EDA and Feature Engineering with Algerian_Forest_Fires_Dataset

```
Life Cycle of Machine Learning Project

* Understanding the Problem Statement
* Data Collection
* Exploratory data analysis
* Data Cleaning
* Data Pre-Processing
* Model Training
* Choose Best Model
```

```
1. Problem Statements

* The dataset includes data of two regions of Algerian, namely the Bajaia region located in the
  northest of Algerian and the Sidi Bel-abbes region locatedin the northwest of Algeria.
* User need to draw / give assumption/prediction that Algeria Forest will catch up fire or not
  based on input data.
* Presdiction result can be used to take correective  action or prevent the mishap.
```

```
2. Data Collection
 * This data set collection from UCI Machine Learning Repository(website)
 * The dataset includes columns and 244 row
```

In [178]:
```
# importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
%matplotlib inline
warnings.filterwarnings('ignore')
```

In [25]:
```
# upload the datasets
df=pd.read_csv(r"C:\Users\Mukul\Data Kaggle\Downloads\Datasets\Algerian_forest_fi
```

In [26]:
```
df.head()
```

Out[26]:

	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

In [27]:
```
df.tail()
```

Out[27]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes
241	26	09	2012	30	65	14	0	85.4	16	44.5	4.5	16.9	6.5	fire
242	27	09	2012	28	87	15	4.4	41.1	6.5	8	0.1	6.2	0	not fire
243	28	09	2012	27	87	29	0.5	45.9	3.5	7.9	0.4	3.4	0.2	not fire
244	29	09	2012	24	54	18	0.1	79.7	4.3	15.2	1.7	5.1	0.7	not fire
245	30	09	2012	24	64	15	0.2	67.3	3.8	16.5	1.2	4.8	0.5	not fire

DATA Cleaning

In [37]:
```
#Removing unnecessary Rows from datasets
```

In [43]:
```
df.drop(index=[122,123],inplace=True)
df.reset_index(inplace=True)
df.drop('index',axis=1 , inplace=True)
```

Adding new feature , named, Region in a dataset

In [44]:
```
df.loc[:122,'region']='bejaia'
df.loc[122:, 'region']='Sidi-Bel Abbes'
```

stripping the name of the columns

In [45]:
```
df.columns = [i.strip() for i in df.columns]
df.columns
```

Out[45]:
```
Index(['day', 'month', 'year', 'Temperature', 'RH', 'Ws', 'Rain', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes', 'region'],
      dtype='object')
```

In [46]:
```
# stripping the Classes Features data
df.Classes = df.Classes.str.strip()
df['Classes'].unique()
```

Out[46]:
```
array(['not fire', 'fire', nan], dtype=object)
```

In [47]:
```
df.head()
```

Out[47]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	reg
0	01	06	2012	29	57	18	0	65.7	3.4	7.6	1.3	3.4	0.5	not fire	be
1	02	06	2012	29	61	13	1.3	64.4	4.1	7.6	1	3.9	0.4	not fire	be
2	03	06	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire	be
3	04	06	2012	25	89	13	2.5	28.6	1.3	6.9	0	1.7	0	not fire	be
4	05	06	2012	27	77	16	0	64.8	3	14.2	1.2	3.9	0.5	not fire	be

Changing the Data Types of the Columns

In [64]:
```
df['day'] = df['day'].astype(int)
df['month']=df['month'].astype(int)
df['year']=df['year'].astype(int)
df['Temperature'] = df['Temperature'].astype(int)
df['RH'] =df['RH'].astype(int)
df['Rain']=df['Rain'].astype(float)
df['FFMC'] = df['FFMC'].astype(float)
df['DMC'] = df['DMC'].astype(float)
df['BUI'] = df['BUI'].astype(float)
df['ISI'] = df['ISI'].astype(float)
df['Ws'] = df['Ws'].astype(float)
```

In [65]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   day          244 non-null    int32  
 1   month        244 non-null    int32  
 2   year         244 non-null    int32  
 3   Temperature  244 non-null    int32  
 4   RH           244 non-null    int32  
 5   Ws           244 non-null    float64
 6   Rain         244 non-null    float64
 7   FFMC         244 non-null    float64
 8   DMC          244 non-null    float64
 9   DC           244 non-null    object 
 10  ISI          244 non-null    float64
 11  BUI          244 non-null    float64
 12  FWI          244 non-null    object 
 13  Classes      243 non-null    object 
 14  region       244 non-null    object 
dtypes: float64(6), int32(5), object(4)
memory usage: 24.0+ KB
```

Adding New Feature , named , 'Date'by Replacing Unnecessary fearture like 'day' , 'month' , 'year'

In [68]:
```
df['date']=pd.to_datetime(df[['day','month','year']])
df.drop(['day','month','year'], axis=1 , inplace = True)
```

In [69]:
```
df
```

Out[69]:

	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	region	date
0	29	57	18.0	0.0	65.7	3.4	7.6	1.3	3.4	0.5	not fire	bejaia	2012-06-01
1	29	61	13.0	1.3	64.4	4.1	7.6	1.0	3.9	0.4	not fire	bejaia	2012-06-02
2	26	82	22.0	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire	bejaia	2012-06-03
3	25	89	13.0	2.5	28.6	1.3	6.9	0.0	1.7	0	not fire	bejaia	2012-06-04
4	27	77	16.0	0.0	64.8	3.0	14.2	1.2	3.9	0.5	not fire	bejaia	2012-06-05
...	...	...	...	...	...	...	...	...	...	...	...	...	...
239	30	65	14.0	0.0	85.4	16.0	44.5	4.5	16.9	6.5	fire	Sidi-Bel Abbes	2012-09-26
240	28	87	15.0	4.4	41.1	6.5	8	0.1	6.2	0	not fire	Sidi-Bel Abbes	2012-09-27
241	27	87	29.0	0.5	45.9	3.5	7.9	0.4	3.4	0.2	not fire	Sidi-Bel Abbes	2012-09-28
242	24	54	18.0	0.1	79.7	4.3	15.2	1.7	5.1	0.7	not fire	Sidi-Bel Abbes	2012-09-29
243	24	64	15.0	0.2	67.3	3.8	16.5	1.2	4.8	0.5	not fire	Sidi-Bel Abbes	2012-09-30

244 rows × 13 columns

Exploring data

In [70]:
```
# Profiling of the Data
```

In [72]:
```
df.shape  # shape of datasets
```

Out[72]:
```
(244, 13)
```

Observation

In [75]:
```
df.columns  # 13 columns
```

Out[75]:
```
Index(['Temperature', 'RH', 'Ws', 'Rain', 'FFMC', 'DMC', 'DC', 'ISI', 'BUI',
       'FWI', 'Classes', 'region', 'date'],
      dtype='object')
```

In [79]:
```
df.isnull()  # indicate the  no  missing values
```

Out[79]:

	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	region	
0	False	False	False	False	False	False	False	False	False	False	False	False	F
1	False	False	False	False	False	False	False	False	False	False	False	False	F
2	False	False	False	False	False	False	False	False	False	False	False	False	F
3	False	False	False	False	False	False	False	False	False	False	False	False	F
4	False	False	False	False	False	False	False	False	False	False	False	False	F
...	...	...	...	...	...	...	...	...	...	...	...	...	
239	False	False	False	False	False	False	False	False	False	False	False	False	F
240	False	False	False	False	False	False	False	False	False	False	False	False	F
241	False	False	False	False	False	False	False	False	False	False	False	False	F
242	False	False	False	False	False	False	False	False	False	False	False	False	F
243	False	False	False	False	False	False	False	False	False	False	False	False	F

244 rows × 13 columns

In [81]:
```
df.isnull().sum()   # indicate the null values in Classes column
```

Out[81]:
```
Temperature    0
RH             0
Ws             0
Rain           0
FFMC           0
DMC            0
DC             0
ISI            0
BUI            0
FWI            0
Classes        1
region         0
date           0
dtype: int64
```

In [82]:
```
df.isnull().sum().sum()
```

Out[82]:
```
1
```

In [84]:
```
df.nunique()  # indicate the uniques values
```

Out[84]:
```
Temperature     19
RH              62
Ws              18
Rain            39
FFMC           173
DMC            166
DC             198
ISI            106
BUI            174
FWI            127
Classes          2
region           2
date           122
dtype: int64
```

In [85]:
```
df['Classes'].unique()
```

Out[85]:
```
array(['not fire', 'fire', nan], dtype=object)
```

Handling Categorical Features Classes

In [86]:
```
df['Classes']=df['Classes'].map({'not fire':0 , 'fire' : 1})
df.head()
```

Out[86]:

	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	region	date
0	29	57	18.0	0.0	65.7	3.4	7.6	1.3	3.4	0.5	0.0	bejaia	2012-06-01
1	29	61	13.0	1.3	64.4	4.1	7.6	1.0	3.9	0.4	0.0	bejaia	2012-06-02
2	26	82	22.0	13.1	47.1	2.5	7.1	0.3	2.7	0.1	0.0	bejaia	2012-06-03
3	25	89	13.0	2.5	28.6	1.3	6.9	0.0	1.7	0	0.0	bejaia	2012-06-04
4	27	77	16.0	0.0	64.8	3.0	14.2	1.2	3.9	0.5	0.0	bejaia	2012-06-05

Focus on Replacing Null Values

In [90]:
```
df['Classes'].mode()[0]
```

Out[90]:
```
1.0
```

In [91]:
```
df['Classes'] = df['Classes'].fillna(df['Classes'].mode()[0])
```

In [94]:
```
df.isnull().sum()   # no null values
```

Out[94]:
```
Temperature    0
RH             0
Ws             0
Rain           0
FFMC           0
DMC            0
DC             0
ISI            0
BUI            0
FWI            0
Classes        0
region         0
date           0
dtype: int64
```

In [95]:
```
df['Classes'].unique()
```

Out[95]:
```
array([0., 1.])
```

Check Datatypes in the datasets

In [96]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 244 entries, 0 to 243
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype         
---  ------       --------------  -----         
 0   Temperature  244 non-null    int32         
 1   RH           244 non-null    int32         
 2   Ws           244 non-null    float64       
 3   Rain         244 non-null    float64       
 4   FFMC         244 non-null    float64       
 5   DMC          244 non-null    float64       
 6   DC           244 non-null    object        
 7   ISI          244 non-null    float64       
 8   BUI          244 non-null    float64       
 9   FWI          244 non-null    object        
 10  Classes      244 non-null    float64       
 11  region       244 non-null    object        
 12  date         244 non-null    datetime64[ns]
dtypes: datetime64[ns](1), float64(7), int32(2), object(3)
memory usage: 23.0+ KB
```

