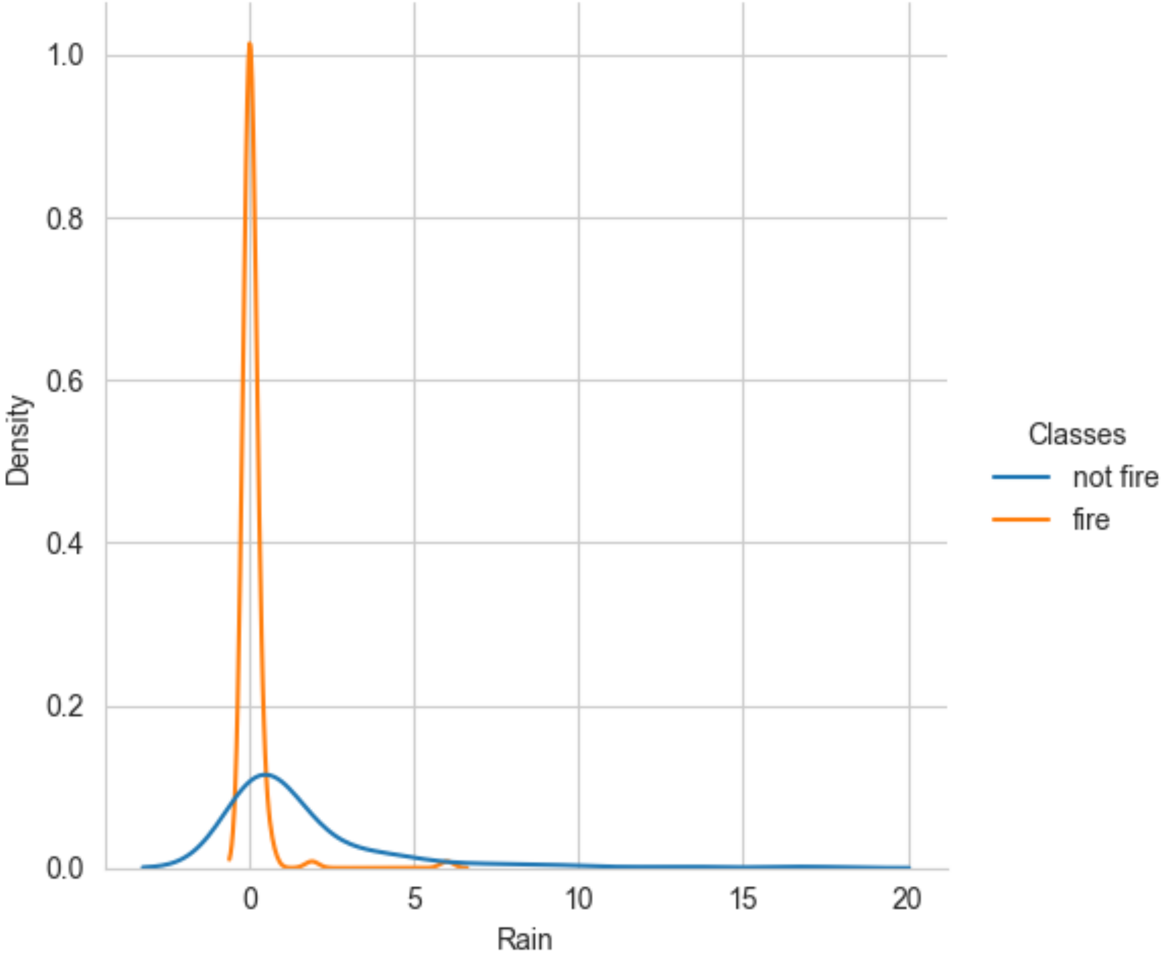
Out[44]:
```
<seaborn.axisgrid.FacetGrid at 0x17514b3d0>
```



