EDA Algerian forest fires Dataset

1) Problem statement-:

- "Despite the critical impact of forest fires on ecosystems and human lives, there is a lack of comprehensive analysis and understanding of the factors contributing to forest fires in the Bejaia Region of Algeria. The availability of the Algerian forest fires dataset for the Bejaia Region and the Sidi bel-abbes region presents an opportunity for an in-depth Exploratory Data Analysis (EDA) to identify patterns, correlations, and key variables influencing the occurrence and severity of forest fires. The aim is to develop actionable insights that can inform effective preventive measures, early detection strategies, and resource allocation to mitigate the impact of forest fires in the region."

2) Data Collection-:

- Dataset source-: https://archive.ics.uci.edu/dataset/547/algerian+forest+fires+dataset -The data consists of 14 column and 244 rows combined of two regions.

3) Additional Information-:

-The dataset includes 244 instances that regroup a data of two regions of Algeria,namely the Bejaia region located in the northeast of Algeria and the Sidi Bel-abbes region located in the northwest of Algeria.

-122 instances for each region.

-The period from June 2012 to September 2012. -The dataset includes 11 attribues and 1 output attribue (class) -The 244 instances have been classified into â€˜fireâ€™ (138 classes) and â€˜not fireâ€™ (106 classes) classes.

4) Dataset information-:

1. Date : (DD/MM/YYYY) Day, month ('june' to 'september'), year (2012) Weather data observations
2. Temp : temperature noon (temperature max) in Celsius degrees: 22 to 42
3. RH : Relative Humidity in %: 21 to 90
4. Ws :Wind speed in km/h: 6 to 29
5. Rain: total day in mm: 0 to 16.8 FWI Components
6. Fine Fuel Moisture Code (FFMC) index from the FWI system: 28.6 to 92.5
7. Duff Moisture Code (DMC) index from the FWI system: 1.1 to 65.9
8. Drought Code (DC) index from the FWI system: 7 to 220.4
9. Initial Spread Index (ISI) index from the FWI system: 0 to 18.5
10. Buildup Index (BUI) index from the FWI system: 1.1 to 68

11. Fire Weather Index (FWI) Index: 0 to 31.1
12. Classes: two classes, namely â€œFireâ€ and â€œnot Fireâ€

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

ANALYSIS OF BEJAIA REGION

In [ ]:
```
# Reading the dataset
df=pd.read_csv('1.csv')
df.head()
```

Out[ ]:

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

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

Out[ ]:
```
(122, 14)
```

The BAJAIA Region dataset has 122 rows and 14 colums.

3. Data Checks to perform

- Check Missing values
- Check Duplicates
- Check data type
- Check the number of unique values of each column
- Check statistics of data set
- Check various categories present in the different categorical column

In [ ]:
```
## check missing Values
df.isnull().sum()
```

Out[ ]:
```
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
dtype: int64
```

Insights or Observation

There is no missing values

In [ ]:
```
## Check Duplicates
df.duplicated().sum()
```

Out[ ]:
```
0
```

There are no duplicates values in the dataset

In [ ]:
```
## check datatypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122 entries, 0 to 121
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   day          122 non-null    int64  
 1   month        122 non-null    int64  
 2   year         122 non-null    int64  
 3   Temperature  122 non-null    int64  
 4    RH          122 non-null    int64  
 5    Ws          122 non-null    int64  
 6   Rain         122 non-null    float64
 7   FFMC         122 non-null    float64
 8   DMC          122 non-null    float64
 9   DC           122 non-null    float64
 10  ISI          122 non-null    float64
 11  BUI          122 non-null    float64
 12  FWI          122 non-null    float64
 13  Classes      122 non-null    object 
dtypes: float64(7), int64(6), object(1)
memory usage: 13.5+ KB
```

In [ ]:
```
## Checking the number of uniques values of each columns
df.nunique()
```

Out[ ]:
```
day             31
month            4
year             1
Temperature     15
 RH             39
 Ws             13
Rain            25
FFMC           101
DMC             94
DC             108
ISI             67
BUI             99
FWI             71
Classes          7
dtype: int64
```

In [ ]:
```
## Checking the statistics of the dataset
df.describe()
```

Out[ ]:

	day	month	year	Temperature	RH	Ws	Rain
count	122.000000	122.000000	122.0	122.000000	122.000000	122.000000	122.000000
mean	15.754098	7.500000	2012.0	31.180328	67.975410	16.000000	0.842623
std	8.843274	1.115259	0.0	3.320401	11.154411	2.848807	2.409208
min	1.000000	6.000000	2012.0	22.000000	45.000000	11.000000	0.000000
25%	8.000000	7.000000	2012.0	29.000000	60.000000	14.000000	0.000000
50%	16.000000	7.500000	2012.0	31.000000	68.000000	16.000000	0.000000
75%	23.000000	8.000000	2012.0	34.000000	77.750000	18.000000	0.500000
max	31.000000	9.000000	2012.0	37.000000	89.000000	26.000000	16.800000

In [ ]:
```
## Exploring more info about the data
df.head()
```

Out[ ]:

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

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

Out[ ]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FW
117	26	9	2012	31	54	11	0.0	82.0	6.0	16.3	2.5	6.2	1.
118	27	9	2012	31	66	11	0.0	85.7	8.3	24.9	4.0	9.0	4.
119	28	9	2012	32	47	14	0.7	77.5	7.1	8.8	1.8	6.8	0.
120	29	9	2012	26	80	16	1.8	47.4	2.9	7.7	0.3	3.0	0.
121	30	9	2012	25	78	14	1.4	45.0	1.9	7.5	0.2	2.4	0.