In [98]:
```
df.memory_usage()   # indicate the use if memory in the datasets
```

Out[98]:
```
Index           128
Temperature     976
RH              976
Ws             1952
Rain           1952
FFMC           1952
DMC            1952
DC             1952
ISI            1952
BUI            1952
FWI            1952
Classes        1952
region         1952
date           1952
dtype: int64
```

Numerical and Categorical columns

In [103]:
```
cat_column= [feature for feature in df.columns if df[feature].dtype=='object']
```

In [104]:
```
cat_column
```

Out[104]:
```
['DC', 'FWI', 'region']
```

In [108]:
```
num_column = [feature for feature in df.columns if df[feature].dtype!='object']
```

In [109]:
```
num_column
```

Out[109]:
```
['Temperature',
 'RH',
 'Ws',
 'Rain',
 'FFMC',
 'DMC',
 'ISI',
 'BUI',
 'Classes',
 'date']
```

Feature Information

In [110]:
```
df.head()
```

Out[110]:

	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	region	date
0	29	57	18.0	0.0	65.7	3.4	7.6	1.3	3.4	0.5	0.0	bejaia	2012-06-01
1	29	61	13.0	1.3	64.4	4.1	7.6	1.0	3.9	0.4	0.0	bejaia	2012-06-02
2	26	82	22.0	13.1	47.1	2.5	7.1	0.3	2.7	0.1	0.0	bejaia	2012-06-03
3	25	89	13.0	2.5	28.6	1.3	6.9	0.0	1.7	0	0.0	bejaia	2012-06-04
4	27	77	16.0	0.0	64.8	3.0	14.2	1.2	3.9	0.5	0.0	bejaia	2012-06-05

Weather data observation:-

1. Temperature : temperature noon (temperature max) in Celsius degree: 22 to 42
2. RH : Relative Humidity in %: 21 to 90
3. Ws : Wind speed in km/h: 6to 29
4. Rain: total day in mm 0 to 16.8

FWI Components

5. FFMC- Fine Fuel Moisture Code index from the FWI system 28.6 to 92.5
6. DMC- Duff Moisture Code index from the FWI system 1.1 to 65.9
7. DC- Drought Code index from the FWI system 7 to 220.4
8. ISI- Initial Speard Index from the FWI system 0 TO 18.5
9. BUI - Buildup Index from the FWI system 1.1 to 68
10. FWI- Fire Weather index 0 to 31.1

Classes---- two classes 1. Fire 2. not fire

Region---two region 1. Bejaia is represent 0 2. Sibi Bel-Abbes Region represent 1

