Luke Scott

COSC 311

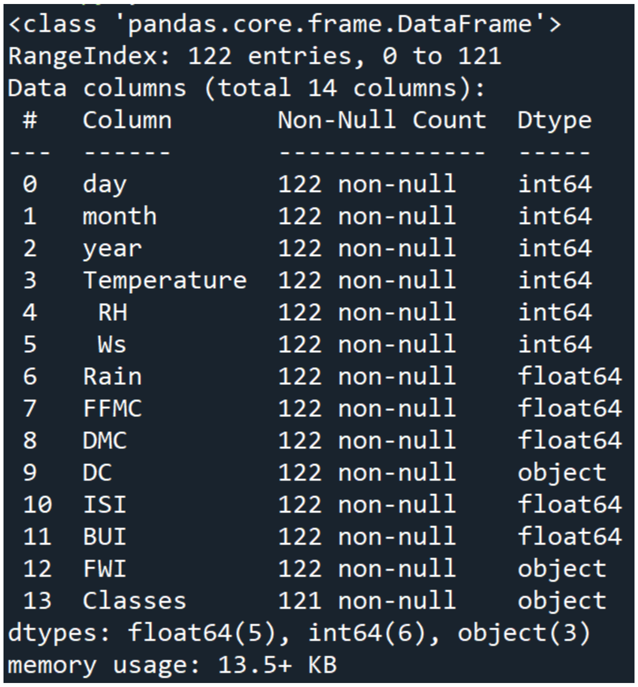
Lab 2

Dr. Wang

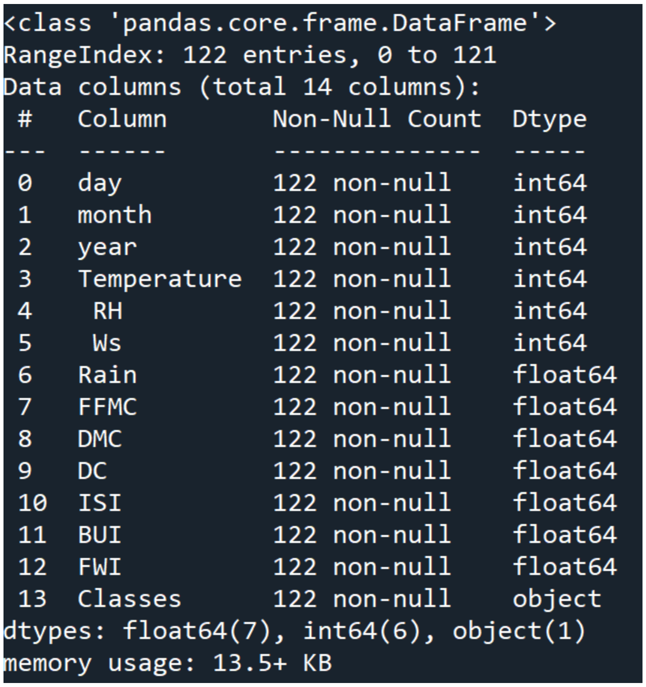
Lab 2 Lab Report and Source Code

Task 1:

.info()