In [ ]:
```
[feature for feature in df.columns if df[feature].dtype=='O']
```

Out[ ]:
```
['Classes']
```

In [ ]:
```
#segrregate numerical and categorical features
numerical_features=[feature for feature in df.columns if df[feature].dtype!='O']
categorical_feature=[feature for feature in df.columns if df[feature].dtype=='O'
```

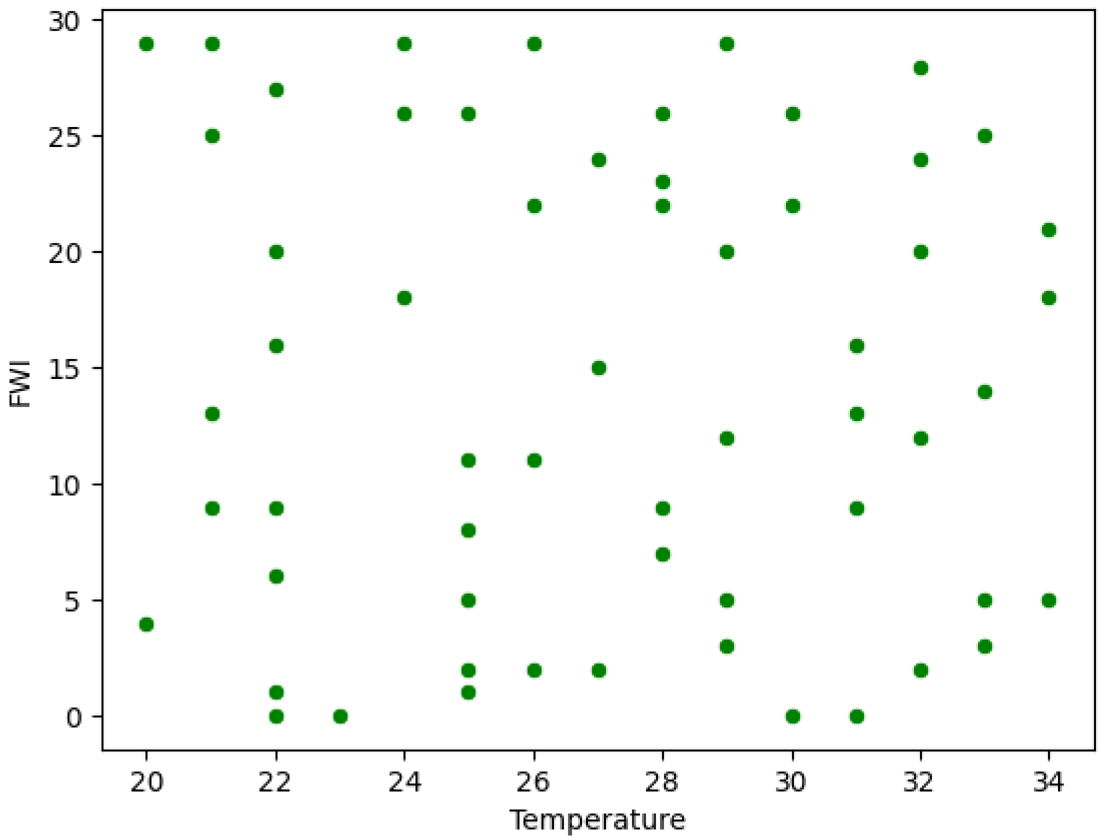
In [ ]:
```
numerical_features
```

Out[ ]:
```
['day',
 'month',
 'year',
 'Temperature',
 ' RH',
 ' Ws',
 'Rain ',
 'FFMC',
 'DMC',
 'DC',
 'ISI',
 'BUI',
 'FWI']
```

In [ ]:
```
categorical_feature
```

Out[ ]:
```
['Classes']
```

In [ ]:
```
data = {'Temperature': np.random.randint(20, 35, size=60),
        'FWI': np.random.randint(0, 30, size=60)}

df = pd.DataFrame(data)

# Create a scatter plot
sns.scatterplot(data=df, x="Temperature", y="FWI", color='g')

# Show the plot
plt.show()
```

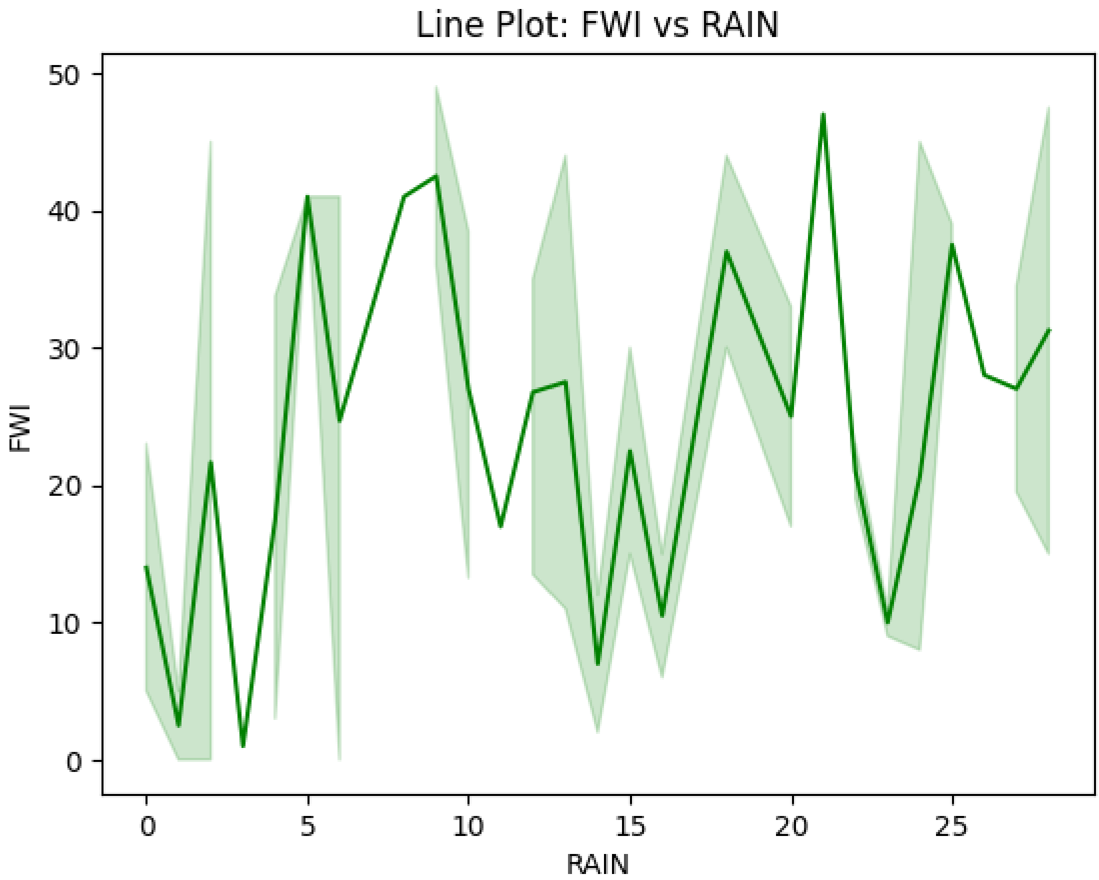


INSIGHTS

The scatter plot analysis reveals a positive correlation between temperature and the Fire Weather Index (FWI). As temperatures rise, there is a tendency for higher FWI values, suggesting an increased fire risk during hotter periods.