Date--- Date Displayed in (DD/MM/YYYY) format in dataset

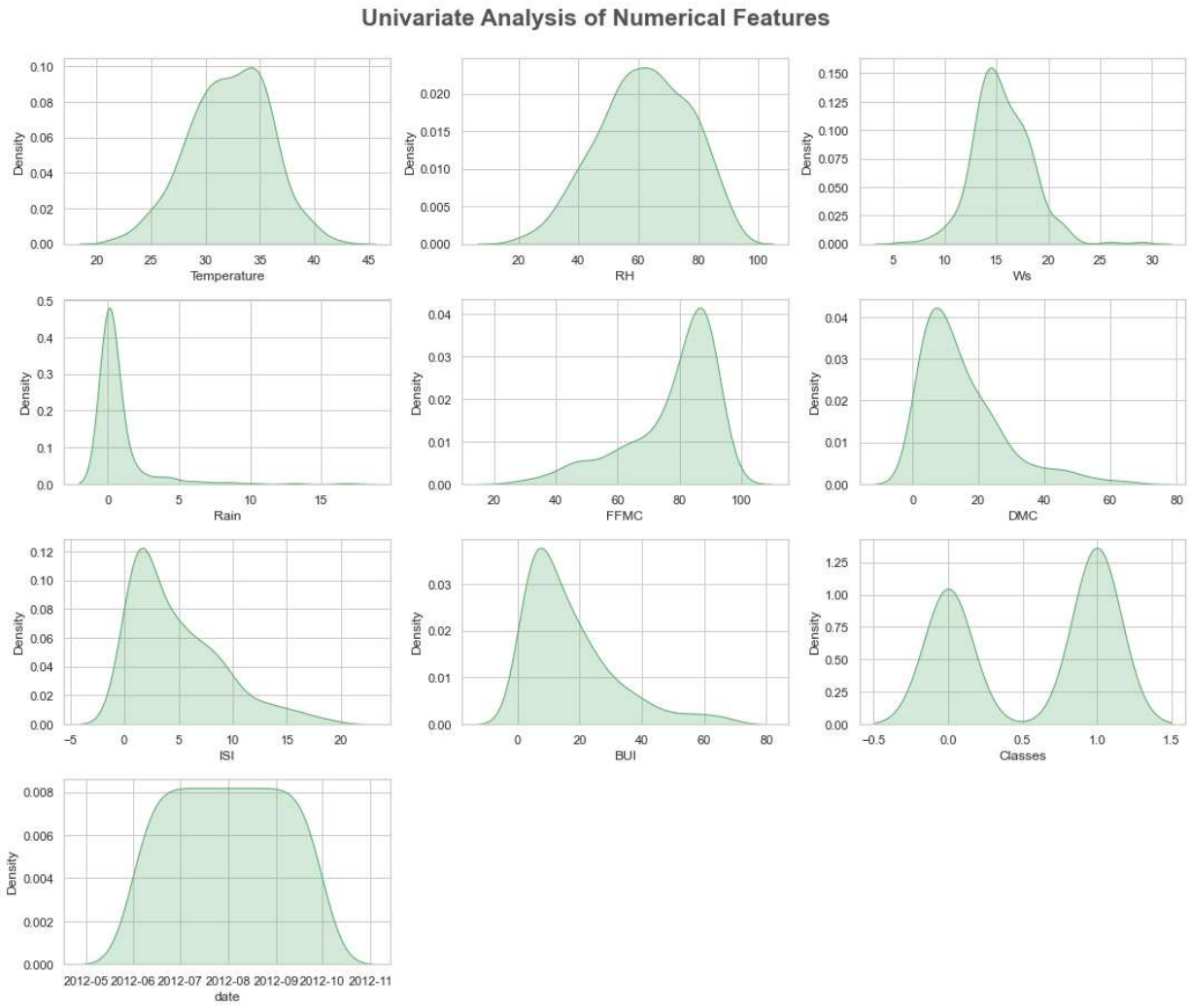
Univariate Analysis

In [112]:
```
df.var()    # analysis of one variable
```

Out[112]:
```
Temperature     13.204817
RH             221.539415
Ws               7.897102
Rain             3.997623
FFMC           205.565939
DMC            152.968382
ISI             17.433281
BUI            201.777024
Classes          0.246711
dtype: float64
```

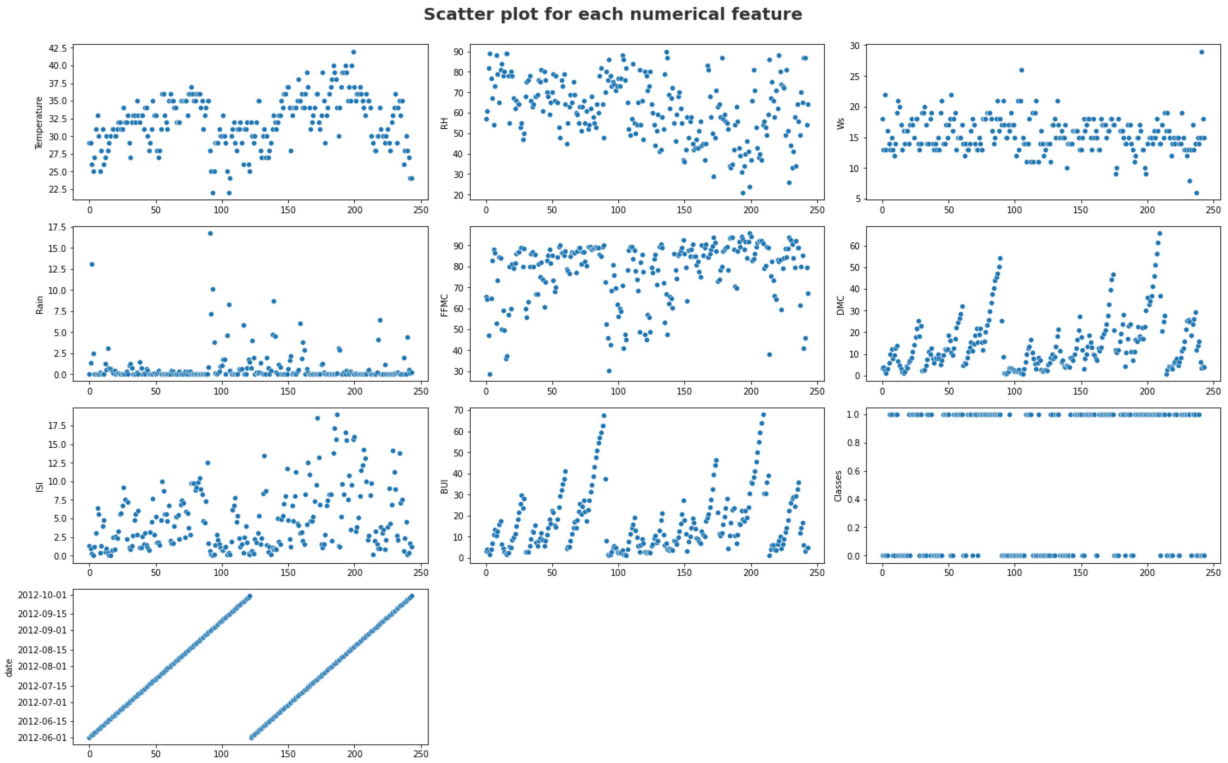
Numerical Features Analysis

In [175]:
```
plt.figure(figsize=(15,15))
plt.suptitle('Univariate Analysis of Numerical Features', fontsize =21 ,fontweigh
for i in range(0, len(num_column)):
    plt.subplot(5, 3, i+1)
    sns.kdeplot(x=df[num_column[i]],shade=True, color='g')
    plt.xlabel(num_column[i])
    plt.tight_layout()
```



Scatter plot to see the trend in each numerical columns

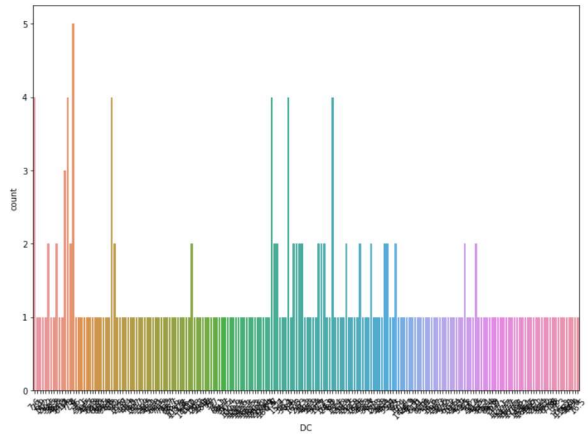
In [128]:
```
plt.figure(figsize=(20,15))
plt.suptitle('Scatter plot for each numerical feature', fontsize = 20 , fontweigh
for i in range(0 , len(num_column)):
    plt.subplot(5,3,i+1)
    sns.scatterplot(y=num_column[i],x=df.index, data = df)
    plt.tight_layout()
```