Rainfall Distribution

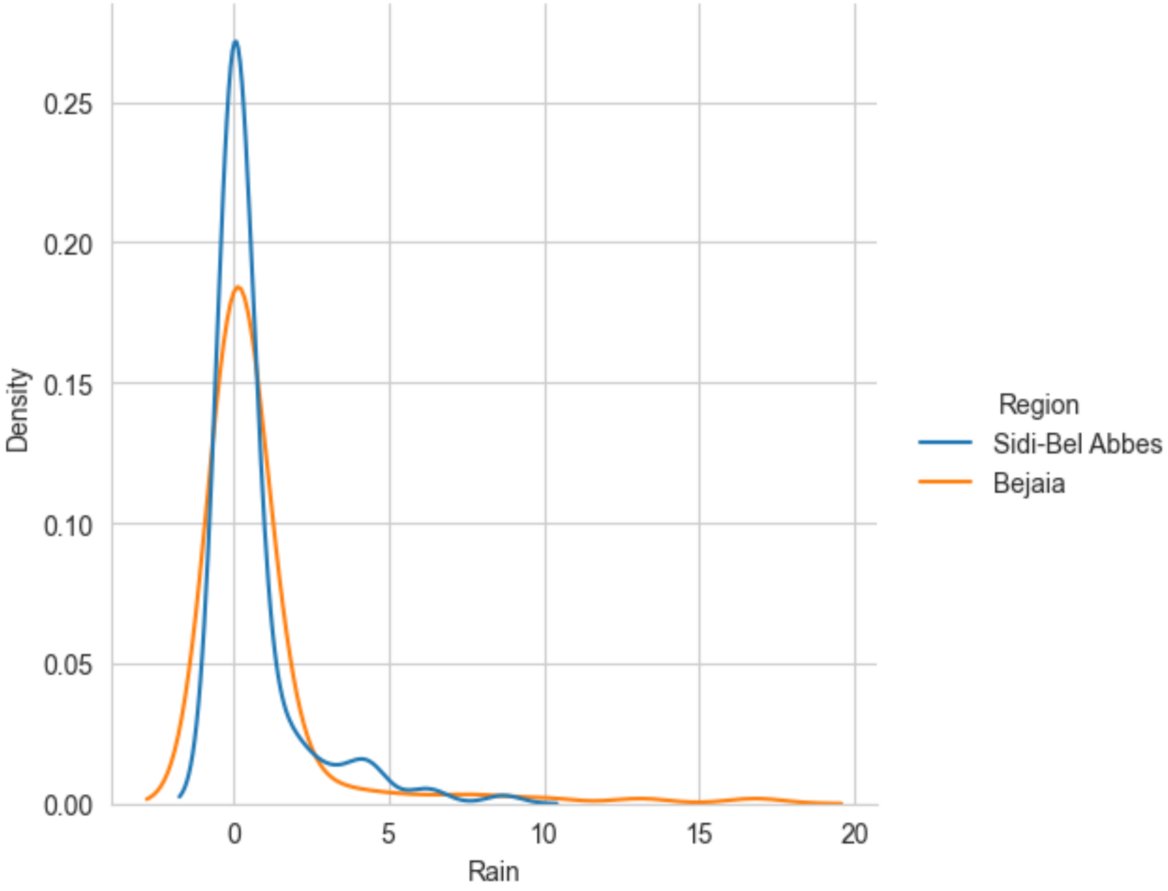
```
In [45]: sns.set_style("whitegrid")
         sns.displot(data=df, x='Rain', hue="Classes", kind="kde")
```

```
Out[45]: <seaborn.axisgrid.FacetGrid at 0x17545b280>
```



```
In [46]: sns.set_style("whitegrid")
         sns.displot(data=df, x='Rain', hue="Region", kind="kde")
```

```
Out[46]: <seaborn.axisgrid.FacetGrid at 0x17549ca60>
```