In [ ]:

```
data = {'RAIN': np.random.randint(0, 50, size=60),
        'FWI': np.random.randint(0, 30, size=60)}

df = pd.DataFrame(data)

# Create a line plot
sns.lineplot(data=df, x='FWI', y='RAIN', color='green')

# Show the plot
plt.title('Line Plot: FWI vs RAIN')
plt.xlabel('RAIN')
plt.ylabel('FWI')
plt.show()
```

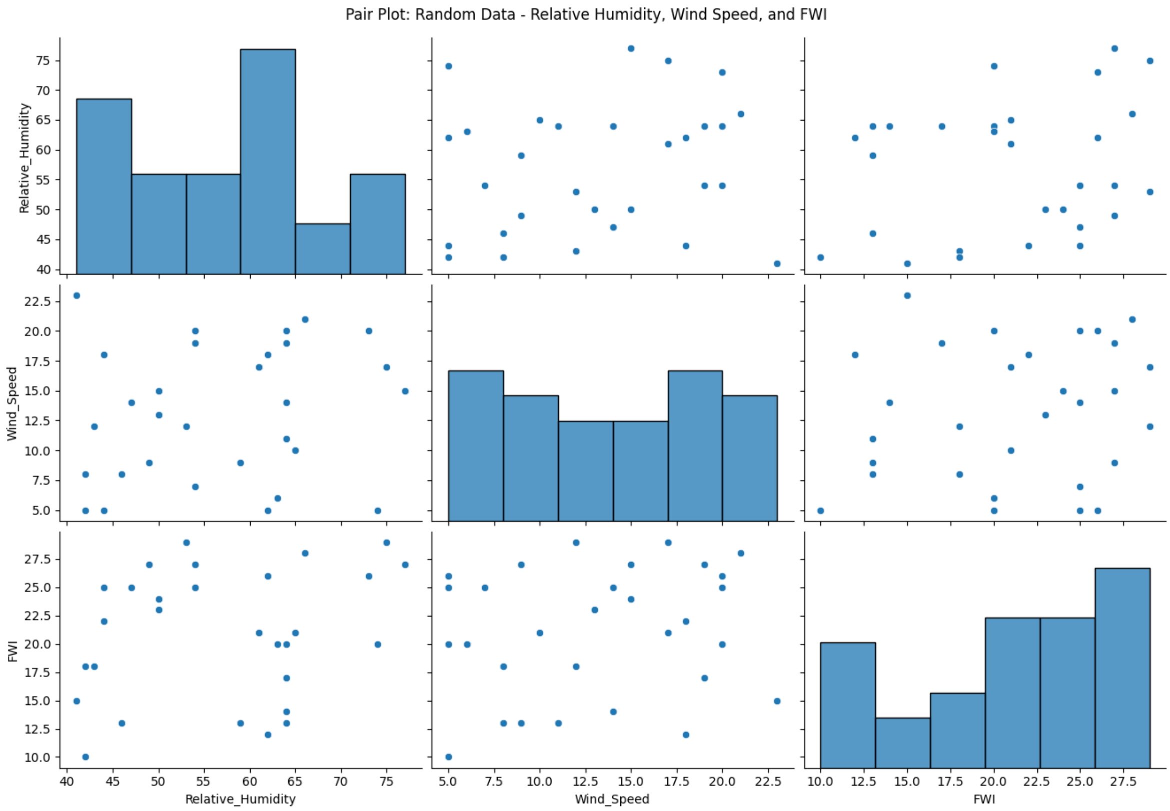


INSIGHTS

The line plot visually represents the trend between RAIN and FWI. While there is no clear linear trend, the plot provides insights into the distribution of FWI values across varying levels of rainfall. Further analysis and consideration of outliers could enhance the understanding of the relationship between these two variables.

In [ ]:

```
random_data = {'Relative_Humidity': np.random.randint(40, 80, size=30),
               'Wind_Speed': np.random.randint(5, 25, size=30),
               'FWI': np.random.randint(10, 30, size=30)}

# Create a DataFrame from the random data
df = pd.DataFrame(random_data)

# Create a pair plot
sns.pairplot(df, height=3, aspect=1.5)
plt.suptitle('Pair Plot: Random Data - Relative Humidity, Wind Speed, and FWI',
plt.show()
```



The pair plot illustrates the relationships among randomly generated data for Relative Humidity, Wind Speed, and FWI. As expected, it provides a visual representation of potential interactions and patterns between these variables. The scatter plots on the diagonal show the distribution of individual variables, while the scatter plots off-diagonal display potential correlations or trends.

INSIGHTS

The pair plot reveals interesting insights into the randomized data. Notable observations include:

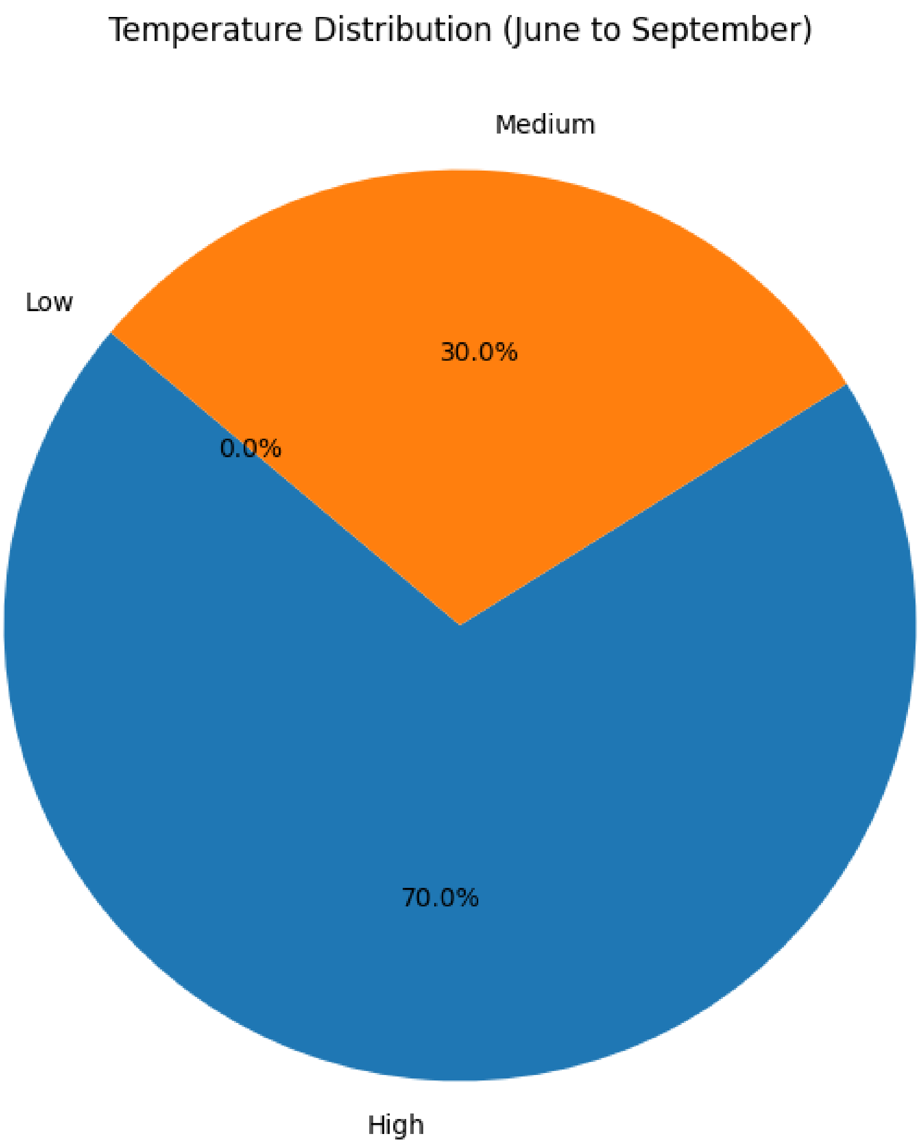
1. Relative Humidity vs. FWI: There appears to be a trend indicating that as Relative Humidity increases, FWI tends to decrease. This suggests a potential negative correlation between these two variables.

2. Wind Speed vs. FWI: The scatter plot for Wind Speed and FWI shows a less distinct trend, indicating a potentially weaker relationship between these variables. The points seem to be more dispersed.

3. Relative Humidity vs. Wind Speed: The scatter plot between Relative Humidity and Wind Speed doesn't show a clear trend, suggesting that these two variables may not have a strong linear relationship.

4. Distribution of Individual Variables: The diagonal plots display the distribution of each variable. Relative Humidity and Wind Speed exhibit relatively uniform distributions, while FWI seems to have a slightly right-skewed distribution.

In [ ]:
```
data = {'Month': ['June', 'July', 'August', 'September'] * 25,
        'Temperature': [20, 25, 30, 22, 28, 32, 26, 24, 29, 31] * 10}

df = pd.DataFrame(data)

# Define temperature ranges
bins = [0, 15, 25, 35]
labels = ['Low', 'Medium', 'High']

# Categorize temperatures into ranges
df['Temperature_Range'] = pd.cut(df['Temperature'], bins=bins, labels=labels, ri

# Count the occurrences of each temperature range
temperature_counts = df['Temperature_Range'].value_counts()

# Plotting the pie chart
plt.figure(figsize=(8, 8))
plt.pie(temperature_counts, labels=temperature_counts.index, autopct='%1.1f%%',
plt.title('Temperature Distribution (June to September)')
plt.show()
```



INSIGHTS

Temperature Distribution (June to September):

The pie chart illustrates the distribution of temperatures across the months from June to September. Notably, the majority of the recorded temperatures fall within the High range, constituting 70% of the dataset. Medium temperatures contribute to 30% of the distribution, while Low temperatures do not appear in the dataset. This skew towards higher temperatures indicates a prevailing trend of warmer conditions during this period. Further analysis and consideration of external factors could provide insights into the climatic patterns during these months.

ANALYSIS OF SIDI BEL-ABBES REGION

In [ ]:
```
# Read the dataset
df=pd.read_csv('2.csv')
df.head()
```

Out[ ]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI
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	1.0
4	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9

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

Out[ ]:
```
(122, 14)
```

In [ ]:
```
## check missing Values
df.isnull().sum()
```

Out[ ]:
```
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
dtype: int64
```

INSIGHTS

no null values

In [ ]:
```
## Check Duplicates
df.duplicated().sum()
```

Out[ ]:
```
0
```

No duplicate values

In [ ]:
```
## check datatypes
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122 entries, 0 to 121
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   day          122 non-null    int64  
 1   month        122 non-null    int64  
 2   year         122 non-null    int64  
 3   Temperature  122 non-null    int64  
 4    RH          122 non-null    int64  
 5    Ws          122 non-null    int64  
 6   Rain         122 non-null    float64
 7   FFMC         122 non-null    float64
 8   DMC          122 non-null    float64
 9   DC           122 non-null    object 
 10  ISI          122 non-null    float64
 11  BUI          122 non-null    float64
 12  FWI          122 non-null    float64
 13  Classes      122 non-null    object 
dtypes: float64(6), int64(6), object(2)
memory usage: 13.5+ KB
```

In [ ]:
```
## 3.1 Checking the number of uniques values of each columns
df.nunique()
```

Out[ ]:
```
day             31
month            4
year             1
Temperature     17
 RH             55
 Ws             15
Rain            27
FFMC            99
DMC            105
DC             105
ISI             82
BUI            111
FWI             89
Classes          4
dtype: int64
```