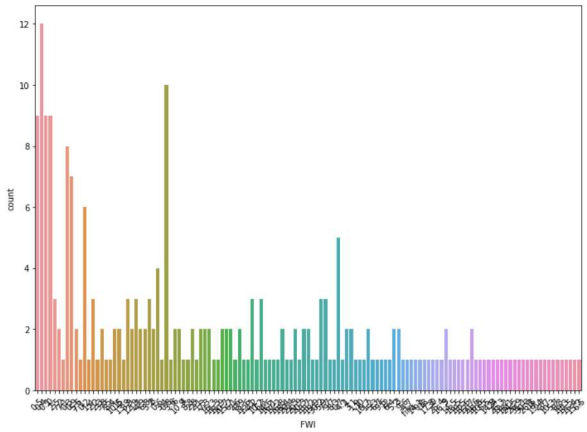


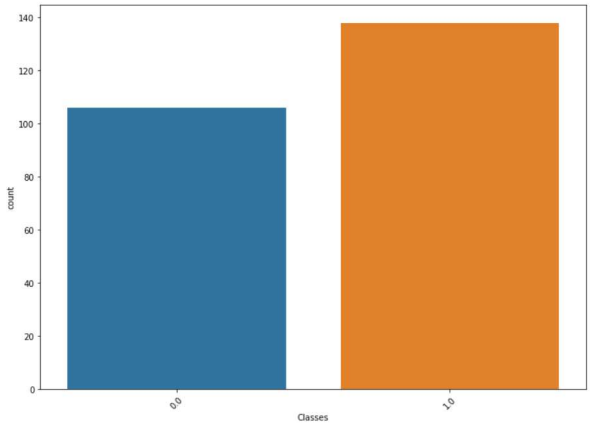
Categorical Feature Analysis

In [134]:
```
plt.figure(figsize=(20,15))
plt.suptitle('Univeriate Analysis of Categorical Feature',fontsize = 20, fontweig
cat_column=['DC','FWI','Classes']
for i in range(0 , len(cat_column)):
    plt.subplot(2 , 2, i+1)
    sns.countplot(x=df[cat_column[i]])
    plt.xlabel(cat_column[i])
    plt.xticks(rotation=45)
    plt.tight_layout()
```

Univeriate Analysis of Categorical Feature







observation
* Extreme value of Temperature is above 40
* Most of the RH is above 30

* WS values lie between 10 to 20

Bivariate analysis and multivariate analysis

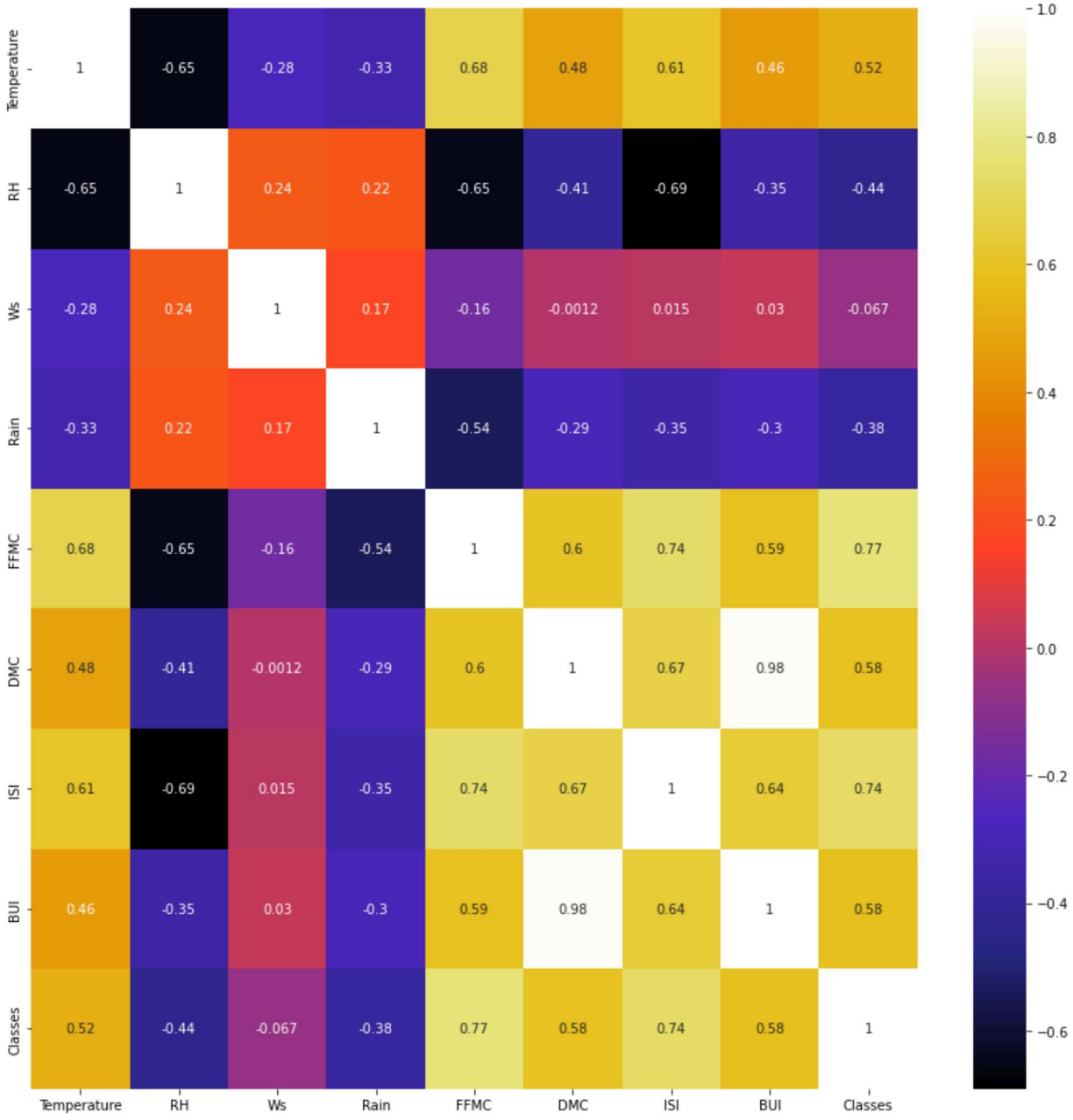
In [135]:
```
# stripplot (categorical vs numerical)
# scatterplot / pairplot (numerical vs numerical) (check correelation)
# bos plot (outliear)
# heatmap (correlation)
# lineplot (trend in numerical featurewith time)
```

In [138]:
```
df.corr()  # mulitcolleniarity in mumerical features
```

Out[138]:

	Temperature	RH	Ws	Rain	FFMC	DMC	ISI	B
Temperature	1.000000	-0.654443	-0.278132	-0.326786	0.677491	0.483105	0.607551	0.4555(
RH	-0.654443	1.000000	0.236084	0.222968	-0.645658	-0.405133	-0.690637	-0.3485
Ws	-0.278132	0.236084	1.000000	0.170169	-0.163255	-0.001246	0.015248	0.0297
Rain	-0.326786	0.222968	0.170169	1.000000	-0.544045	-0.288548	-0.347105	-0.2991
FFMC	0.677491	-0.645658	-0.163255	-0.544045	1.000000	0.602391	0.739730	0.5896
DMC	0.483105	-0.405133	-0.001246	-0.288548	0.602391	1.000000	0.674499	0.9820
ISI	0.607551	-0.690637	0.015248	-0.347105	0.739730	0.674499	1.000000	0.6358
BUI	0.455504	-0.348587	0.029756	-0.299171	0.589652	0.982073	0.635891	1.0000(
Classes	0.518119	-0.435023	-0.066529	-0.379449	0.770114	0.584188	0.735511	0.5838