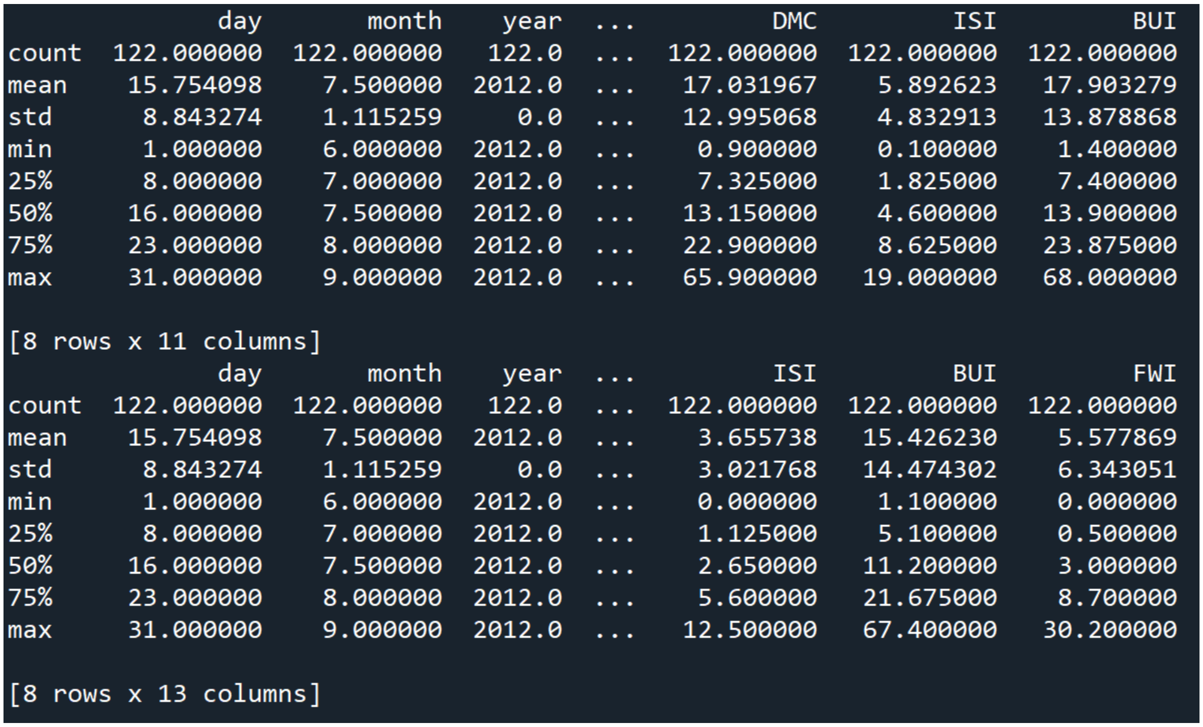


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122 entries, 0 to 121
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   day          122 non-null    int64
 1   month        122 non-null    int64
 2   year         122 non-null    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    object
 13  Classes      121 non-null    object
dtypes: float64(5), int64(6), object(3)
memory usage: 13.5+ KB
```



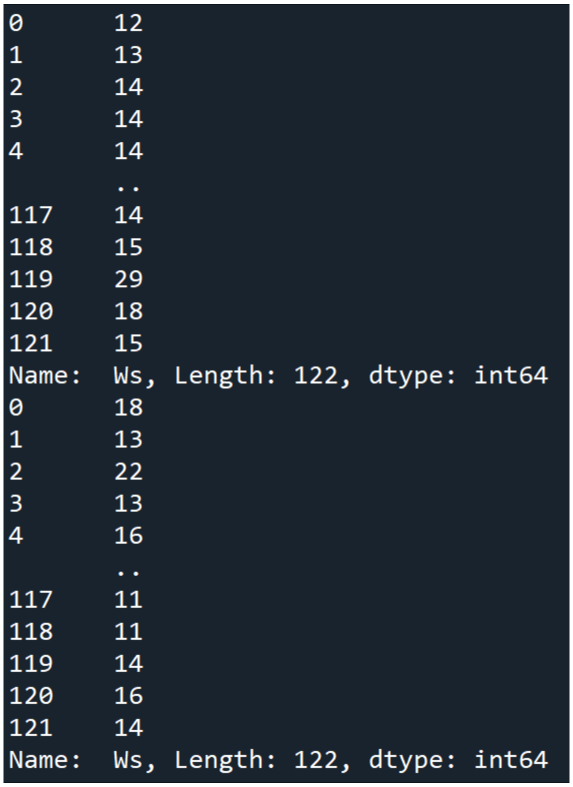
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 122 entries, 0 to 121
Data columns (total 14 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   day          122 non-null    int64
 1   month        122 non-null    int64
 2   year         122 non-null    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
```

.describe()