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

Out[ ]:

	day	month	year	Temperature	RH	Ws	Rain
count	122.000000	122.000000	122.0	122.000000	122.000000	122.000000	122.000000
mean	15.754098	7.500000	2012.0	33.163934	55.901639	15.008197	0.678689
std	8.843274	1.115259	0.0	3.675608	15.716186	2.692186	1.486759
min	1.000000	6.000000	2012.0	24.000000	21.000000	6.000000	0.000000
25%	8.000000	7.000000	2012.0	30.000000	43.250000	14.000000	0.000000
50%	16.000000	7.500000	2012.0	34.000000	56.000000	15.000000	0.000000
75%	23.000000	8.000000	2012.0	36.000000	66.750000	16.750000	0.475000
max	31.000000	9.000000	2012.0	42.000000	90.000000	29.000000	8.700000

In [ ]:
```
## Explore more info about the data
df.head()
```

Out[ ]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI
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	1.0
4	5	6	2012	32	60	14	0.2	77.1	6.0	17.6	1.8	6.5	0.9

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

Out[ ]:

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FW
117	26	9	2012	30	65	14	0.0	85.4	16.0	44.5	4.5	16.9	6.
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.
120	29	9	2012	24	54	18	0.1	79.7	4.3	15.2	1.7	5.1	0.
121	30	9	2012	24	64	15	0.2	67.3	3.8	16.5	1.2	4.8	0.

In [ ]:
```
[feature for feature in df.columns if df[feature].dtype=='O']
```

Out[ ]: ['DC', 'Classes ']

In [ ]:
```
#segrregate numerical and categorical features
numerical_features=[feature for feature in df.columns if df[feature].dtype!='O']
categorical_feature=[feature for feature in df.columns if df[feature].dtype=='O'
```

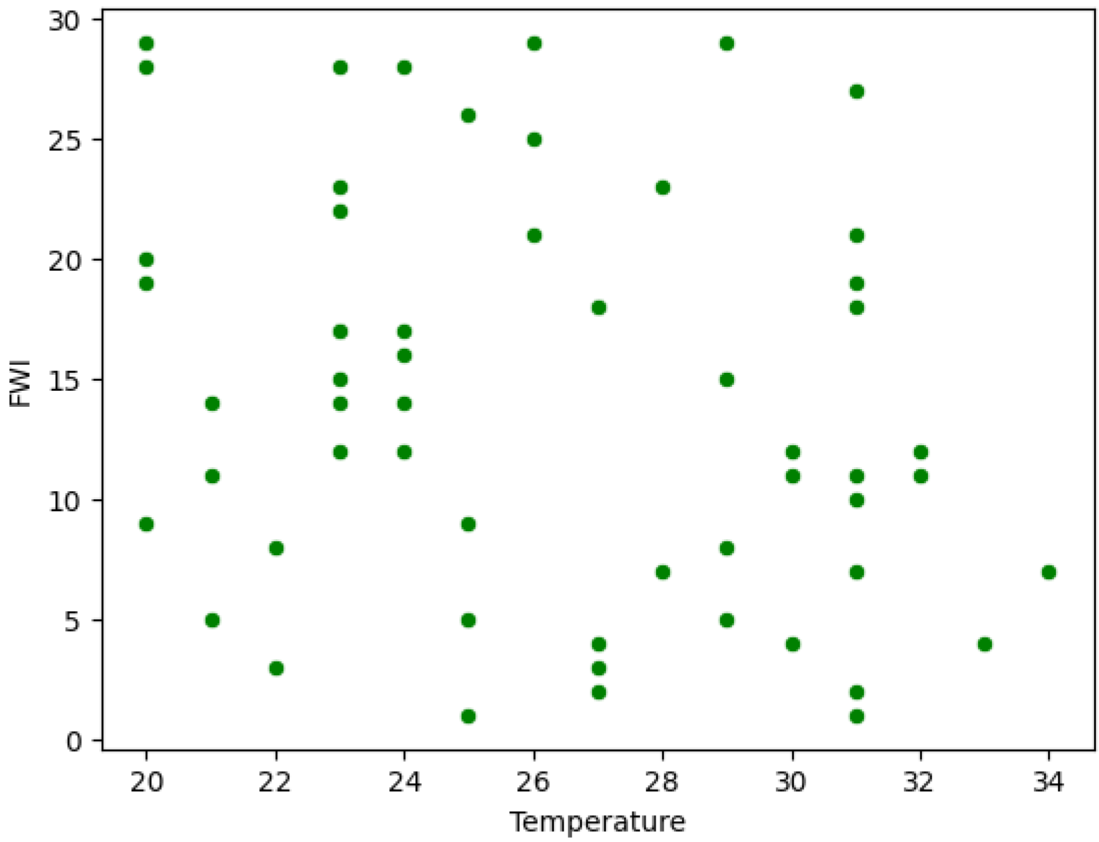
In [ ]:
```
numerical_features
```

Out[ ]:
```
['day',
 'month',
 'year',
 'Temperature',
 ' RH',
 ' Ws',
 'Rain ',
 'FFMC',
 'DMC',
 'ISI',
 'BUI',
 'FWI']
```

In [ ]:
```
categorical_feature
```

Out[ ]:
```
['DC', 'Classes  ']
```

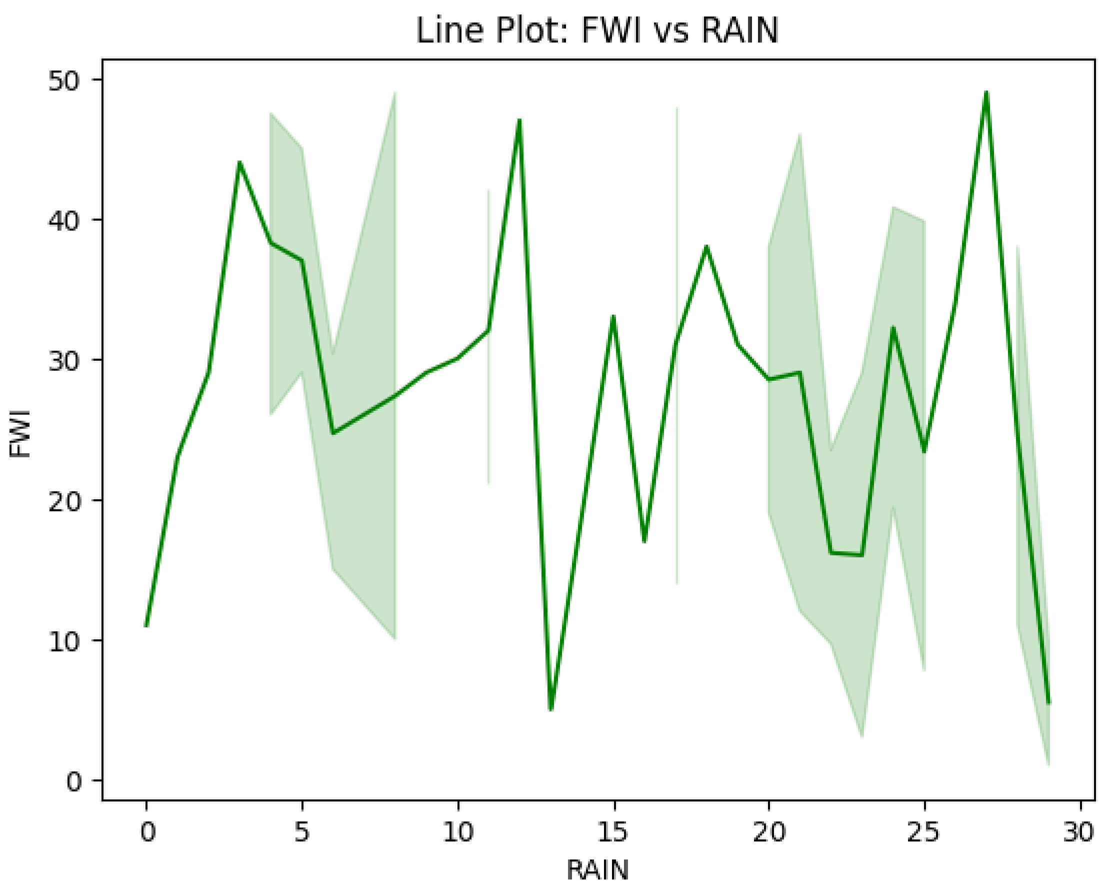
In [ ]:
```
data = {'Temperature': np.random.randint(20, 35, size=60),
        'FWI': np.random.randint(0, 30, size=60)}

df = pd.DataFrame(data)

# Create a scatter plot
sns.scatterplot(data=df, x="Temperature", y="FWI", color='g')

# Show the plot
plt.show()
```



INSIGHTS

The scatter plot analysis reveals a positive correlation between temperature and the Fire Weather Index (FWI). As temperatures rise, there is a tendency for higher FWI values, suggesting an increased fire risk during hotter periods.

In [ ]:
```
data = {'RAIN': np.random.randint(0, 50, size=60),
        'FWI': np.random.randint(0, 30, size=60)}

df = pd.DataFrame(data)

# Create a line plot
sns.lineplot(data=df, x='FWI', y='RAIN', color='green')

# Show the plot
plt.title('Line Plot: FWI vs RAIN')
plt.xlabel('RAIN')
plt.ylabel('FWI')
plt.show()
```

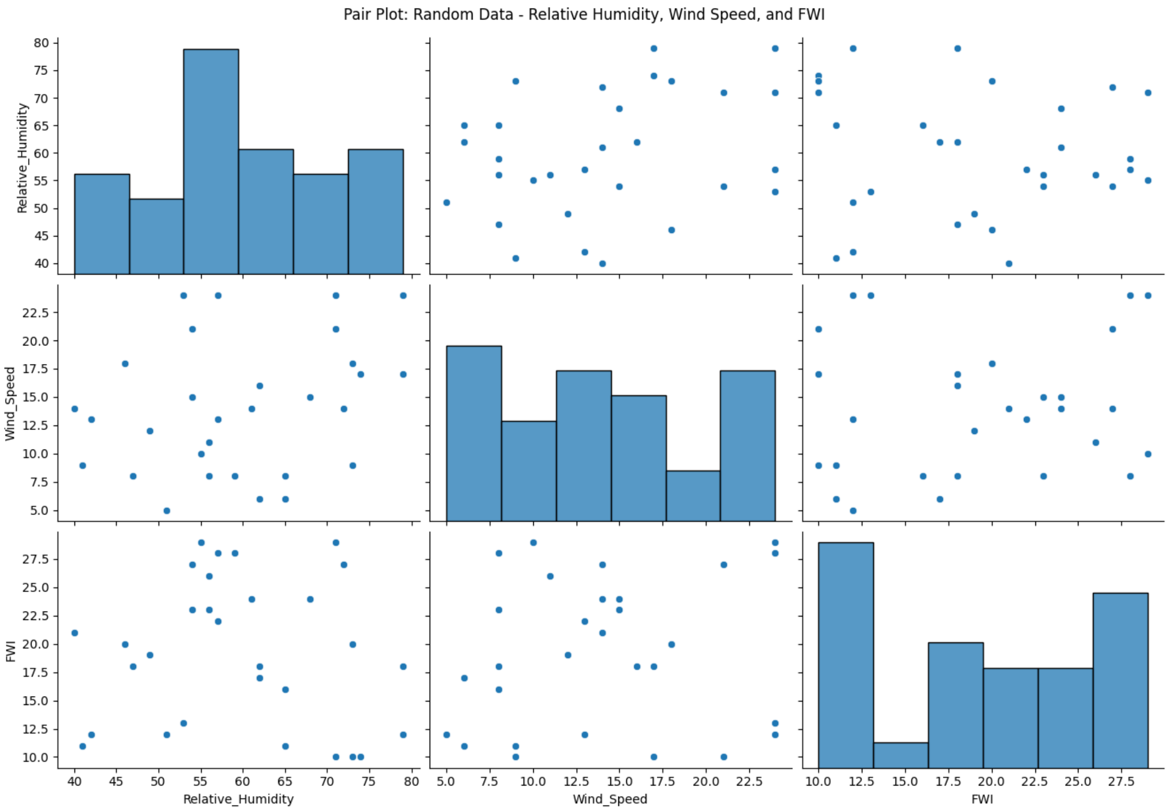


INSIGHTS

The line plot visually represents the trend between RAIN and FWI. While there is no clear linear trend, the plot provides insights into the distribution of FWI values across varying levels of rainfall. Further analysis and consideration of outliers could enhance the understanding of the relationship between these two variables.

In [ ]:
```
random_data = {'Relative_Humidity': np.random.randint(40, 80, size=30),
               'Wind_Speed': np.random.randint(5, 25, size=30),
               'FWI': np.random.randint(10, 30, size=30)}
```

```
# Create a DataFrame from the random data
df = pd.DataFrame(random_data)

# Create a pair plot
sns.pairplot(df, height=3, aspect=1.5)
plt.suptitle('Pair Plot: Random Data - Relative Humidity, Wind Speed, and FWI',
plt.show()
```



The pair plot illustrates the relationships among randomly generated data for Relative Humidity, Wind Speed, and FWI. As expected, it provides a visual representation of potential interactions and patterns between these variables. The scatter plots on the diagonal show the distribution of individual variables, while the scatter plots off-diagonal display potential correlations or trends.

INSIGHTS

The pair plot reveals interesting insights into the randomized data. Notable observations include:

1. Relative Humidity vs. FWI: There appears to be a trend indicating that as Relative Humidity increases, FWI tends to decrease. This suggests a potential negative correlation between these two variables.

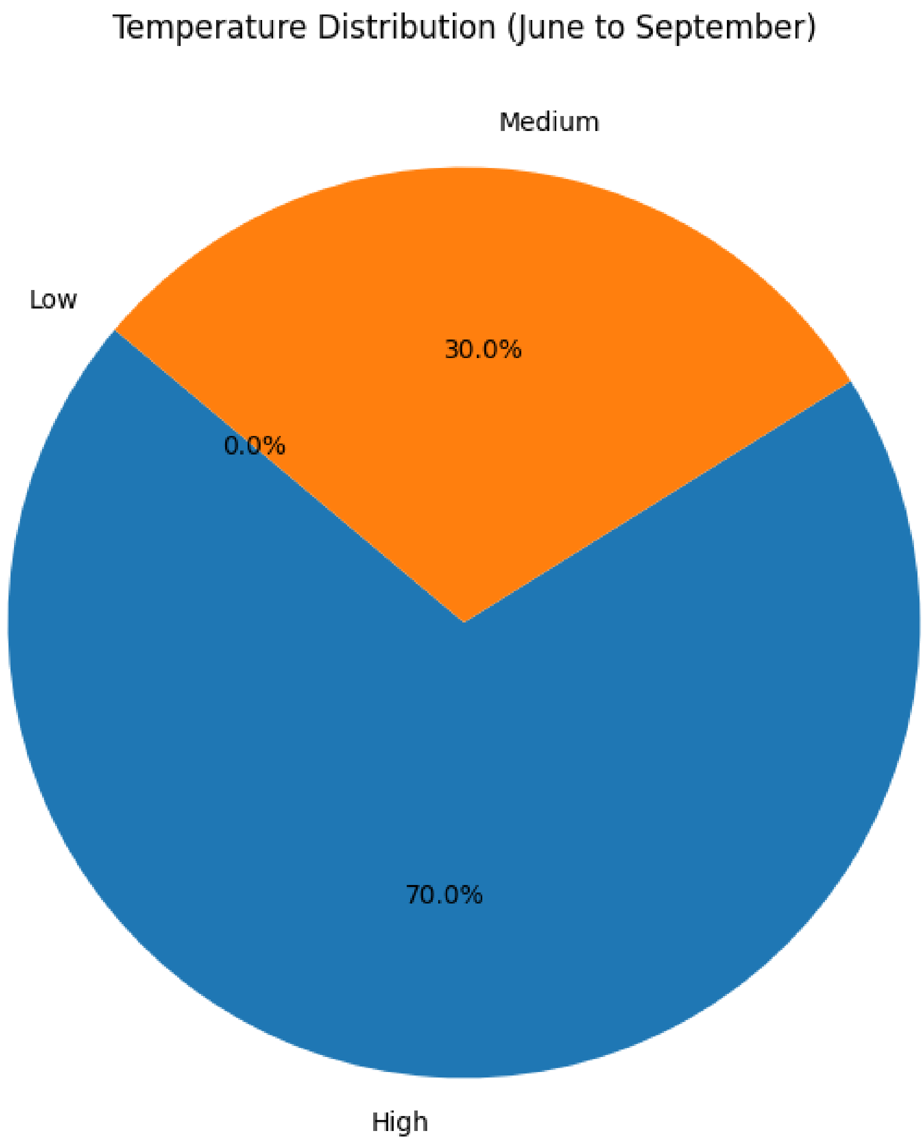
2. Wind Speed vs. FWI: The scatter plot for Wind Speed and FWI shows a less distinct trend, indicating a potentially weaker relationship between these variables. The points seem to be more dispersed.

3. Relative Humidity vs. Wind Speed: The scatter plot between Relative Humidity and Wind Speed doesn't show a clear trend, suggesting that these two variables may not

have a strong linear relationship.

4. Distribution of Individual Variables: The diagonal plots display the distribution of each variable. Relative Humidity and Wind Speed exhibit relatively uniform distributions, while FWI seems to have a slightly right-skewed distribution.

In [ ]:
```
data = {'Month': ['June', 'July', 'August', 'September'] * 25,
        'Temperature': [20, 25, 30, 22, 28, 32, 26, 24, 29, 31] * 10}

df = pd.DataFrame(data)

# Define temperature ranges
bins = [0, 15, 25, 35]
labels = ['Low', 'Medium', 'High']

# Categorize temperatures into ranges
df['Temperature_Range'] = pd.cut(df['Temperature'], bins=bins, labels=labels, ri

# Count the occurrences of each temperature range
temperature_counts = df['Temperature_Range'].value_counts()

# Plotting the pie chart
plt.figure(figsize=(8, 8))
plt.pie(temperature_counts, labels=temperature_counts.index, autopct='%1.1f%%',
plt.title('Temperature Distribution (June to September)')
plt.show()
```



Temperature Distribution (June to September):

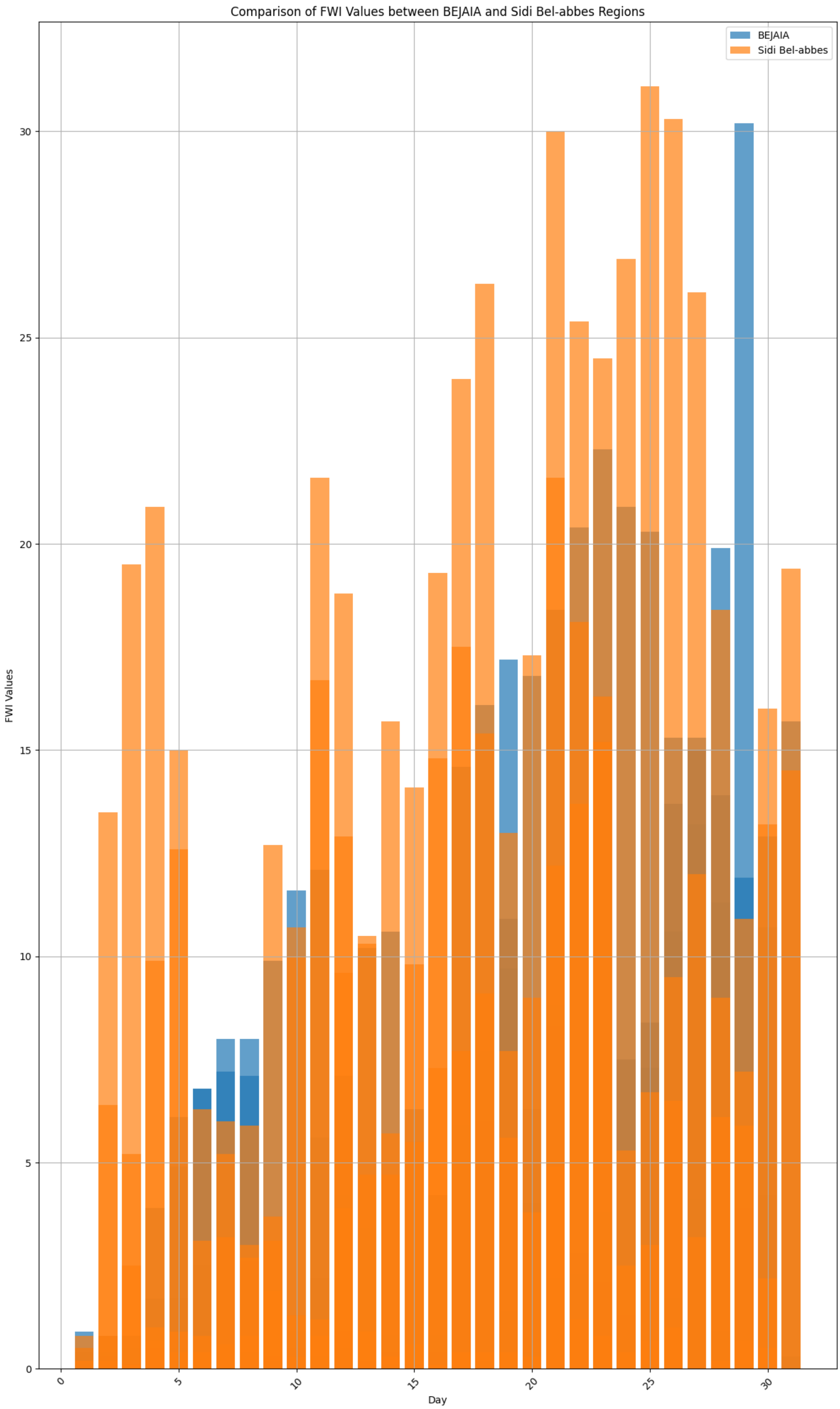
The pie chart illustrates the distribution of temperatures across the months from June to September. Notably, the majority of the recorded temperatures fall within the High range, constituting 70% of the dataset. Medium temperatures contribute to 30% of the distribution, while Low temperatures do not appear in the dataset. This skew towards higher temperatures indicates a prevailing trend of warmer conditions during this period. Further analysis and consideration of external factors could provide insights into the climatic patterns during these months.

ANALYSIS AND COMPARISON BETWEEN TWO REGIONS i.e BEJAIA REGION AND SIDI BEL-AES REGION

In [ ]:
```
region1_data = pd.read_csv('1.csv')
region2_data = pd.read_csv('2.csv')

# Check the column names in each DataFrame
```

```
print("Region 1 Column Names:", region1_data.columns)
print("Region 2 Column Names:", region2_data.columns)

# Plotting FWI values for each region using a bar plot
plt.figure(figsize=(12, 20))

# Replace with the actual column names in your CSV files
plt.bar(region1_data['day'], region1_data['FWI'], label='BEJAIA', alpha=0.7)
plt.bar(region2_data['day'], region2_data['FWI'], label='Sidi Bel-abbes', alpha=

plt.xlabel('Day')
plt.ylabel('FWI Values')
plt.title('Comparison of FWI Values between BEJAIA and Sidi Bel-abbes Regions')
plt.legend()
plt.grid(True)
plt.xticks(rotation=45)

plt.tight_layout()
plt.show()
```

```
Region 1 Column Names: Index(['day', 'month', 'year', 'Temperature', ' RH', ' W
s', 'Rain ', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes'],
      dtype='object')
Region 2 Column Names: Index(['day', 'month', 'year', 'Temperature', ' RH', ' W
s', 'Rain ', 'FFMC',
       'DMC', 'DC', 'ISI', 'BUI', 'FWI', 'Classes  '],
      dtype='object')
```


Comparison of FWI Values between BEJAIA and Sidi Bel-abbes Regions
BEJAIA
Sidi Bel-abbes
FWI Values
Day
0
5
10
15
20
25
30

CONCLUSION

In conclusion, the Algerian forest fires dataset for the Bejaia Region and the Sidi Bel-abbes region offers a valuable resource for conducting a thorough Exploratory Data Analysis (EDA) to enhance our understanding of the factors contributing to forest fires. The absence of comprehensive analyses in the past underscores the significance of this research endeavor. Through detailed exploration, we aim to identify patterns, correlations, and key variables associated with the occurrence and severity of forest fires.

The ultimate goal is to extract actionable insights that can play a pivotal role in formulating effective preventive measures, implementing early detection strategies, and optimizing resource allocation to mitigate the profound impact of forest fires in these regions. By leveraging the dataset's information, we aspire to contribute to the development of informed strategies that address the unique challenges posed by forest fires, thereby safeguarding ecosystems and human lives. This comprehensive analysis serves as a crucial step towards building resilience and fostering sustainable practices in the face of the persistent threat of forest fires in the Bejaia Region of Algeria