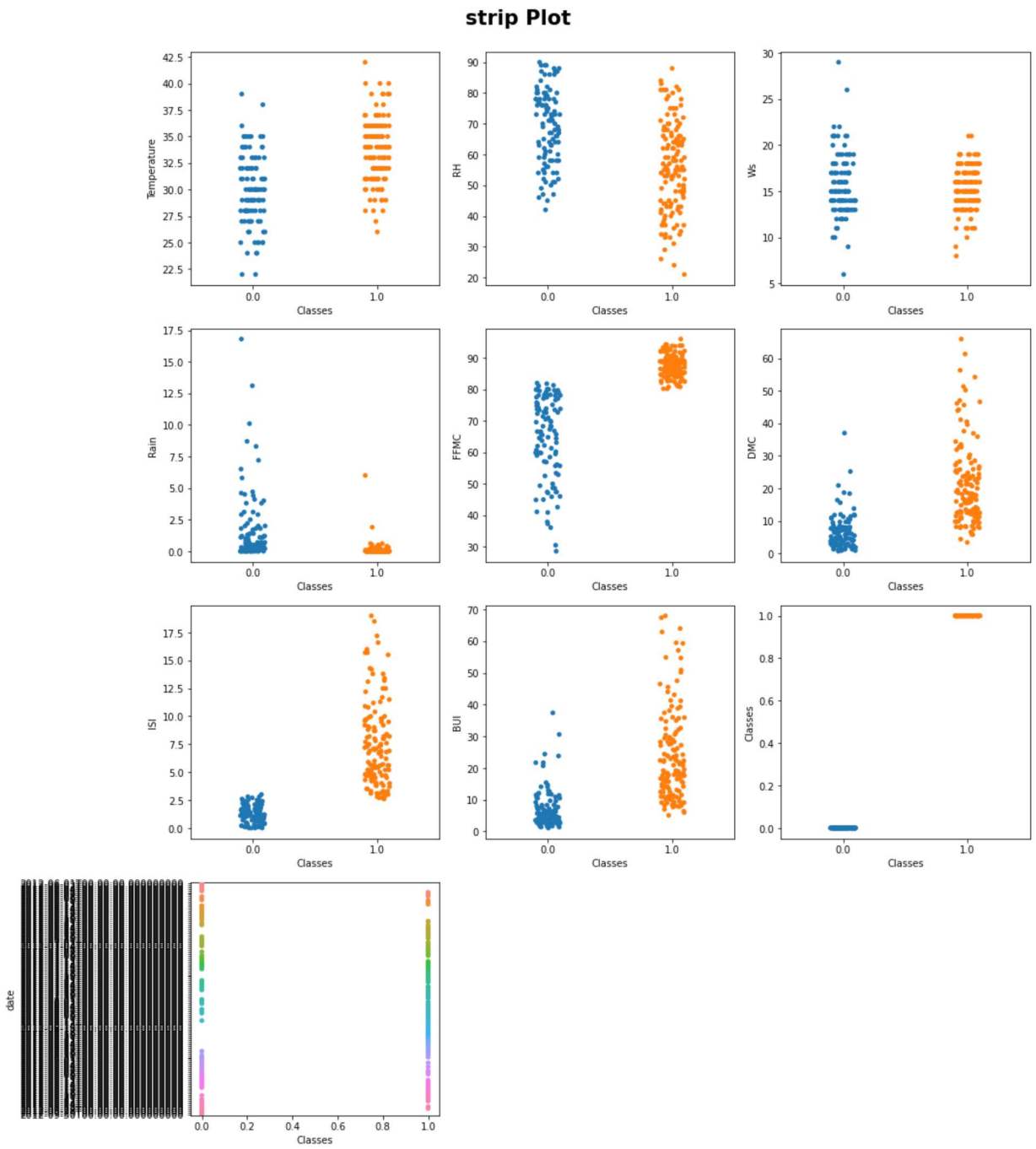
In [139]:
```
plt.figure(figsize=(15,15))
sns.heatmap(df.corr(),cmap ='CMRmap',annot=True)
plt.show()
```



```
observation
    * Highly +ve correlated feature are DMC and BUI
    * Highly -ve correlated feature are RH and FFMC , RH and ISI
```

Strip plot to see the relationship between numerical features and target

In [143]:
```
plt.figure(figsize=(15,20))
plt.suptitle('strip Plot', fontsize =21 , fontweight ='bold' , alpha = 1 , y= 1)
for i in range (0 , len(num_column)):
    plt.subplot(5 , 3 ,i+1)
    sns.stripplot(y= num_column[i], x='Classes' , data=df)
    plt.tight_layout()
```

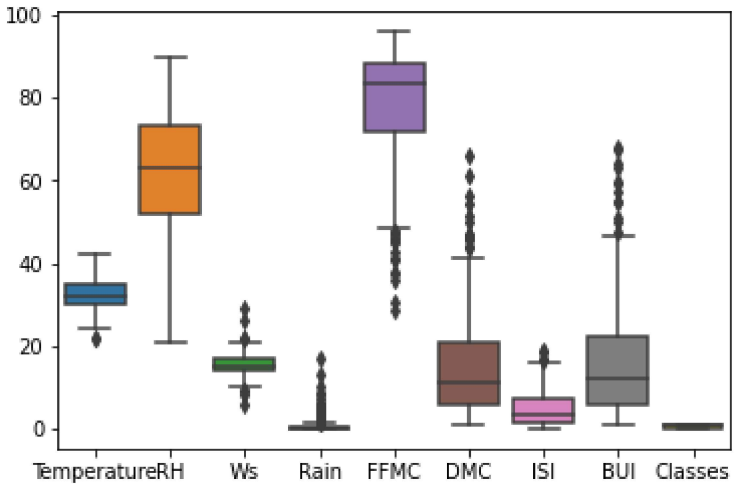


```
observation
* Note = Here 0 ='not fire' and 1 = 'fire'
* place with higher temperature has fire
* place with lower RH has fire
* place with FFMC > 80 has fire
* place with ISI > 2.5 has fire
* place with Rain < 2 has fire
```

Boxplot to find Outlier in the features

In [145]:
```
sns.boxplot(data=df,orient='v')   # indicate the outliear TM , Ws ,Rain , FFMC ,
```

Out[145]: <AxesSubplot:>



Statistical Analysis

In [146]:
```
df.describe()
```

Out[146]:

	Temperature	RH	Ws	Rain	FFMC	DMC	ISI	
count	244.000000	244.000000	244.000000	244.000000	244.000000	244.000000	244.000000	244.0
mean	32.172131	61.938525	15.504098	0.760656	77.887705	14.673361	4.774180	16.6
std	3.633843	14.884200	2.810178	1.999406	14.337571	12.368039	4.175318	14.2
min	22.000000	21.000000	6.000000	0.000000	28.600000	0.700000	0.000000	1.1
25%	30.000000	52.000000	14.000000	0.000000	72.075000	5.800000	1.400000	6.0
50%	32.000000	63.000000	15.000000	0.000000	83.500000	11.300000	3.500000	12.2
75%	35.000000	73.250000	17.000000	0.500000	88.300000	20.750000	7.300000	22.5
max	42.000000	90.000000	29.000000	16.800000	96.000000	65.900000	19.000000	68.0