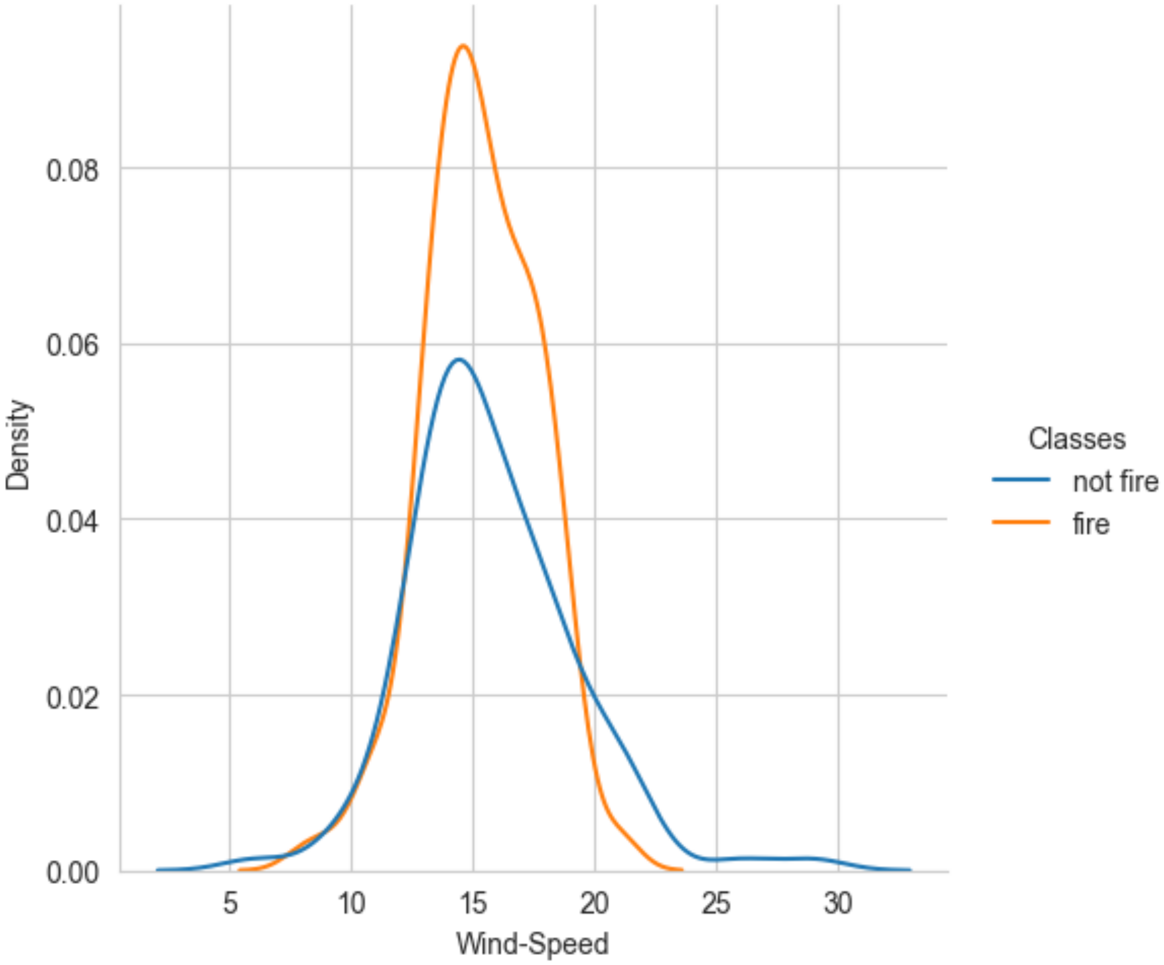


OBSERVATION : Fire incident happened in the low rainfall regions mostly.

Wind-Speed Distribution

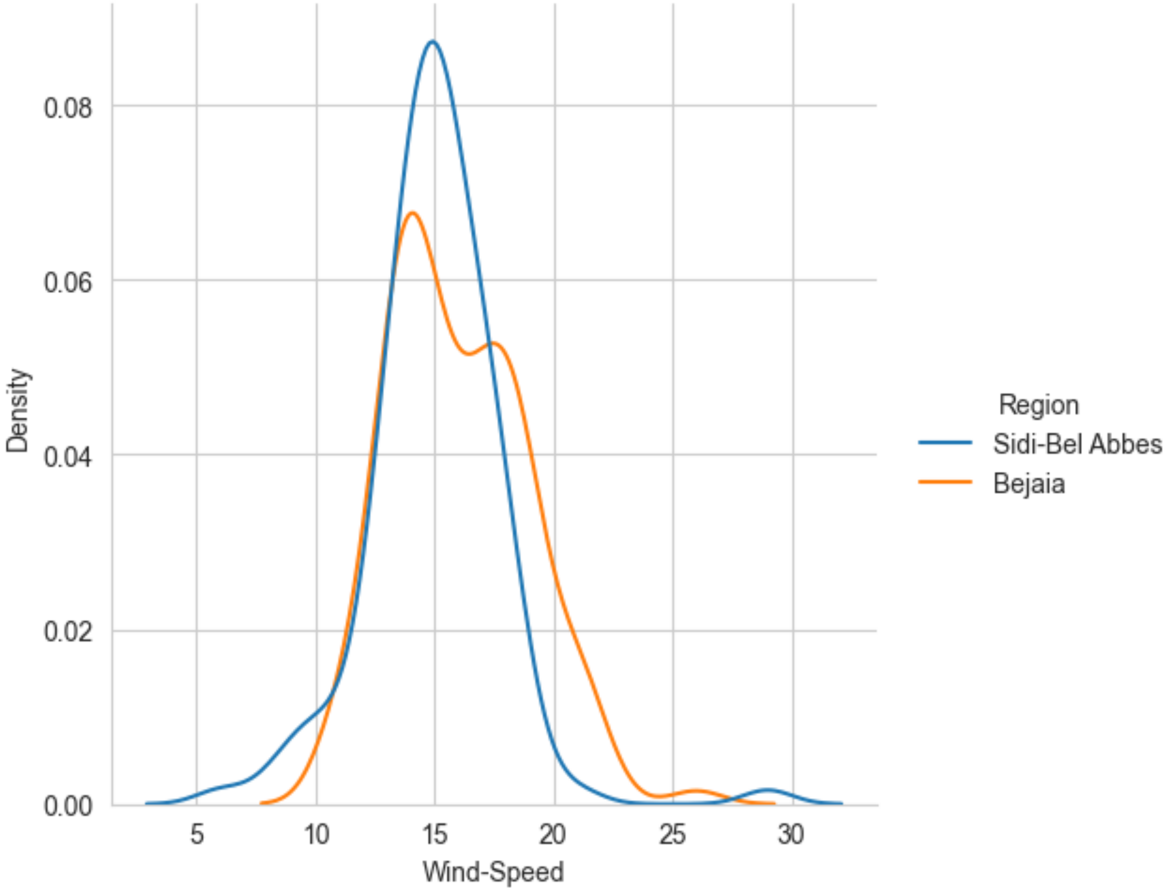
In [47]:
```
sns.set_style("whitegrid")
sns.displot(data=df, x='Wind-Speed', hue='Classes', kind="kde")
```

Out[47]:
```
<seaborn.axisgrid.FacetGrid at 0x17541c250>
```



In [48]:
```
sns.set_style("whitegrid")
sns.displot(data=df, x='Wind-Speed', hue="Region", kind="kde")
```

Out[48]:
```
<seaborn.axisgrid.FacetGrid at 0x17554eb00>
```

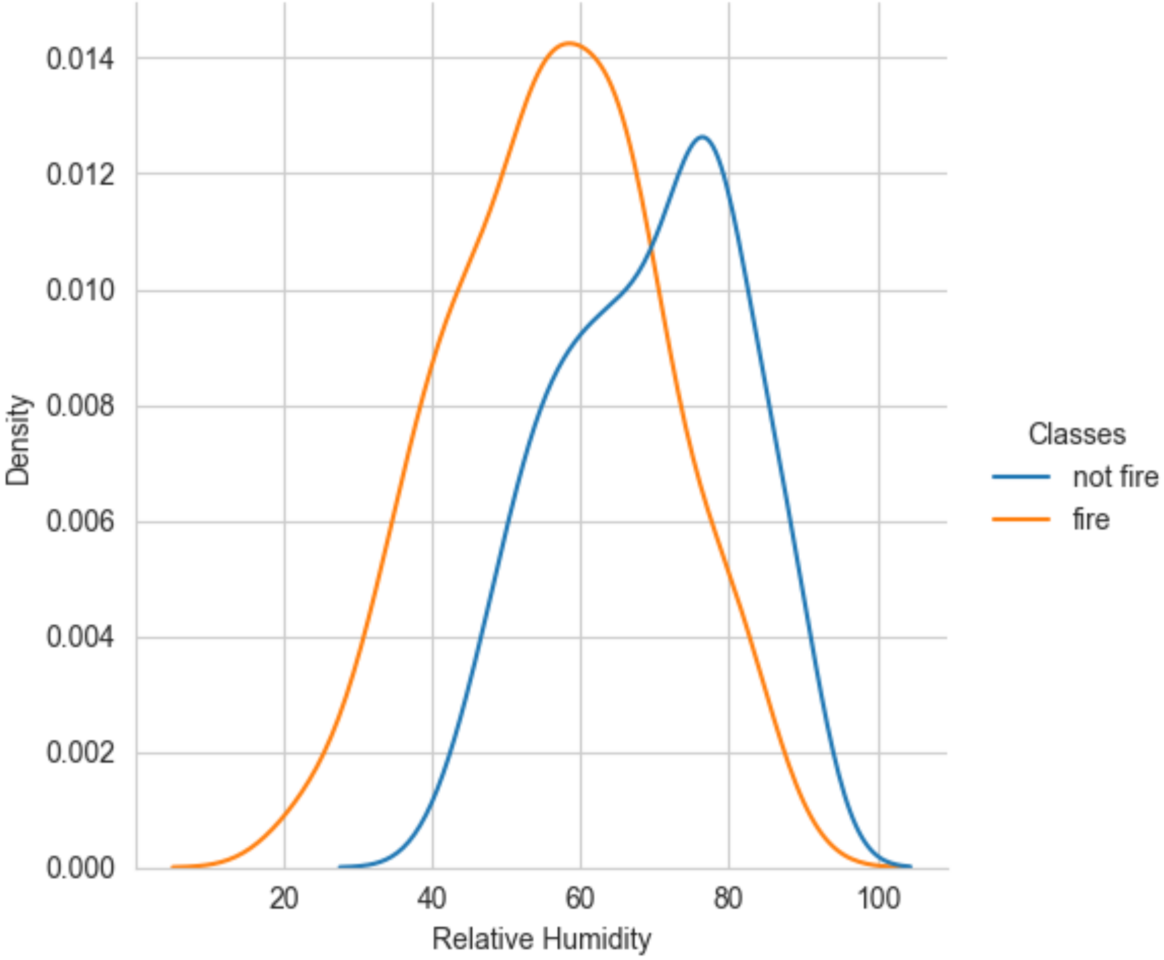


OBSERVATION : Fire incident happened in the low rainfall regions mostly.

Relative Humidity Distribution

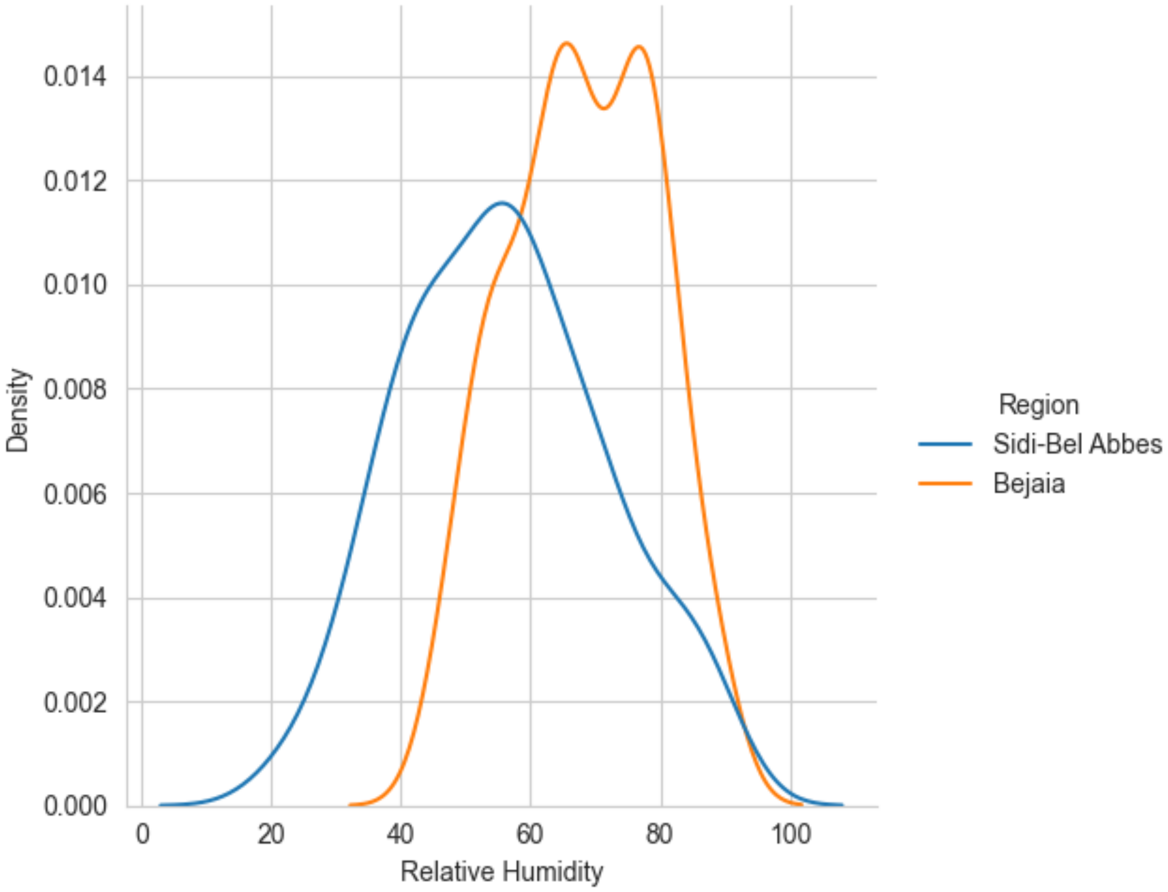
```
In [49]: sns.set_style("whitegrid")
         sns.displot(data=df, x='Relative Humidity', hue="Classes", kind="kd
```

```
Out[49]: <seaborn.axisgrid.FacetGrid at 0x17550ace0>
```



In [50]:
```
sns.set_style("whitegrid")
sns.displot(data=df, x='Relative Humidity', hue="Region", kind="kde
```

Out[50]:
```
<seaborn.axisgrid.FacetGrid at 0x1755cac50>
```



OBSERVATION : Relative Humidity is generally low in the observations where fire incidents happend.

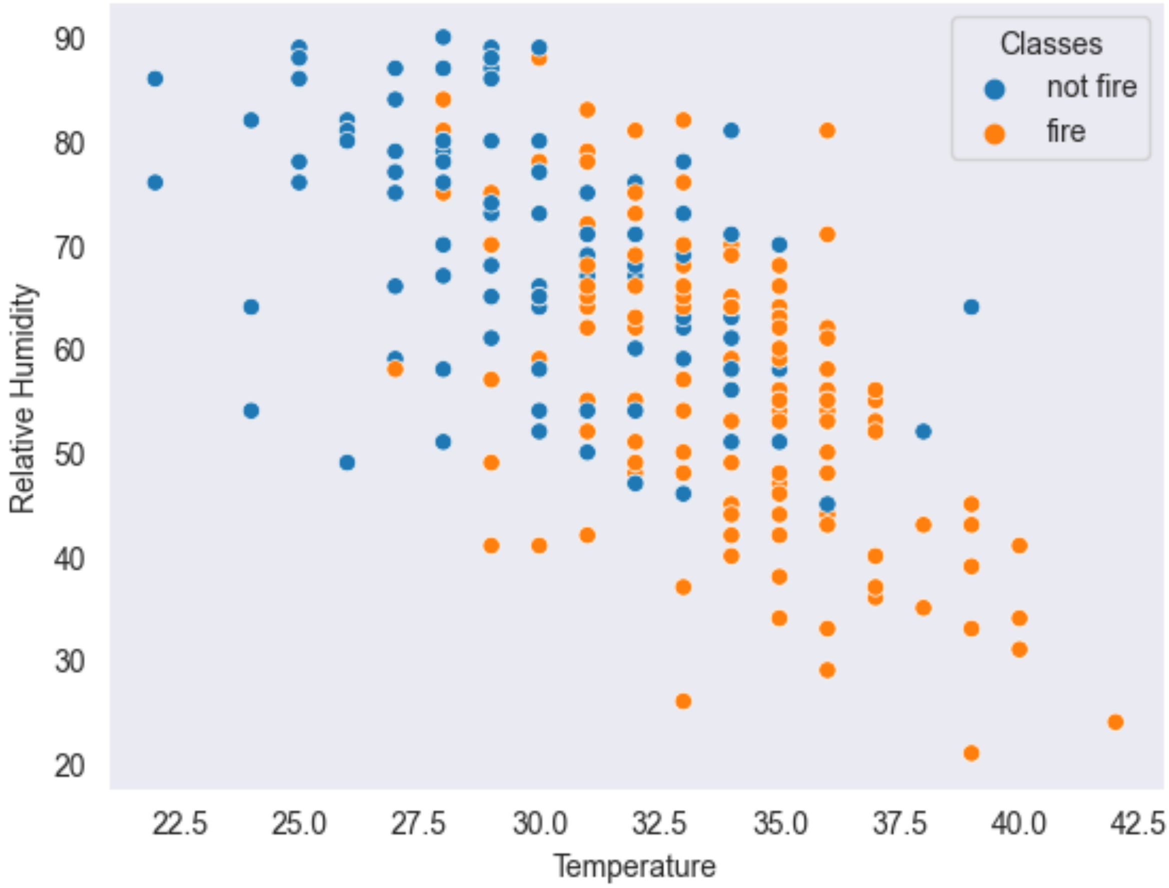
In [51]:
```
df.columns
```

Out[51]:
```
Index(['date', 'Temperature', 'Relative Humidity', 'Wind-Speed', '
Rain',
       'FFMC', 'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'Regio
n'],
      dtype='object')
```

Scatter Plot between Temperature & Relative Humidity

```
In [52]: sns.set_style('dark')
         sns.scatterplot(data=df, x='Temperature', y='Relative Humidity', hu
```

```
Out[52]: <AxesSubplot:xlabel='Temperature', ylabel='Relative Humidity'>
```

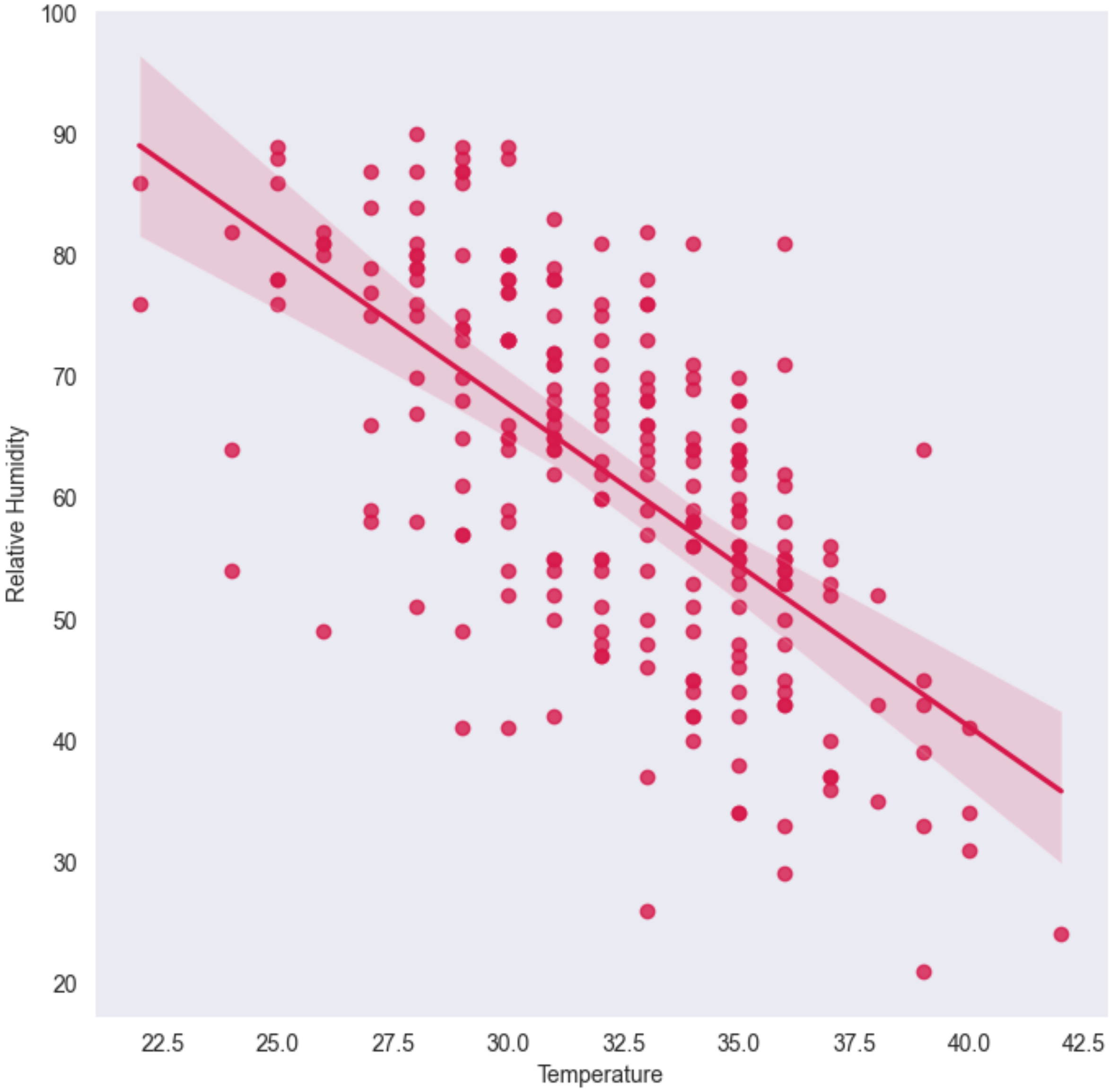


In [53]:

```
plt.figure(figsize=(8, 8))
sns.regplot(data=df, x='Temperature', y="Relative Humidity", ci=100
```

Out[53]:

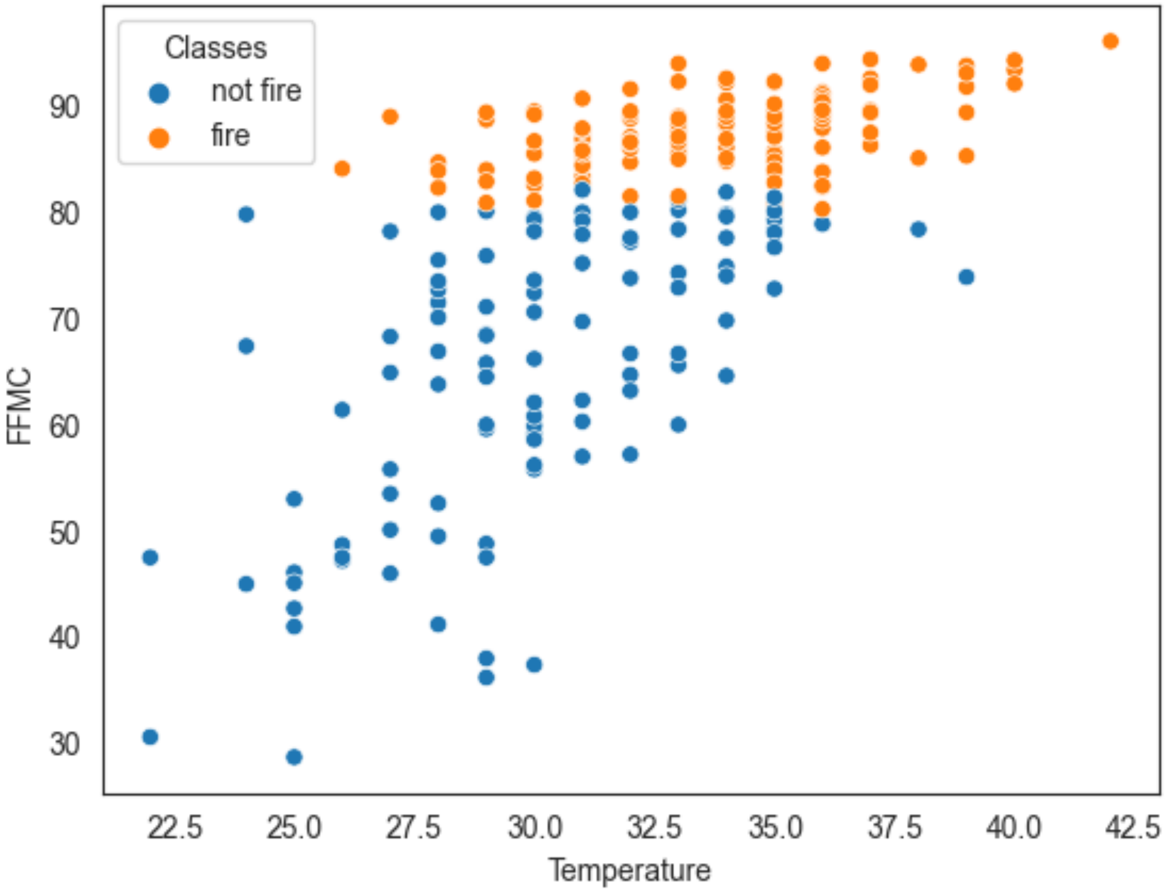
```
<AxesSubplot:xlabel='Temperature', ylabel='Relative Humidity'>
```



OBSERVATION : There is a negative correlation between Temperature & Relative Humidity.

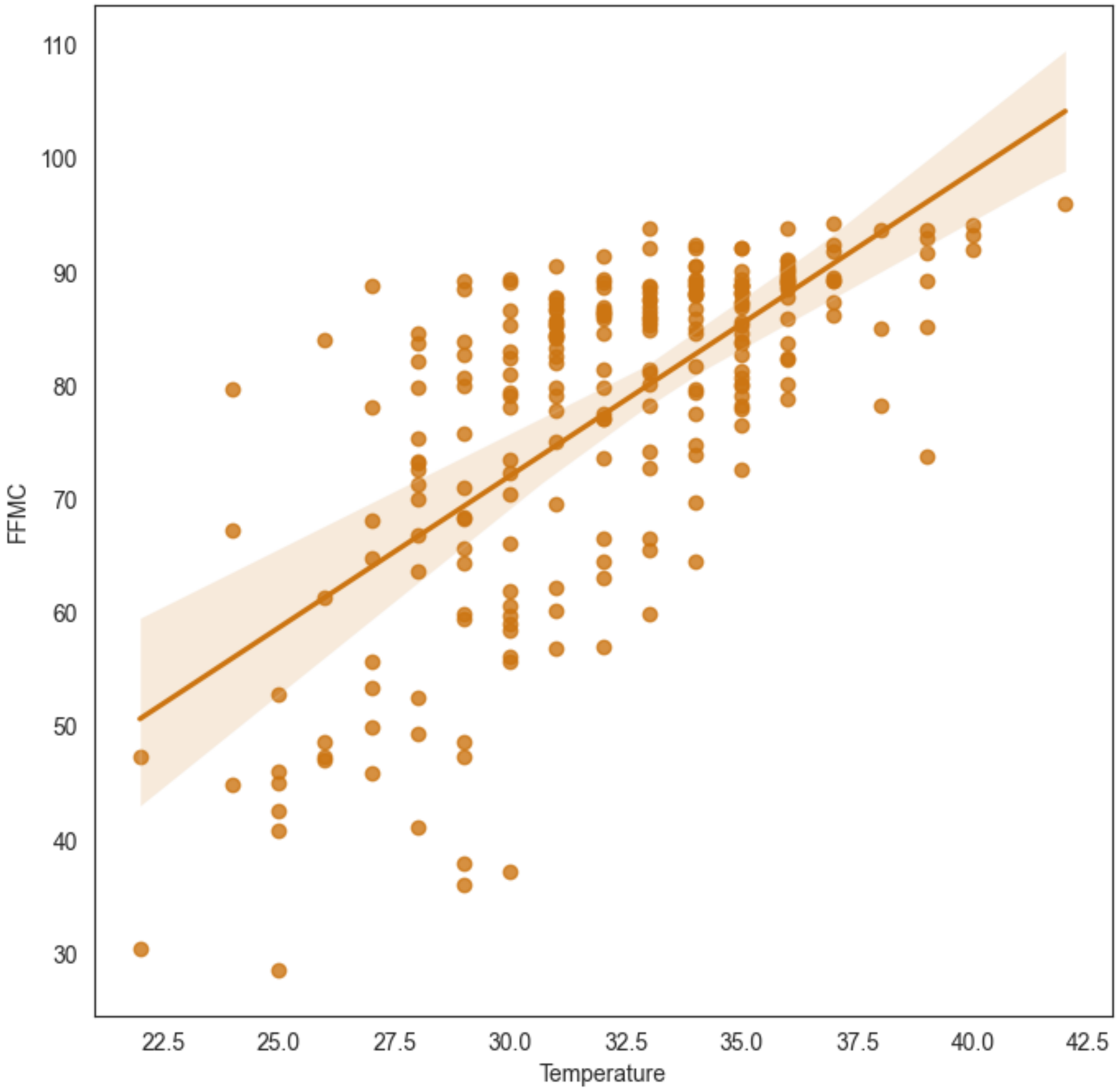
Scatter Plot between FFMC & Temperature

```
In [54]: sns.set_style('white')
         sns.scatterplot(data=df, x='Temperature', y='FFMC', hue="Classes")

Out[54]: <AxesSubplot:xlabel='Temperature', ylabel='FFMC'>
```



```
In [55]: plt.figure(figsize=(8, 8))
         sns.regplot(data=df, x='Temperature', y="FFMC", ci=100, color="#cc7
```

```
Out[55]: <AxesSubplot:xlabel='Temperature', ylabel='FFMC'>
```



OBSERVATION : Observations with Fire class have higher FFMC values.

```
In [ ]:
```