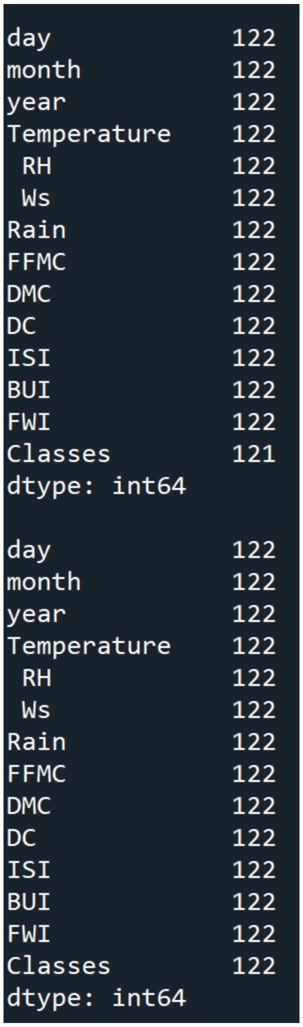
```
              day       month    year  ...         DMC         ISI         BUI
count  122.000000  122.000000   122.0  ...  122.000000  122.000000  122.000000
mean    15.754098    7.500000  2012.0  ...   17.031967    5.892623   17.903279
std      8.843274    1.115259     0.0  ...   12.995068    4.832913   13.878868
min      1.000000    6.000000  2012.0  ...    0.900000    0.100000    1.400000
25%      8.000000    7.000000  2012.0  ...    7.325000    1.825000    7.400000
50%     16.000000    7.500000  2012.0  ...   13.150000    4.600000   13.900000
75%     23.000000    8.000000  2012.0  ...   22.900000    8.625000   23.875000
max     31.000000    9.000000  2012.0  ...   65.900000   19.000000   68.000000

[8 rows x 11 columns]
              day       month    year  ...         ISI         BUI         FWI
count  122.000000  122.000000   122.0  ...  122.000000  122.000000  122.000000
mean    15.754098    7.500000  2012.0  ...    3.655738   15.426230    5.577869
std      8.843274    1.115259     0.0  ...    3.021768   14.474302    6.343051
min      1.000000    6.000000  2012.0  ...    0.000000    1.100000    0.000000
25%      8.000000    7.000000  2012.0  ...    1.125000    5.100000    0.500000
50%     16.000000    7.500000  2012.0  ...    2.650000   11.200000    3.000000
75%     23.000000    8.000000  2012.0  ...    5.600000   21.675000    8.700000
max     31.000000    9.000000  2012.0  ...   12.500000   67.400000   30.200000

[8 rows x 13 columns]
```

[‘ Ws’]



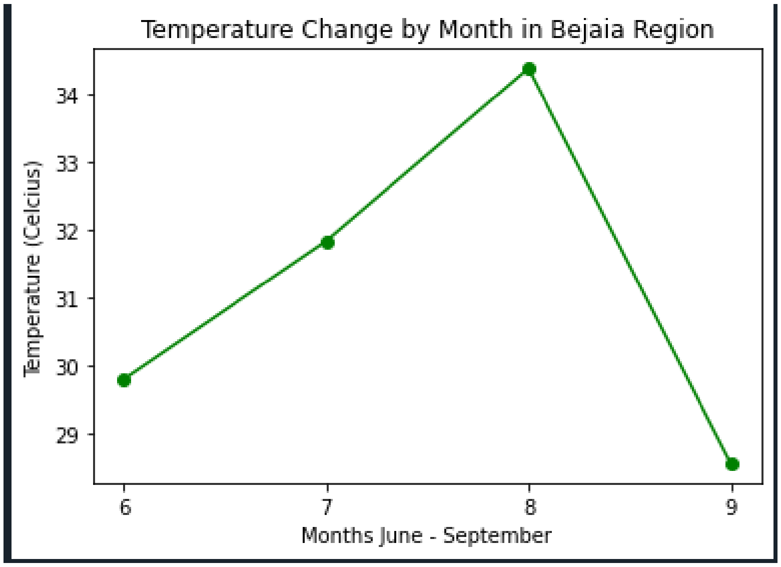
```
0       12
1       13
2       14
3       14
4       14
        ..
117     14
118     15
119     29
120     18
121     15
Name:   Ws, Length: 122, dtype: int64
0       18
1       13
2       22
3       13
4       16
        ..
117     11
118     11
119     14
120     16
121     14
Name:   Ws, Length: 122, dtype: int64
```