In [147]:
```
df.mean()
```

Out[147]:
```
Temperature    32.172131
RH             61.938525
Ws             15.504098
Rain            0.760656
FFMC           77.887705
DMC            14.673361
ISI             4.774180
BUI            16.664754
Classes         0.565574
dtype: float64
```

In [148]:
```
df.median()
```

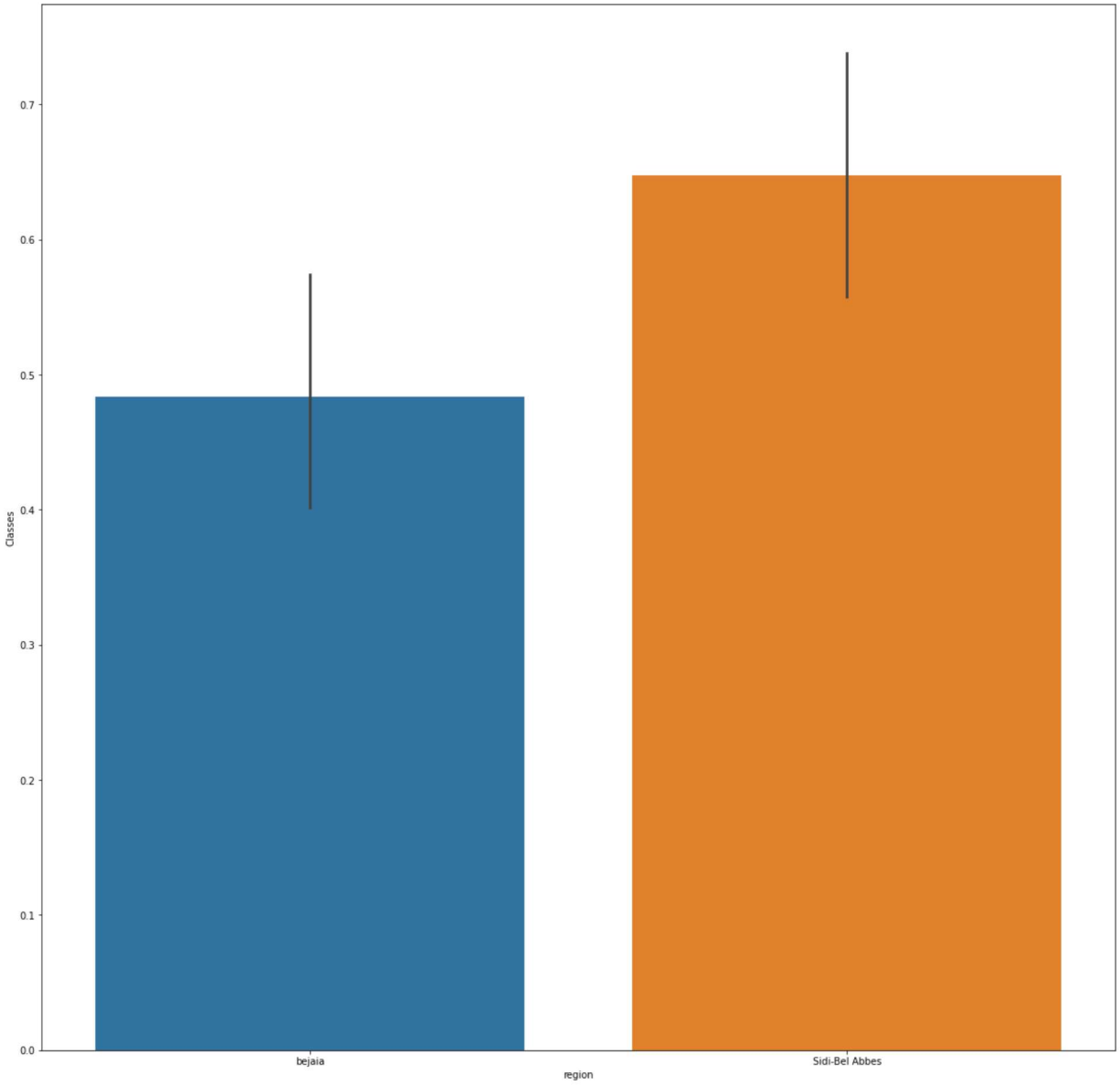
Out[148]:
```
Temperature    32.00
RH             63.00
Ws             15.00
Rain            0.00
FFMC           83.50
DMC            11.30
ISI             3.50
BUI            12.25
Classes         1.00
dtype: float64
```

Graphical Analysis

In [149]:
```
# which area has most of the time fire happen?
```

In [151]:
```
import matplotlib
matplotlib.rcParams['figure.figsize']=(20,20)
sns.barplot(x='region' , y ='Classes' , data=df)
```

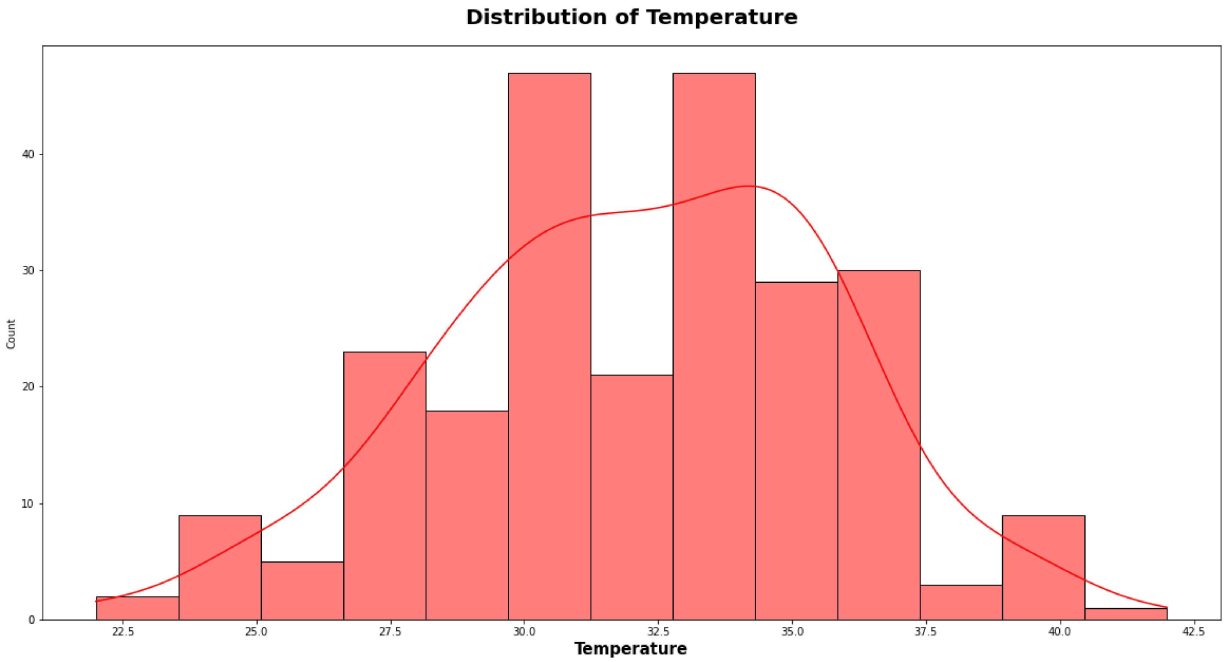
Out[151]: <AxesSubplot:xlabel='region', ylabel='Classes'>



Temperature Range which is in most of the place?

In [156]:
```
plt.subplots(figsize=(20,10))
sns.histplot('Distribution of temperature ', x = df.Temperature,color='r', kde=Tr
plt.title('Distribution of Temperature', weight = 'bold' , fontsize=20, pad =20)
plt.xlabel('Temperature' ,weight ='bold' , fontsize=15)
```

Out[156]:
```
Text(0.5, 0, 'Temperature')
```

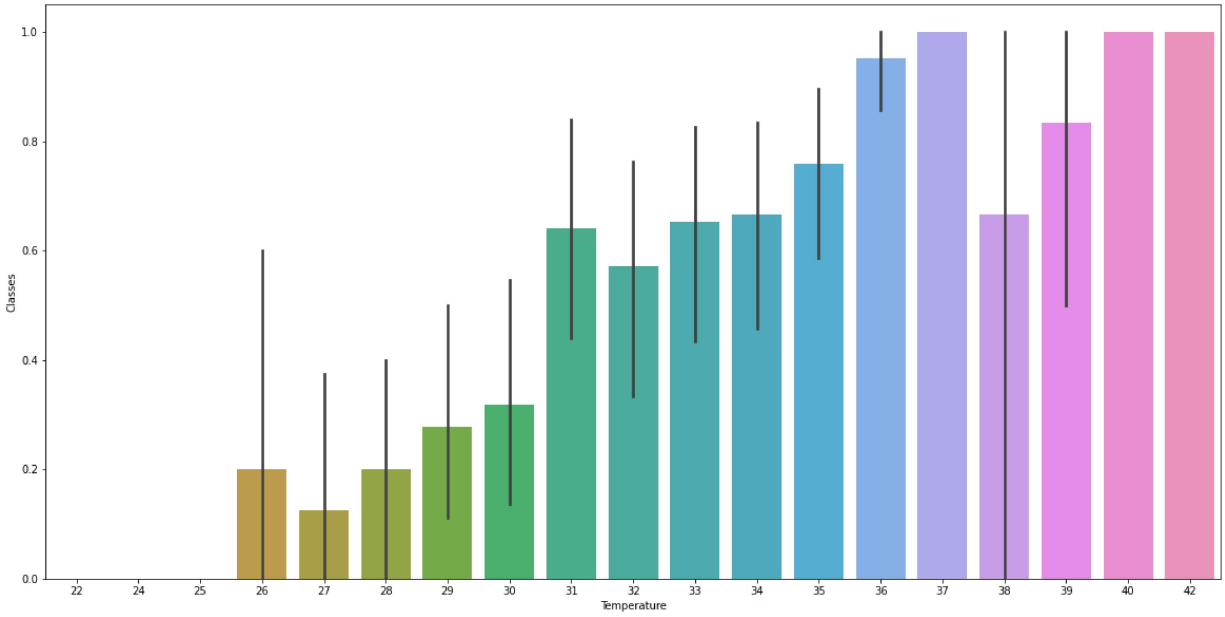



```
observation--
* Temperature occure most of the time in range 32.5 to 35.0
```

Highest Temperature attained

In [162]:
```
import matplotlib
matplotlib.rcParams['figure.figsize']=(20,10)
sns.barplot(x='Temperature' , y = 'Classes' , data=df)
```

Out[162]: <AxesSubplot:xlabel='Temperature', ylabel='Classes'>

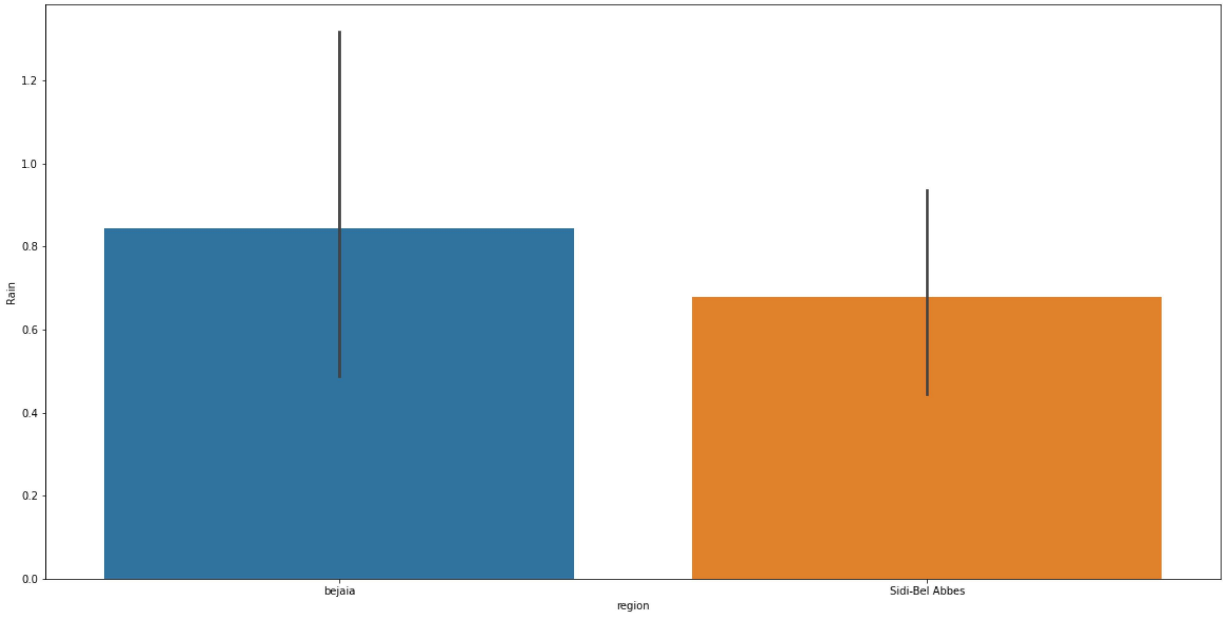



```
Observation
* Highest temperature is 42,40,37
```

which region has most time rain happens

In [163]:
```
import matplotlib
matplotlib.rcParams['figure.figsize']=(20,10)
sns.barplot(x='region' , y ='Rain' , data = df)
```