.count()

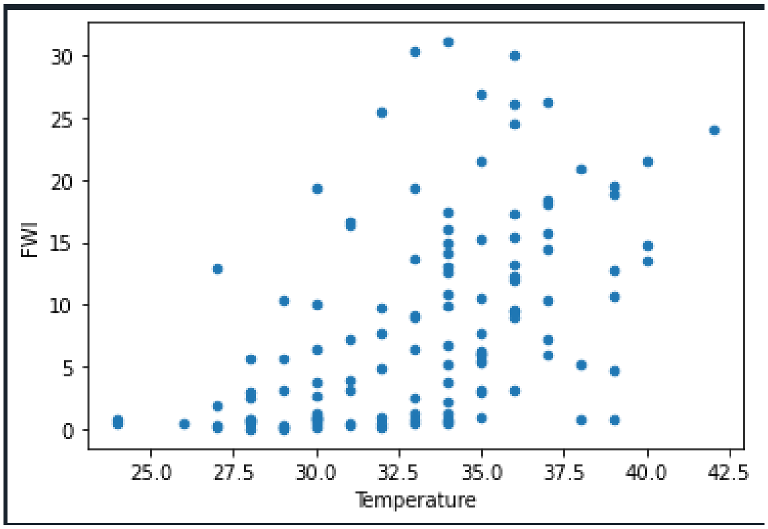


```
day            122
month          122
year           122
Temperature    122
 RH            122
 Ws            122
Rain           122
FFMC           122
DMC            122
DC             122
ISI            122
BUI            122
FWI            122
Classes        121
dtype: int64

day            122
month          122
year           122
Temperature    122
 RH            122
 Ws            122
Rain           122
FFMC           122
DMC            122
DC             122
ISI            122
BUI            122
FWI            122
Classes        122
dtype: int64
```

Task 2:

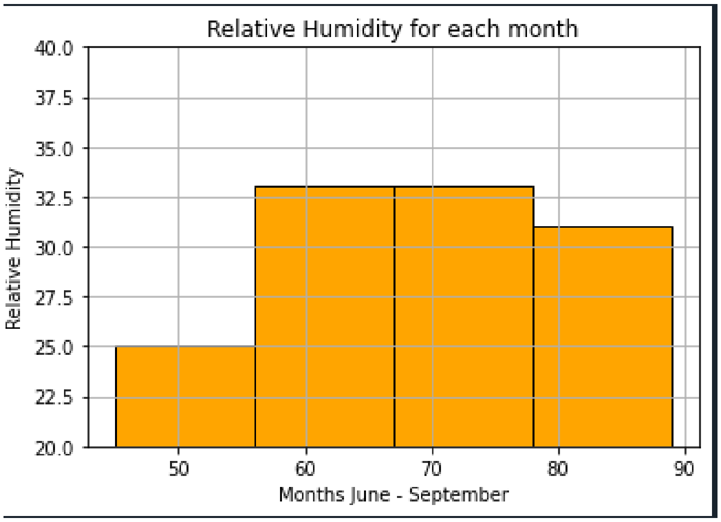


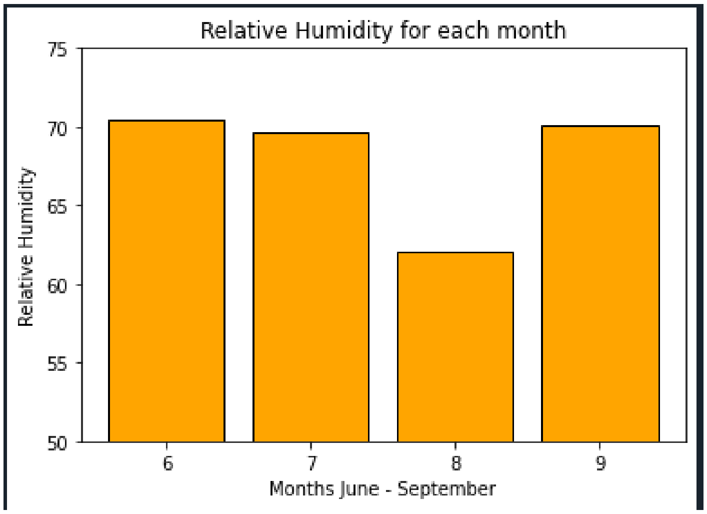
Task 3:



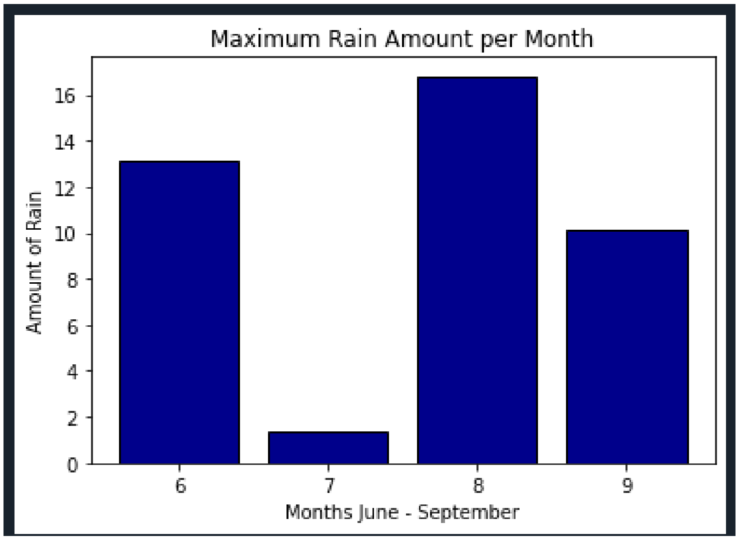
Task 4:

--- NOTE : I was experimenting with matplotlib and wasn't happy with the histogram result for this task. I included a bar graph of the data as well, this is commented out from lines 52-61

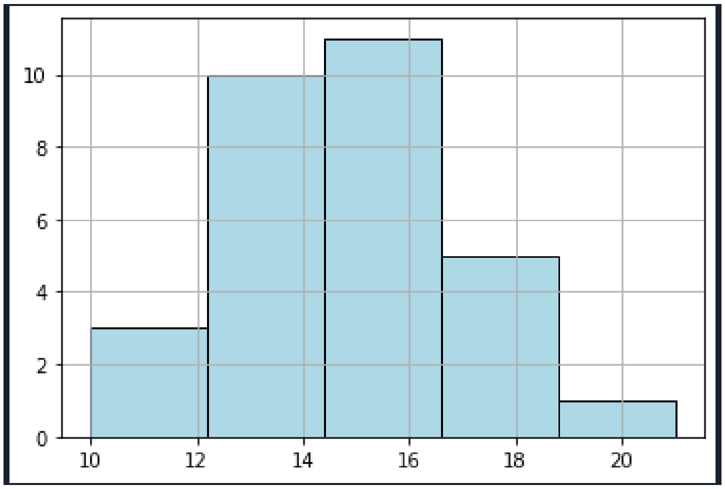



Relative Humidity for each month
Relative Humidity
75
70
65
60
55
50
6
7
8
9
Months June - September

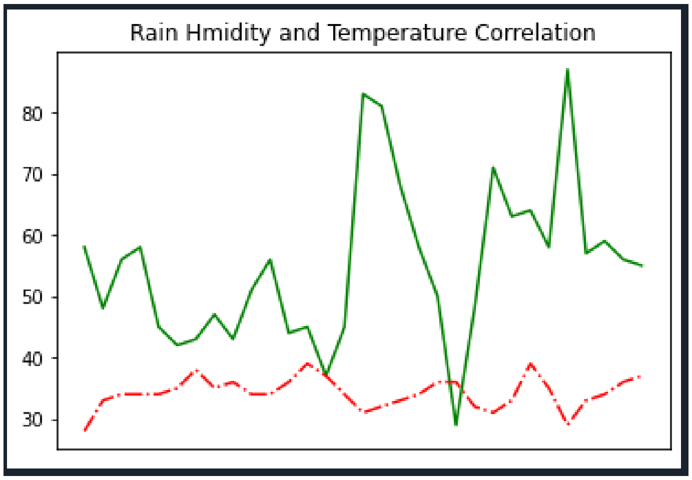
Task 5:


Maximum Rain Amount per Month
Amount of Rain
16
14
12
10
8
6
4
2
0
6
7
8
9
Months June - September

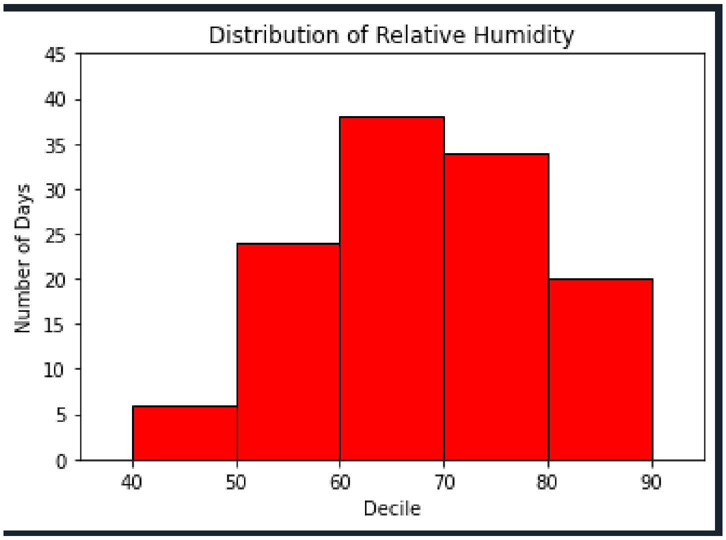
Task 6:


10
8
6
4
2
0
10
12
14
16
18
20

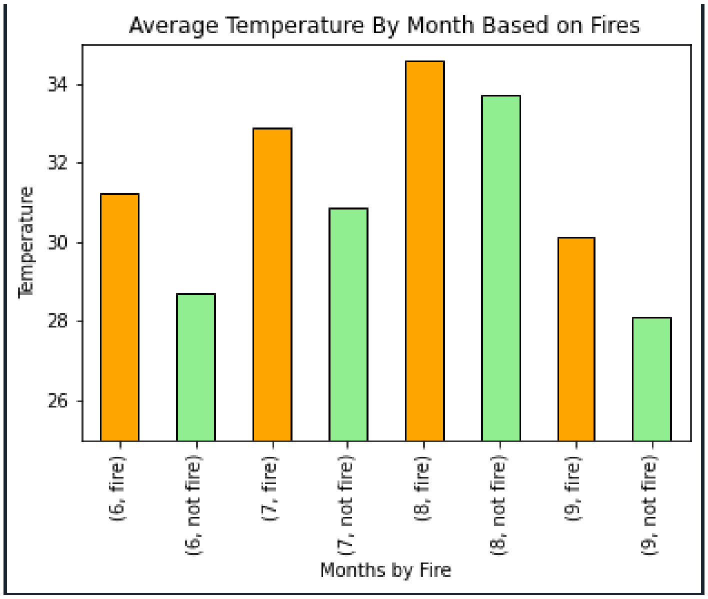
Task 7:



Task 8:



Task 9:



Source Code:

```
import pandas as pd

from matplotlib import pyplot as plt

from collections import Counter


SBA_Region = pd.read_csv('Sidi-Bel_Abbes_Region.csv')

B_Region = pd.read_csv('Bejaia_Region.csv')


# Task 1 : Show Info(), Describe(), 'Ws', count() of both data sets


SBA_Region.info()

B_Region.info()


B_DF = pd.DataFrame(B_Region)

SBA_DF = pd.DataFrame(SBA_Region)


print(SBA_Region.describe(), '\n')
print(B_Region.describe(), '\n')

print(SBA_Region[' Ws'], '\n')
print(B_Region[' Ws'], '\n')

print(SBA_Region.count(), '\n')
print(B_Region.count(), '\n')

# Gathers the mean of all temperatures based on month
temperature_mean = B_Region.groupby('month')[['Temperature']].mean().sort_values('month')

# Creates a list of integers referencing months from the data set
months = Counter(B_Region['month'])

# Task 2 = Show line graph of average temperature change by month
plt.plot(months.keys(), temperature_mean, color='green', marker='o', linestyle='solid')
plt.title('Temperature Change by Month in Bejaia Region')
plt.ylabel('Temperature (Celcius)')
plt.xlabel('Months June - September')
plt.xticks([6, 7, 8, 9])
plt.show()

# Task 3 = Create scatter plot for temperature based on Fire Weather Index
SBA_Region.plot.scatter(x = 'Temperature', y = 'FWI')
plt.show()

# Task 4 = Draw a historgram of average Relative Humidity by month
B_DF[' RH'].plot(kind = 'hist', bins = 4, color = 'orange', edgecolor = 'black')
```

```
plt.grid(visible = True)
plt.title('Relative Humidity for each month')
plt.xlabel('Months June - September')
plt.ylabel('Relative Humidity')
plt.ylim(20, 40)
plt.show()

'''
b_rh = B_DF.groupby('month')[' RH'].mean()
plt.bar(b_rh.keys(), b_rh, color = 'orange', edgecolor = 'black')
plt.title('Relative Humidity for each month')
plt.xlabel('Months June - September')
plt.ylabel('Relative Humidity')
plt.xticks([6,7,8,9])
plt.ylim(50, 75)
plt.show()
'''

# Task 5 = Draw a bar graph to show maximum rain amount for each month
b_max_rain = B_DF.groupby('month')['Rain '].max()
plt.bar(b_max_rain.keys(), b_max_rain, color = 'darkblue', edgecolor = 'black')
plt.title('Maximum Rain Amount per Month')
plt.xlabel('Months June - September')
plt.ylabel('Amount of Rain')
plt.xticks([6, 7, 8, 9])
plt.show()

# Task 6 = Draw a histogram of Windspeed in the month of June
```

```
(SBA_DF[' Ws'][SBA_DF['month'] == 6]).hist(bins = 5, color = 'lightblue', edgecolor = 'black')
plt.show()


# Task 7 = Draw a line figure that shows correlation between Temperature and Relative Humidity
SBA_Temp = (SBA_DF[' RH'][SBA_DF['month'] == 7])
SBA_RH = (SBA_DF['Temperature'][SBA_DF['month'] == 7])
xs = [i for i, _ in enumerate(SBA_Temp)]


plt.plot(xs, SBA_Temp,  'g-',   label = 'Temperature')
plt.plot(xs, SBA_RH,    'r-.',  label = 'Rain Humidity')


plt.xticks([])
plt.title('Rain Hmidity and Temperature Correlation')
plt.show()


# Task 8 = Draw a bar graph to show the distribution of Relative Humidity
values = Counter(min(humid // 10 * 10, 90) for humid in B_DF[' RH'])


plt.bar([x + 5 for x in values.keys()], values.values(), 10, edgecolor = 'black', color = 'red')
plt.axis([35, 95, 0, 45])
plt.xticks([10 * i for i in range(4, 10)])
plt.xlabel('Decile')
plt.ylabel('Number of Days')
plt.title('Distribution of Relative Humidity')
plt.show()


# Task 9 = Draw a figure to show average temperature for each month with fires and no fires
fire = B_DF.groupby(['month', 'Classes  '])['Temperature'].mean()
```

```
fire.plot(x = 'month', y = 'Temperature', kind = 'bar',\
        color = ['orange','lightgreen'], edgecolor = 'black', ylim=(25,35),\
        xlabel = 'Months by Fire', ylabel = 'Temperature', title = 'Average Temperature By Month Based on Fires')
```