Out[163]: <AxesSubplot:xlabel='region', ylabel='Rain'>

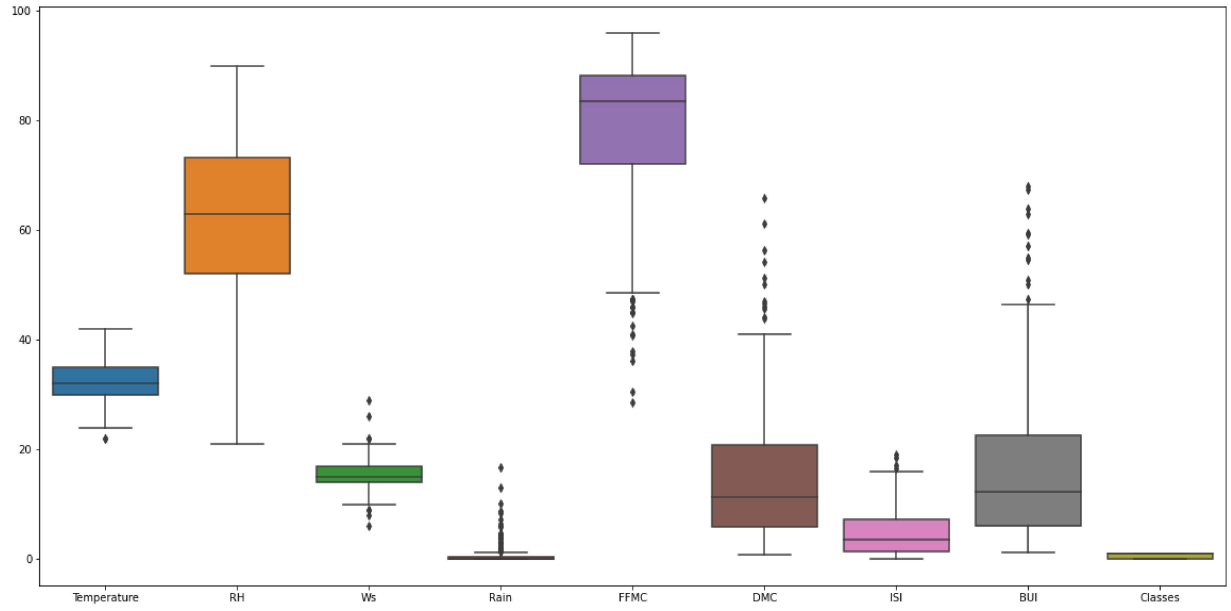



```
Observation
* Bejaia is the region in which most of the time rain happens
```

Boxplot to find Outliers in the features

In [164]:
```
sns.boxplot(data=df, orient='v')
```

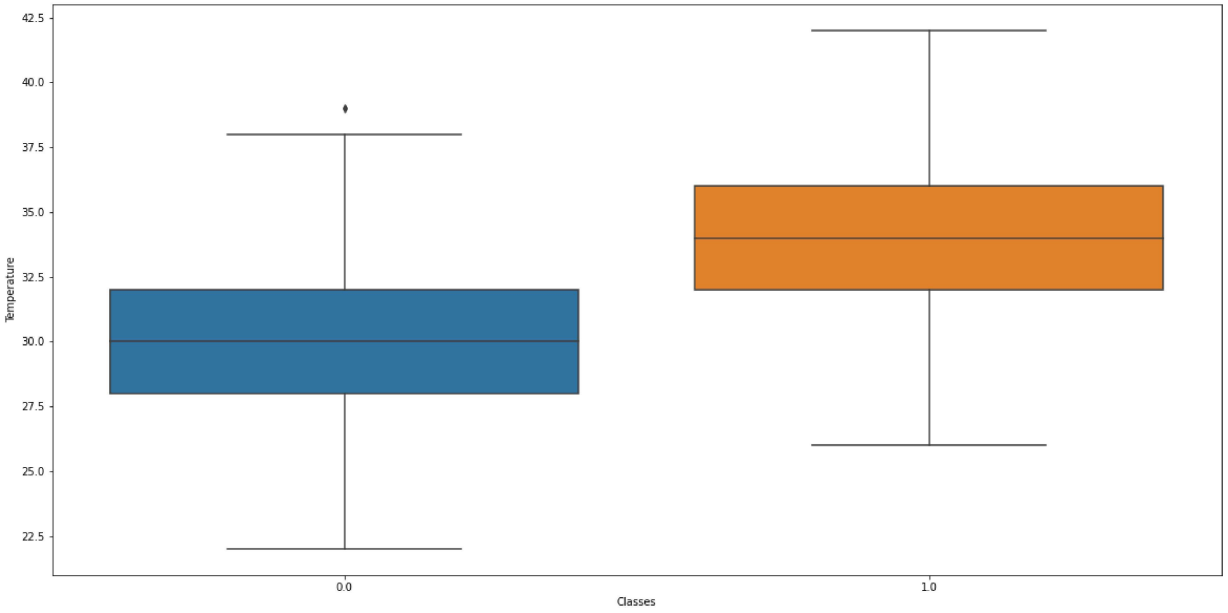
Out[164]: <AxesSubplot:>



Boxplot of Class VS Temperature

In [167]:
```
import seaborn as sns
sns.boxplot(x='Classes' , y='Temperature' , data =df)
```

Out[167]: <AxesSubplot:xlabel='Classes', ylabel='Temperature'>

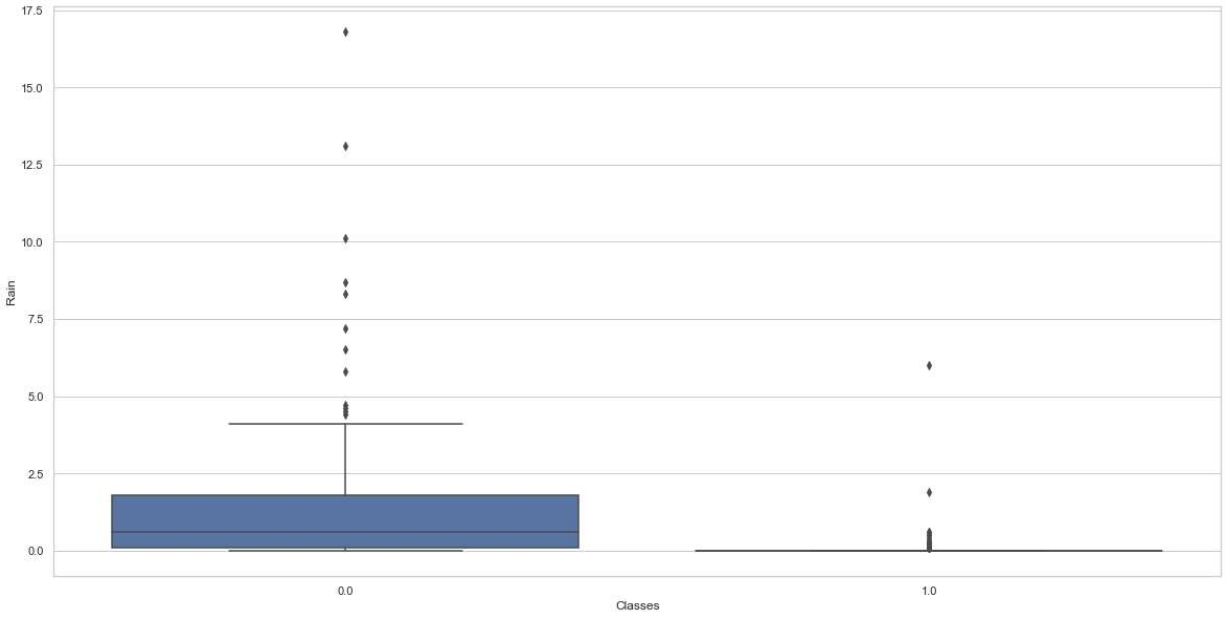



```
Observation
* one day at lower temperature fires occur
```

Boxplot of Classes VS Rain

In [173]:
```
import seaborn as sns
sns.boxplot(x='Classes' , y='Rain' , data = df)
```

Out[173]: <AxesSubplot:xlabel='Classes', ylabel='Rain'>



In [ ]: