2017

HOMEWORK 4

GUIDE TO ENGINEERING DATA SCIENCE

OLABODE ALAMU 1498663

Table of Contents

PROBLEM

Purpose: To learn how to **treat and Impute Missing observations**.

Go to the Blackboard and download the file "HW4.xlsx". This file contains the Airquality data available in R, it has missing records. Use this file to complete the following tasks.

a) Plot the data contained in the file with appropriate legends.

b) Perhaps, a more helpful visual representation is needed. Make a histogram showing the number of missing observations. Analyze your plot.

c) **Treat the missing observations** in the variables by **deleting** the observations that have the missing values. Plot the data that displays this treatment of the data.

d) **Treat the missing observations** in the variables by **replacing the missing observations with the mean imputation technique** for the given variables. Plot the data that displays this treatment of the data.

Submission: You will have the choice of using **Python or RStudio**. **All** scripts and PDF file should be submitted **together** in ZIP file.

SOLUTION

This assignment was completed using Python 3x, this document containsthe results, the accompanying code can be found in the .py file in the zip folder.

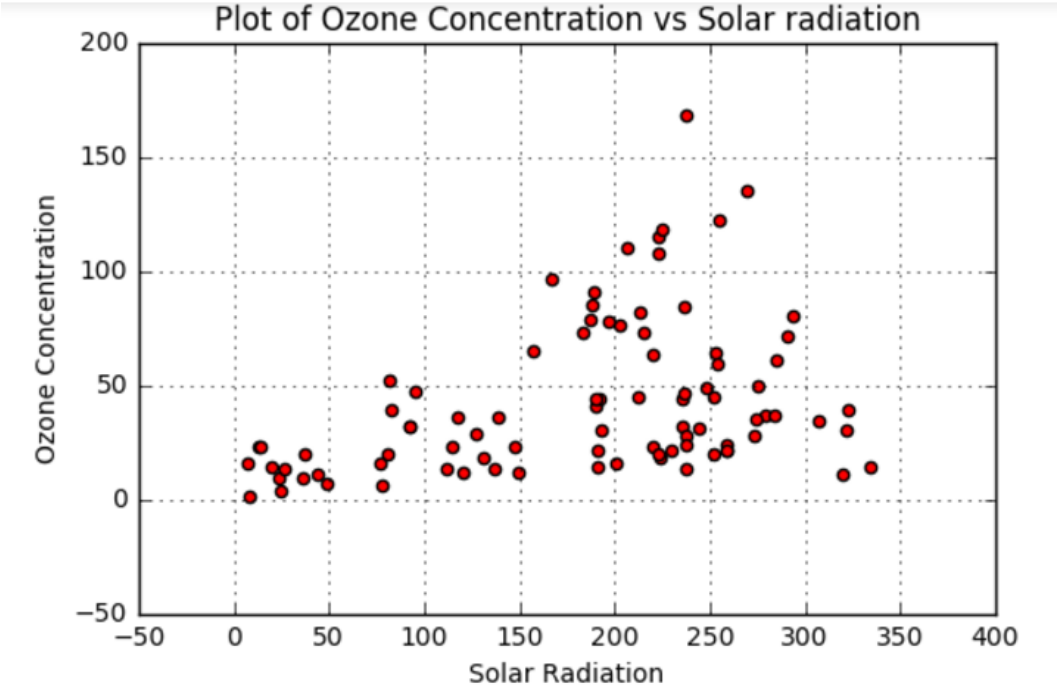
```
In [141]: Air_quality = pd.read_excel('HW4.xlsx') # Reads the excel spreadsheet which contains the dataset
```

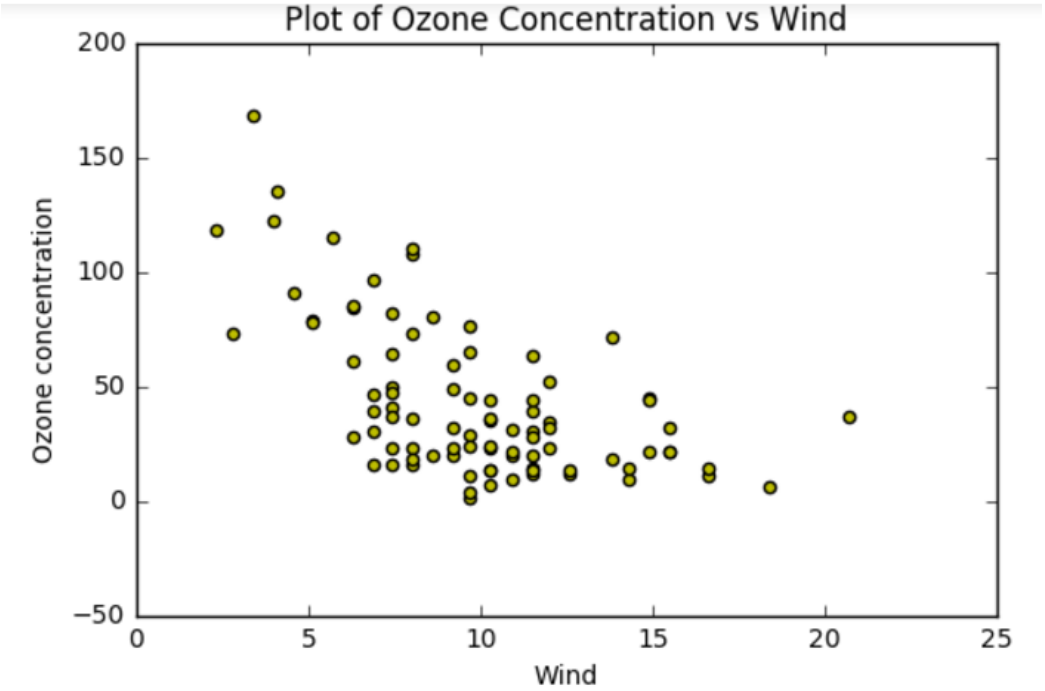
```
In [142]: Air_quality
```

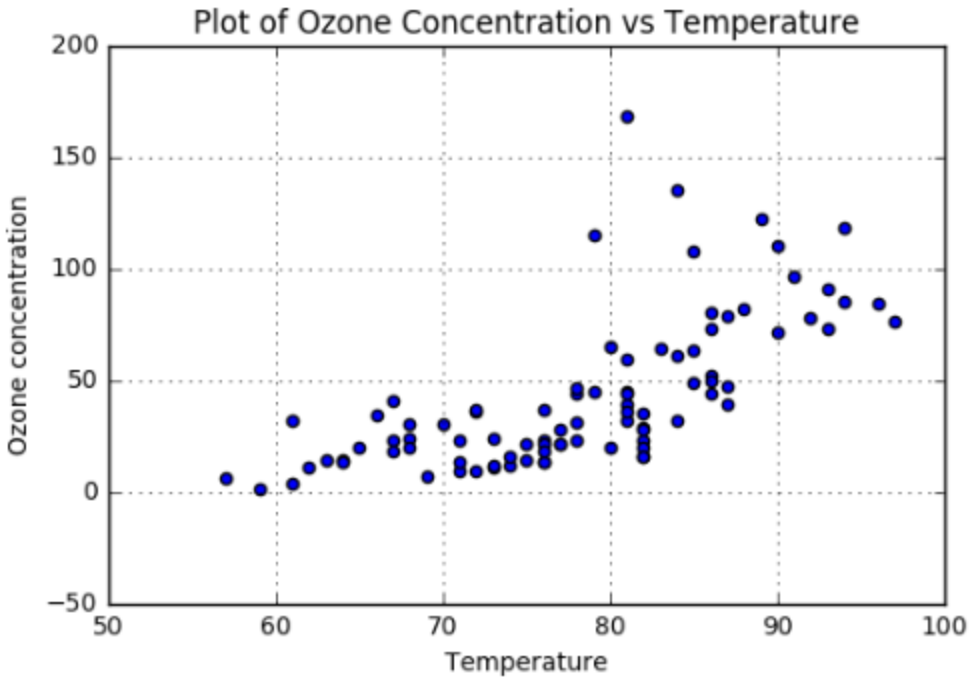
Out[142]:

	Ozone	Solar.R	Wind	Temp
1	41.0	190.0	7.4	67.0
2	36.0	118.0	8.0	72.0
3	12.0	149.0	12.6	74.0
4	18.0	313.0	NaN	62.0
5	NaN	NaN	NaN	56.0
6	28.0	NaN	NaN	NaN
7	23.0	299.0	NaN	NaN
8	19.0	99.0	NaN	59.0
9	8.0	19.0	NaN	61.0
10	NaN	194.0	NaN	NaN
11	7.0	NaN	6.9	NaN
12	16.0	256.0	9.7	NaN
13	11.0	290.0	9.2	NaN
14	14.0	274.0	10.9	NaN
15	18.0	65.0	13.2	NaN

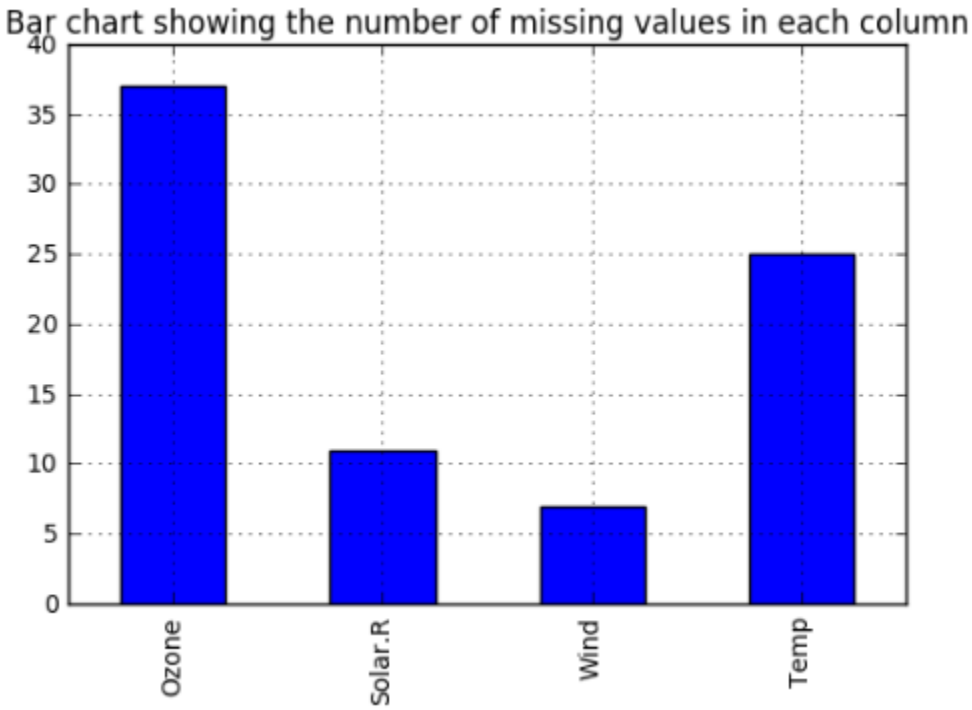
a) The data points in the excel file were paired up and plotted. The plots can be seen below:







b) Histogram of missing values.

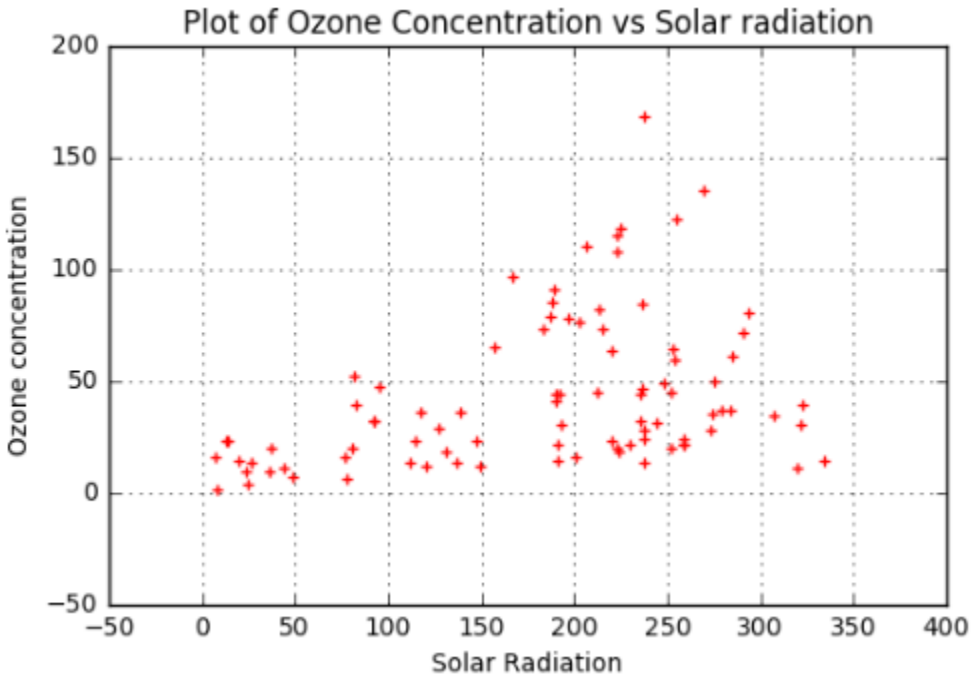


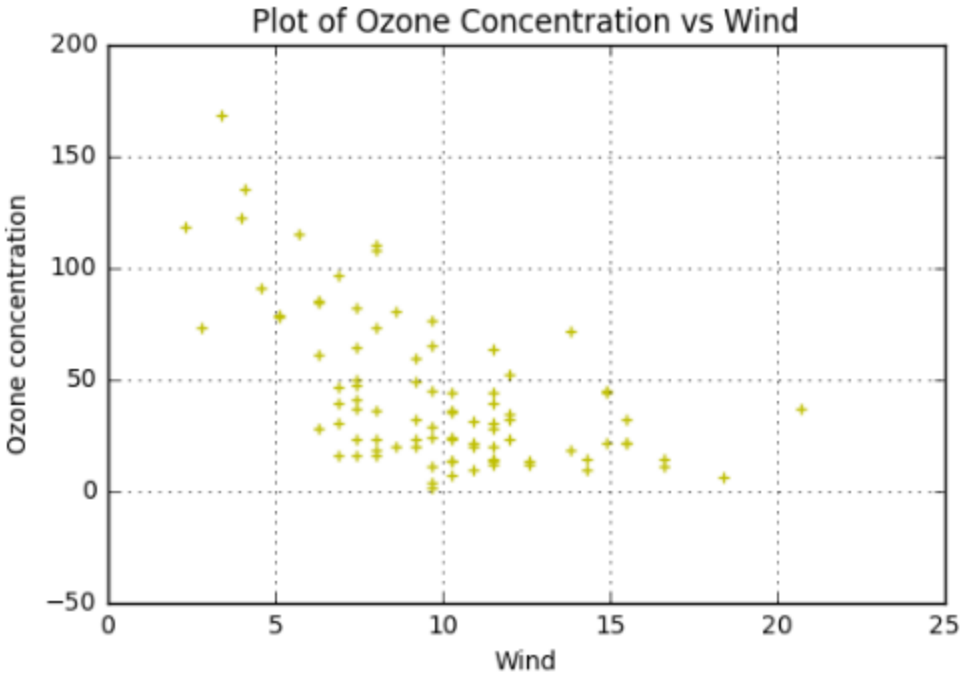
c) The rows with the missing values were removed and the charts plotted again as shown below:

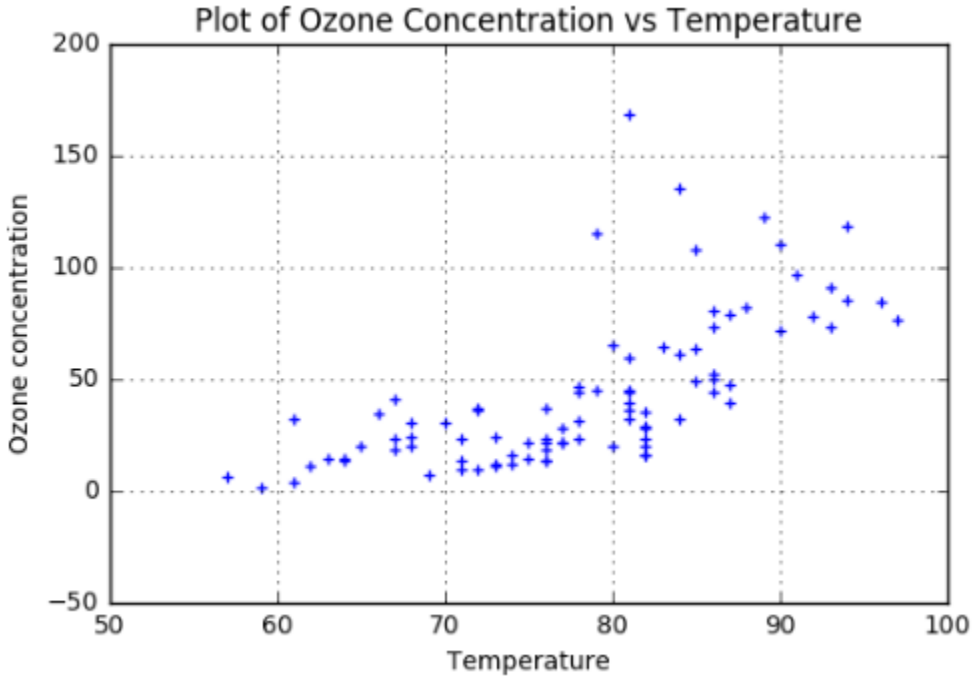
```
In [167]: Air_quality = Air_quality.dropna() # drops all the rows which have a missing value

In [168]: Air_quality
```

Out[168]:

	Ozone	Solar.R	Wind	Temp
1	41.0	190.0	7.4	67.0
2	36.0	118.0	8.0	72.0
3	12.0	149.0	12.6	74.0
16	14.0	334.0	11.5	64.0
17	34.0	307.0	12.0	66.0
18	6.0	78.0	18.4	57.0
19	30.0	322.0	11.5	68.0
20	11.0	44.0	9.7	62.0
21	1.0	8.0	9.7	59.0
22	11.0	320.0	16.6	73.0
23	4.0	25.0	9.7	61.0
24	32.0	92.0	12.0	61.0




Plot of Ozone Concentration vs Wind
Ozone concentration
Wind
200
150
100
50
0
−50
0
5
10
15
20
25


Plot of Ozone Concentration vs Temperature
Ozone concentration
Temperature
200
150
100
50
0
−50
50
60
70
80
90
100

d) The missing values were replaced with the mean of each column, but first, the excel file was inputted and named 'Air'.

```
In [173]: # Import the new dataset
          Air = pd.read_excel('HW4.xlsx')
```

```
In [209]: Air.head() #Shows only the top 5 values
```

Out[209]:

	Ozone	Solar.R	Wind	Temp
1	41.0	190.0	7.4	67.0
2	36.0	118.0	8.0	72.0
3	12.0	149.0	12.6	74.0
4	18.0	313.0	NaN	62.0
5	NaN	NaN	NaN	56.0

The mean of each column was calculated

```
In [210]: # Calculate the mean values for each column
          Ozone_mean = Air['Ozone'].mean()
          Solar_mean = Air['Solar.R'].mean()
          Wind_mean = Air['Wind'].mean()
          Temp_mean = Air['Temp'].mean()
```

```
In [211]: Ozone_mean
Out[211]: 42.12931034482759

In [212]: Solar_mean
Out[212]: 185.95070422535213

In [213]: Wind_mean
Out[213]: 9.806164383561642

In [214]: Temp_mean
Out[214]: 77.859375
```

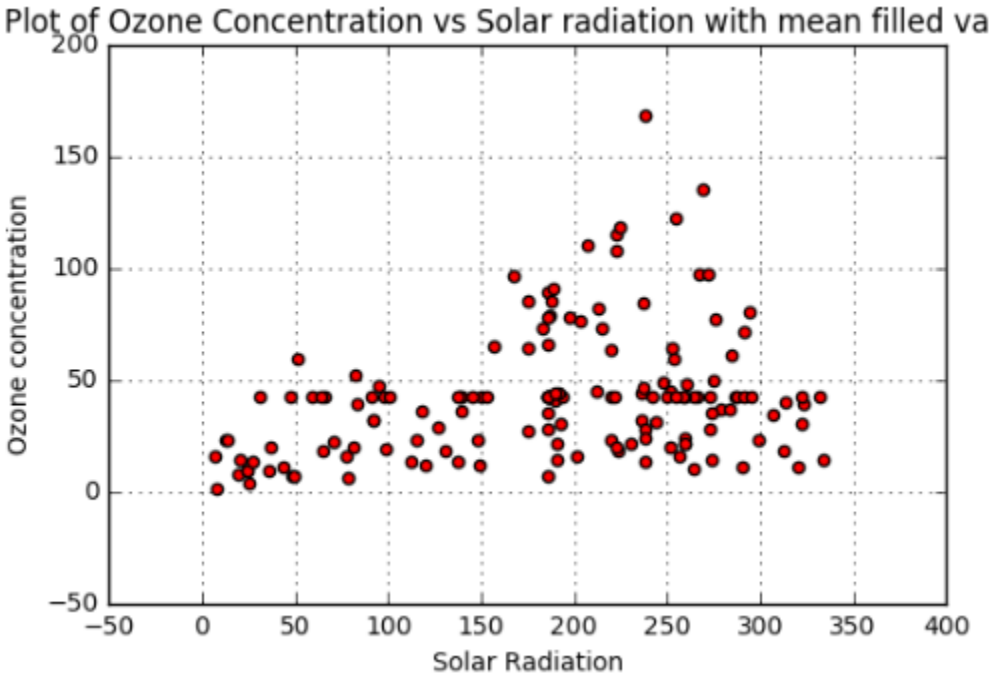
Next, the mean value of each column was used to fill the NaN values by creating a dictionary with the column name and mean value pairs, this dictionary was passed as the value argument for the fillna() method.

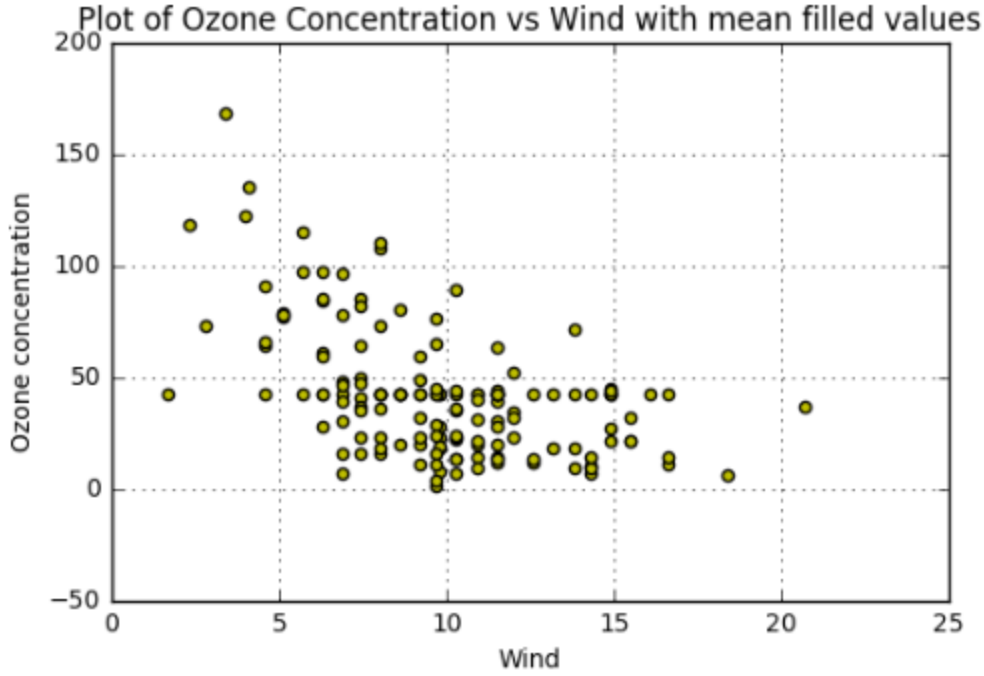
```
In [216]: Air.fillna(value = {"Ozone": Ozone_mean, 'Solar.R': Solar_mean, 'Wind': Wind_mean, 'Temp': Temp_mean }, inplace = True)
```

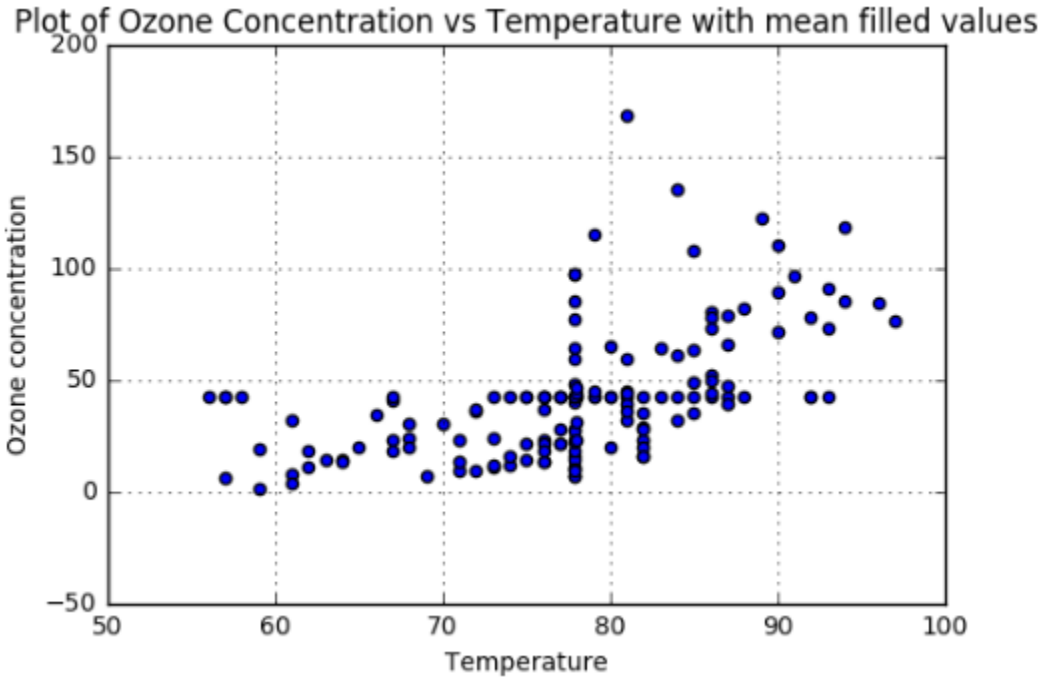
Out[216]:

	Ozone	Solar.R	Wind	Temp
1	41.00000	190.000000	7.400000	67.000000
2	36.00000	118.000000	8.000000	72.000000
3	12.00000	149.000000	12.600000	74.000000
4	18.00000	313.000000	9.806164	62.000000
5	42.12931	185.950704	9.806164	56.000000

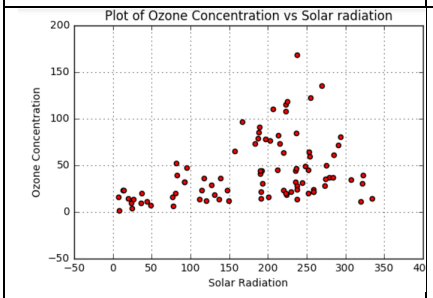
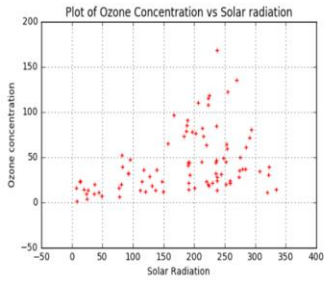
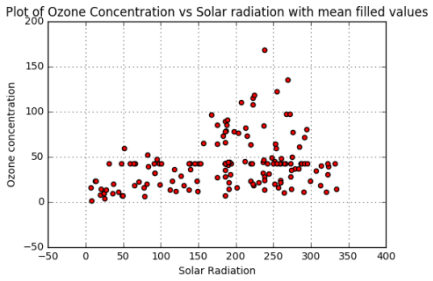
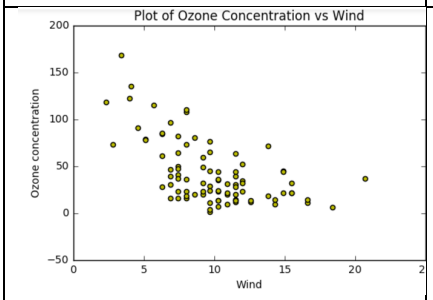
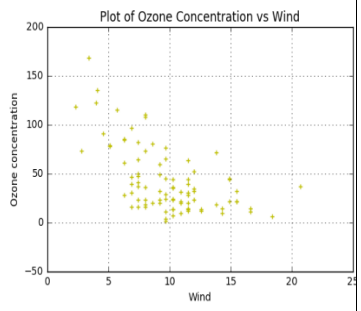
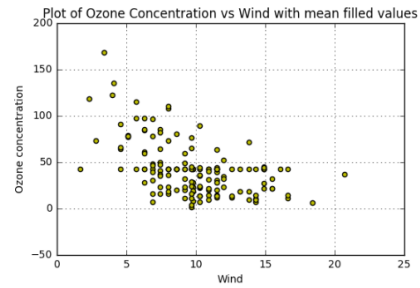
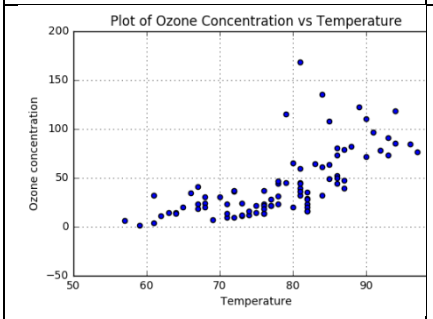
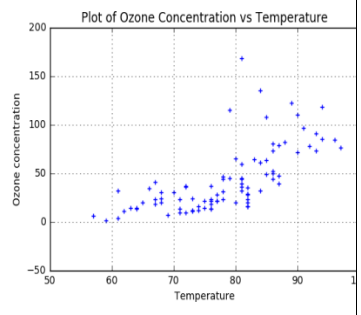
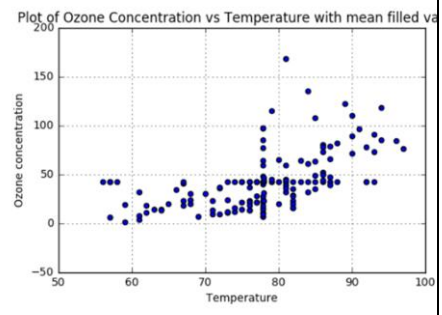
All the NaN values have been replaced with the respective means. The new plots can be found below:




Plot of Ozone Concentration vs Wind with mean filled values
Ozone concentration
Wind
200
150
100
50
0
−50
0
5
10
15
20
25


Plot of Ozone Concentration vs Temperature with mean filled values
Ozone concentration
Temperature
200
150
100
50
0
−50
50
60
70
80
90
100

COMPARISON OF THE PLOTS

PLOT WITH RAW DATA	PLOT WITH NAN VALUE REMOVED	PLOT WITH NAN VALUES REPLACED BY MEAN VALUES
		
		
		

In [222]:
```
"""
cODE WRITTEN FOR hOMEWORK 4
AUTHOR: OLABODE AFOLABI ALAMU
PEOPLESOFT ID: 1498663
GUIDE TO ENGINEERING DATA SCIENCE
27TH SEPTEMBER 2017

"""
```

Out[222]:
```
'\ncODE WRITTEN FOR hOMEWORK 4\nAUTHOR: OLABODE AFOLABI ALAMU\nPEOPLESOFT ID:
1498663\nGUIDE TO ENGINEERING DATA SCIENCE\n27TH SEPTEMBER 2017\n\n'
```

In [223]:
```
# Import the relevant libraries
import numpy as np
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt
```

In [224]:
```
Air_quality = pd.read_excel('HW4.xlsx') # Reads the excel spreadsheet which co
ntains the dataset
```

In [226]:
```
Air_quality.head()
```

Out[226]:

	Ozone	Solar.R	Wind	Temp
1	41.0	190.0	7.4	67.0
2	36.0	118.0	8.0	72.0
3	12.0	149.0	12.6	74.0
4	18.0	313.0	NaN	62.0
5	NaN	NaN	NaN	56.0

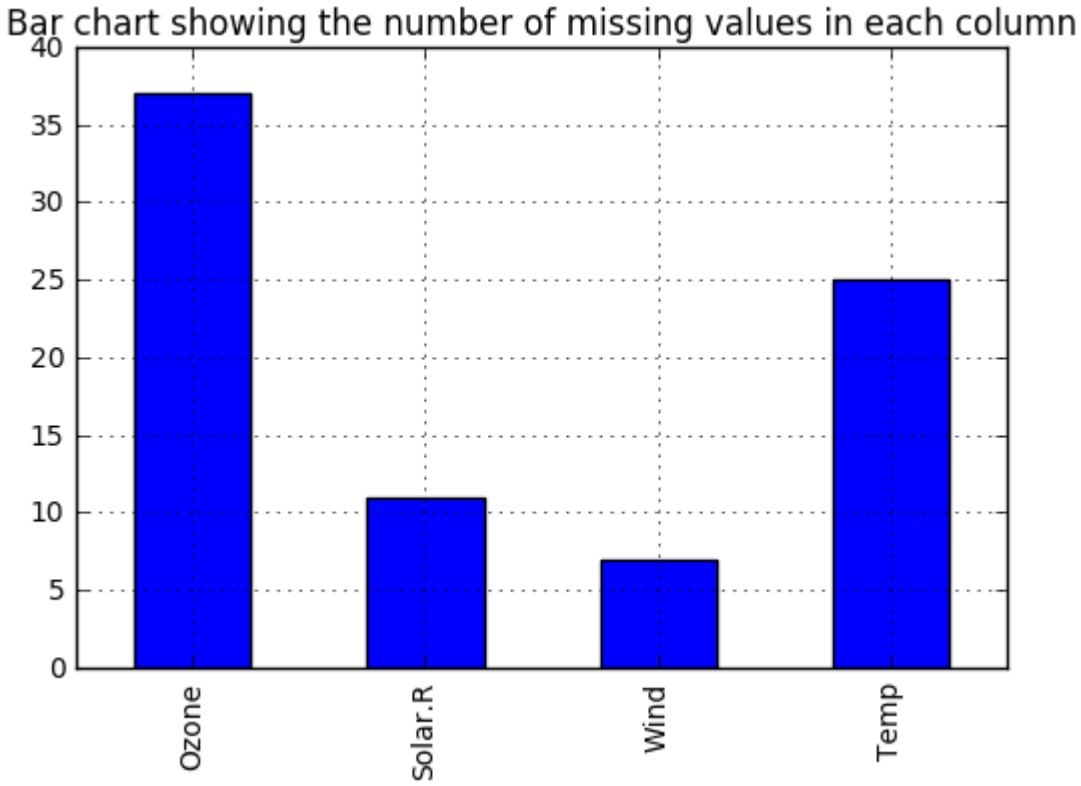
In [227]:
```
NanValue = Air_quality.isnull().sum() # COmputes the number of rows that have
 a nan value
```

In [228]:
```
dfNanValue = pd.DataFrame(data=NanValue, columns=[1])
```

In [229]:
```
dfNanValue[1] # SHows the number of nan values per column
```

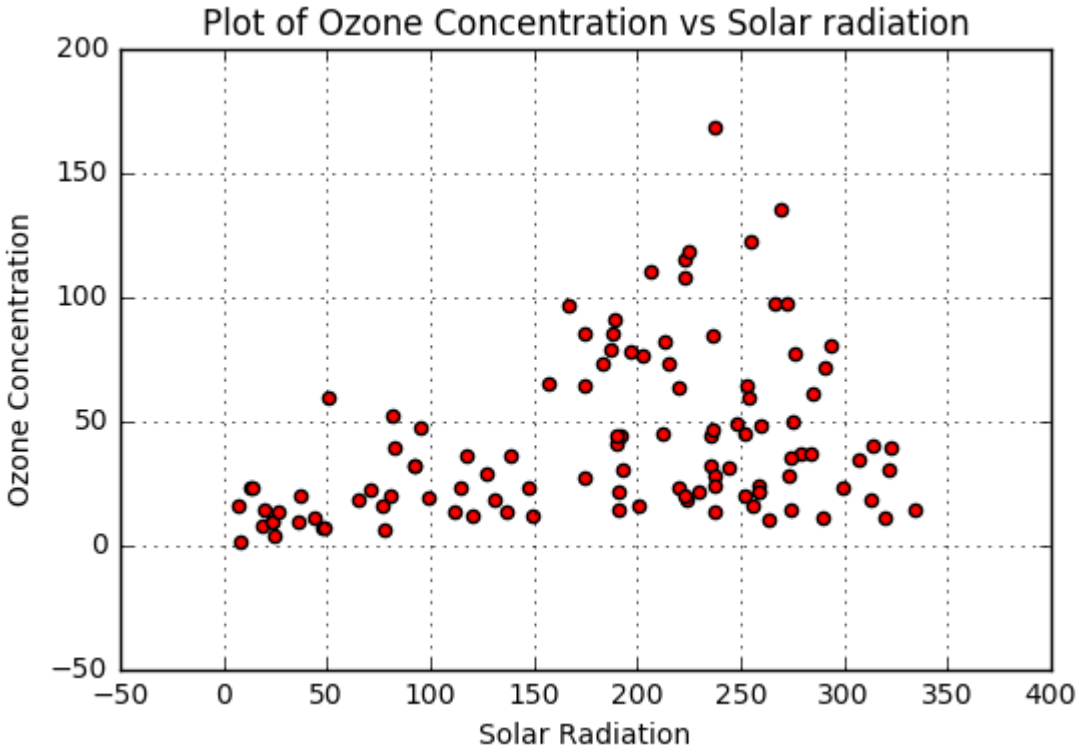
Out[229]:
```
Ozone      37
Solar.R    11
Wind        7
Temp       25
Name: 1, dtype: int64
```

In [230]:
```
dfNanValue[1].plot(kind = 'bar')    # Creates a bar chart of the number of mis
sing values per column
plt.title('Bar chart showing the number of missing values in each column')
plt.grid()
plt.show()
```

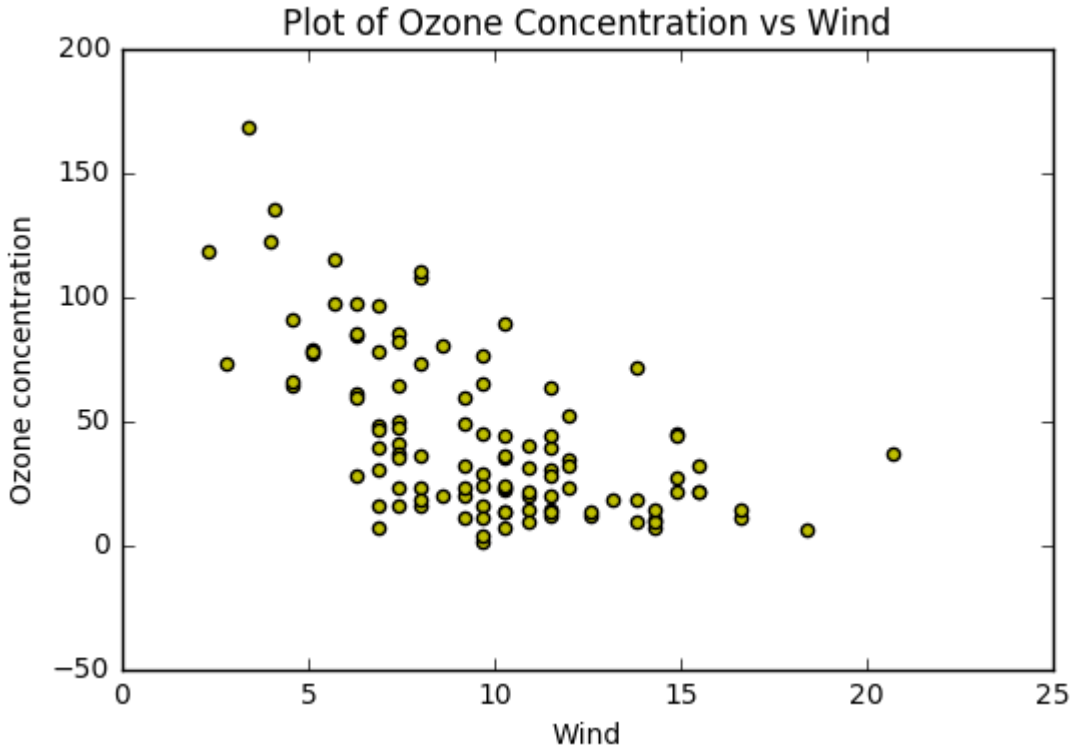


In [162]:
```
# Create a scatter plot of Ozone concentration against each of the other param
eters
```

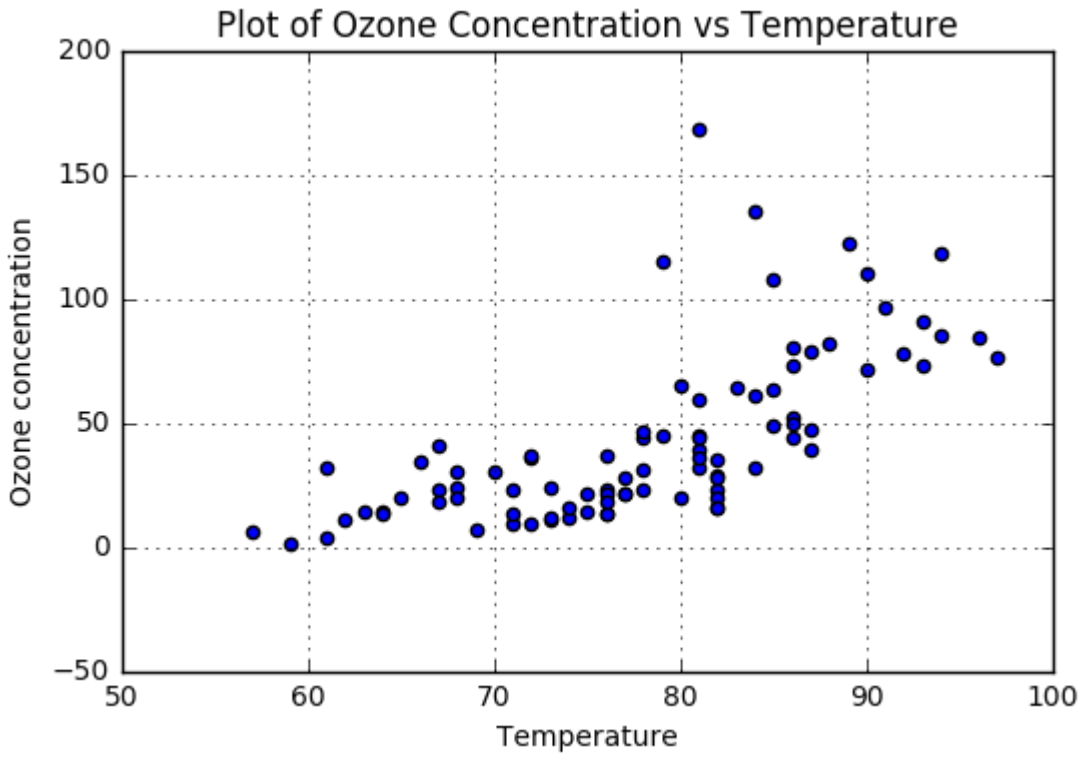
In [231]:
```
plt.scatter(Air_quality['Solar.R'],Air_quality['Ozone'],c = 'r')
plt.xlabel('Solar Radiation')
plt.ylabel('Ozone Concentration')
plt.title('Plot of Ozone Concentration vs Solar radiation')
plt.grid()
plt.show()
```



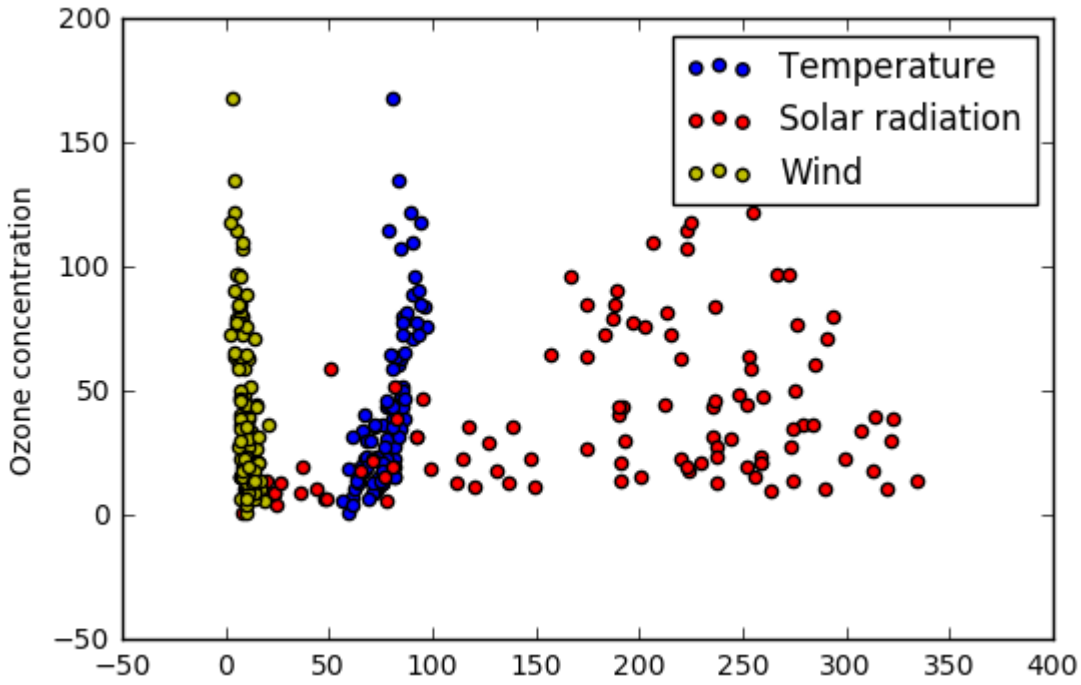
In [232]:
```
plt.scatter(Air_quality['Wind'],Air_quality['Ozone'],c = 'y')
plt.ylabel('Ozone concentration')
plt.xlabel('Wind')
plt.title('Plot of Ozone Concentration vs Wind')
plt.show()
```



In [165]:
```
plt.scatter(Air_quality['Temp'],Air_quality['Ozone'])
plt.grid()
plt.ylabel('Ozone concentration')
plt.xlabel('Temperature')
plt.title('Plot of Ozone Concentration vs Temperature')
plt.show()
```



In [233]:
```
# Create a scatter plot of Ozone concentration against all the variables
plt.scatter(Air_quality['Temp'], Air_quality['Ozone'], label='Temperature')
plt.scatter(Air_quality['Solar.R'],Air_quality['Ozone'],c = 'r',label = 'Solar
 radiation')
plt.scatter(Air_quality['Wind'],Air_quality['Ozone'],c = 'y', label = 'Wind')
plt.ylabel('Ozone concentration')
plt.legend()
plt.show()
```



In [245]:
```
Air_quality = Air_quality.dropna() # drops all the rows which have a missing v
alue
```

In [246]:
```
Air_quality
```

Out[246]:

	Ozone	Solar.R	Wind	Temp
1	41.0	190.0	7.4	67.0
2	36.0	118.0	8.0	72.0
3	12.0	149.0	12.6	74.0
16	14.0	334.0	11.5	64.0
17	34.0	307.0	12.0	66.0
18	6.0	78.0	18.4	57.0
19	30.0	322.0	11.5	68.0
20	11.0	44.0	9.7	62.0
21	1.0	8.0	9.7	59.0
22	11.0	320.0	16.6	73.0
23	4.0	25.0	9.7	61.0
24	32.0	92.0	12.0	61.0
28	23.0	13.0	12.0	67.0
29	45.0	252.0	14.9	81.0
30	115.0	223.0	5.7	79.0
31	37.0	279.0	7.4	76.0
38	29.0	127.0	9.7	82.0
40	71.0	291.0	13.8	90.0
41	39.0	323.0	11.5	87.0
44	23.0	148.0	8.0	82.0
47	21.0	191.0	14.9	77.0
48	37.0	284.0	20.7	72.0
49	20.0	37.0	9.2	65.0
50	12.0	120.0	11.5	73.0
51	13.0	137.0	10.3	76.0
62	135.0	269.0	4.1	84.0
63	49.0	248.0	9.2	85.0
64	32.0	236.0	9.2	81.0
78	35.0	274.0	10.3	82.0
79	61.0	285.0	6.3	84.0
...	...	...	...	...
123	85.0	188.0	6.3	94.0

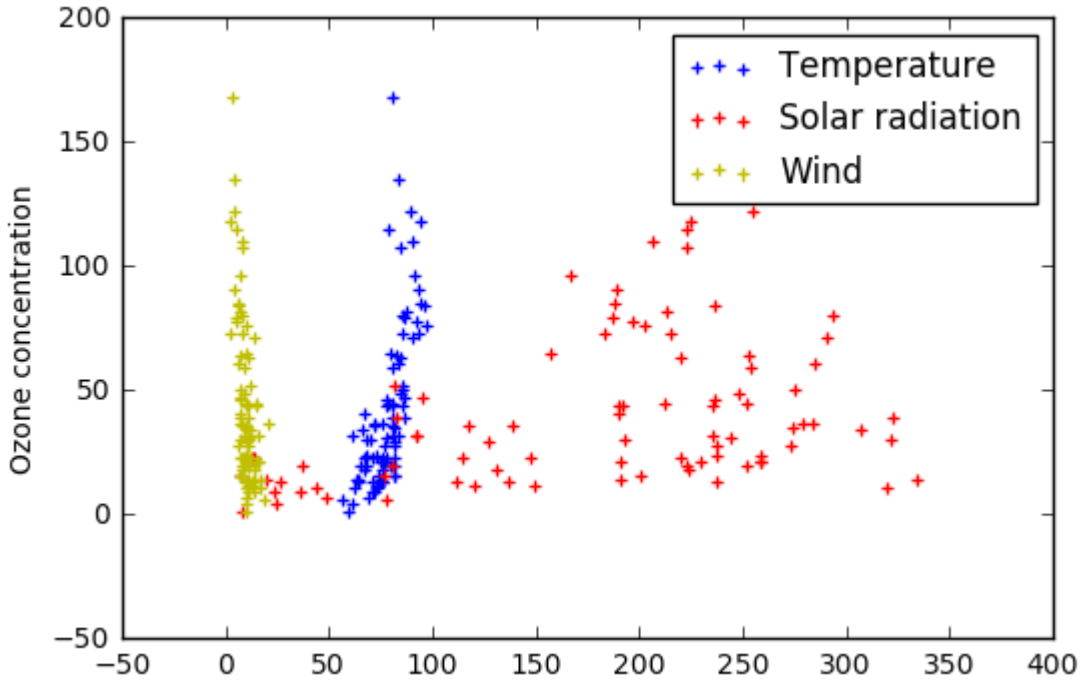
	Ozone	Solar.R	Wind	Temp
124	96.0	167.0	6.9	91.0
125	78.0	197.0	5.1	92.0
126	73.0	183.0	2.8	93.0
127	91.0	189.0	4.6	93.0
128	47.0	95.0	7.4	87.0
129	32.0	92.0	15.5	84.0
130	20.0	252.0	10.9	80.0
131	23.0	220.0	10.3	78.0
132	21.0	230.0	10.9	75.0
133	24.0	259.0	9.7	73.0
134	44.0	236.0	14.9	81.0
135	21.0	259.0	15.5	76.0
136	28.0	238.0	6.3	77.0
137	9.0	24.0	10.9	71.0
138	13.0	112.0	11.5	71.0
139	46.0	237.0	6.9	78.0
140	18.0	224.0	13.8	67.0
141	13.0	27.0	10.3	76.0
142	24.0	238.0	10.3	68.0
143	16.0	201.0	8.0	82.0
144	13.0	238.0	12.6	64.0
145	23.0	14.0	9.2	71.0
146	36.0	139.0	10.3	81.0
147	7.0	49.0	10.3	69.0
148	14.0	20.0	16.6	63.0
149	30.0	193.0	6.9	70.0
151	14.0	191.0	14.3	75.0
152	18.0	131.0	8.0	76.0
153	20.0	223.0	11.5	68.0

89 rows × 4 columns

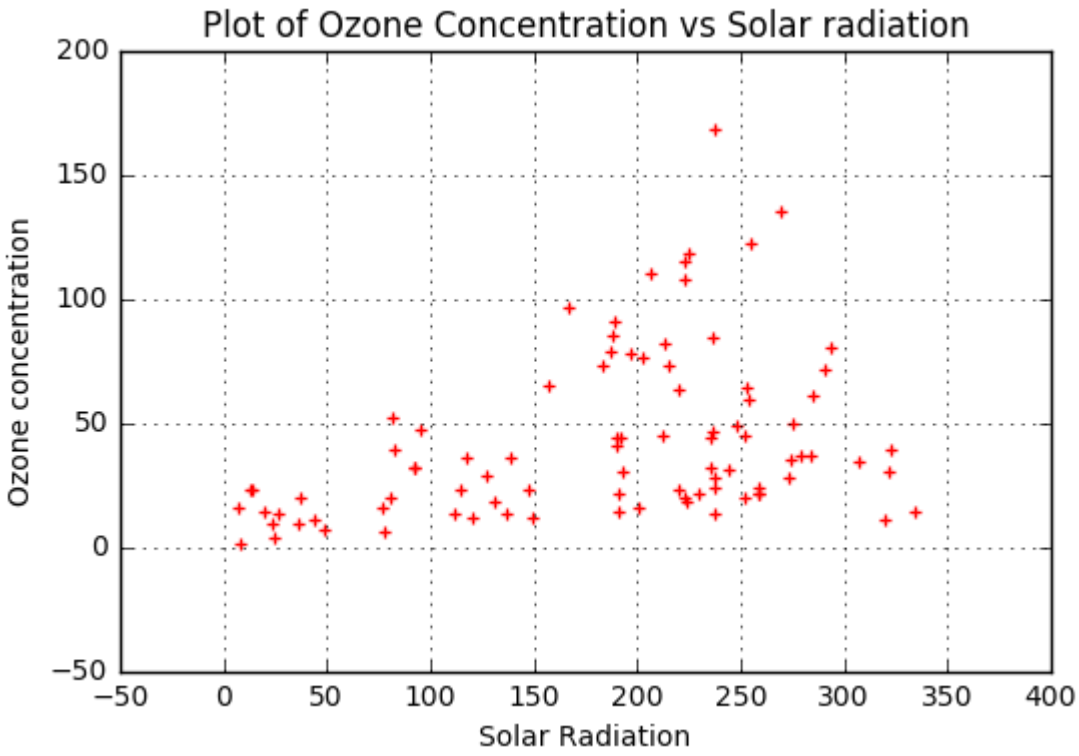
In [267]:
```
Air_quality.isnull().sum() # All the rows with missing values have been remove
d
```

Out[267]:
```
Ozone      0
Solar.R    0
Wind       0
Temp       0
dtype: int64
```

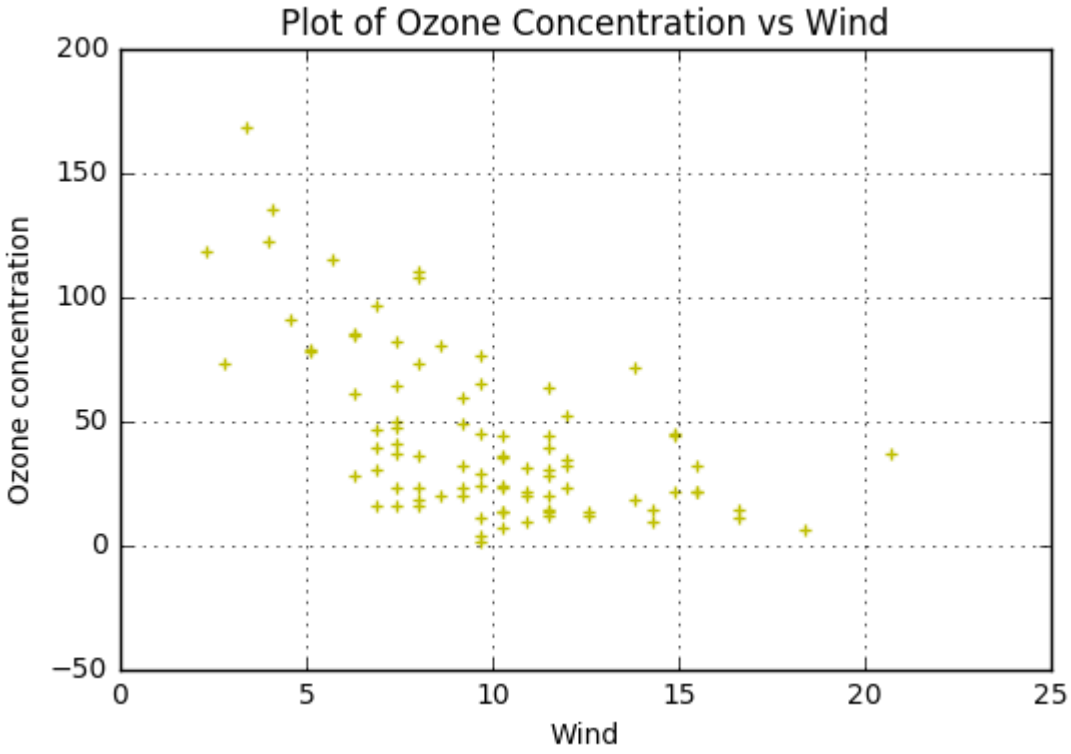
In [268]:
```
# Create a scatter plot of Ozone concentration against SOlar radiation
plt.scatter(Air_quality['Temp'], Air_quality['Ozone'], label='Temperature',mar
ker ='+')
plt.scatter(Air_quality['Solar.R'],Air_quality['Ozone'],c = 'r',label = 'Solar
 radiation',marker ='+')
plt.scatter(Air_quality['Wind'],Air_quality['Ozone'],c = 'y', label = 'Wind',m
arker ='+')
plt.ylabel('Ozone concentration')
plt.legend()
plt.show()
```



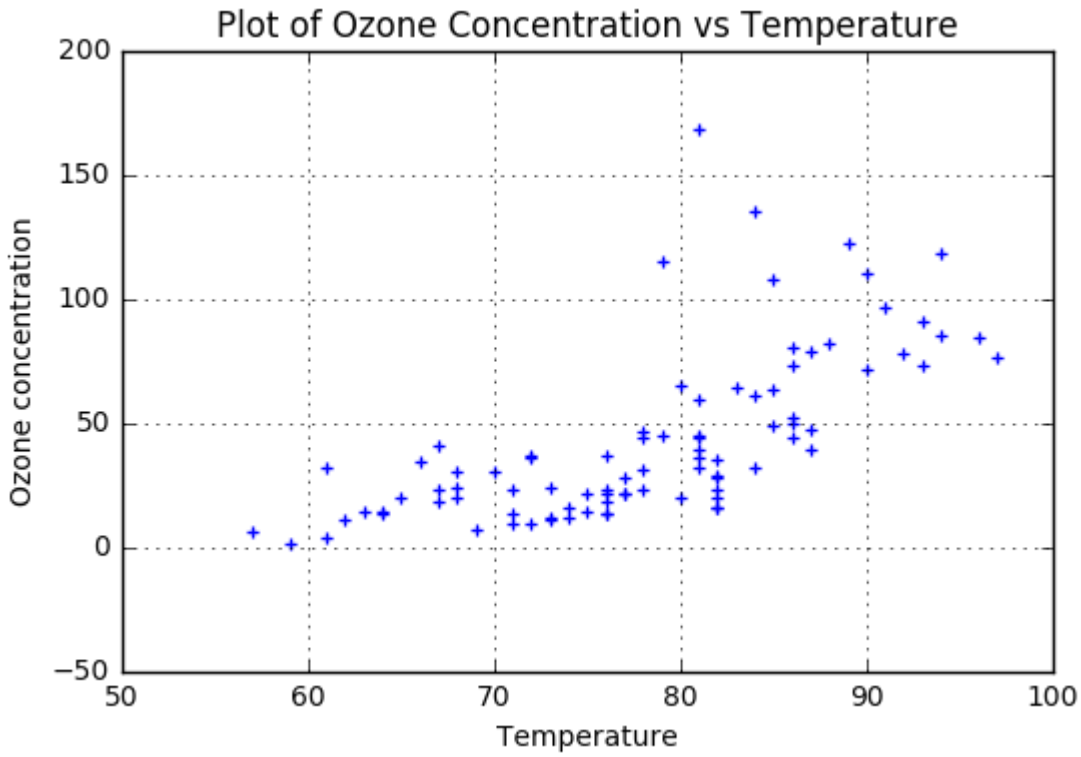
In [269]:
```
plt.scatter( Air_quality['Solar.R'],Air_quality['Ozone'],c = 'r', marker ='+')
plt.ylabel('Ozone concentration')
plt.xlabel('Solar Radiation')
plt.title('Plot of Ozone Concentration vs Solar radiation')
plt.grid()
plt.show()
```



In [270]:
```
plt.scatter(Air_quality['Wind'],Air_quality['Ozone'],c = 'y',marker ='+')
plt.ylabel('Ozone concentration')
plt.xlabel('Wind')
plt.title('Plot of Ozone Concentration vs Wind')
plt.grid()
plt.show()
```



In [271]:
```
plt.scatter(Air_quality['Temp'],Air_quality['Ozone'],marker ='+')
plt.ylabel('Ozone concentration')
plt.xlabel('Temperature')
plt.title('Plot of Ozone Concentration vs Temperature')
plt.grid()
plt.show()
```



In [272]:
```
# Import the dataset afresh
Air = pd.read_excel('HW4.xlsx')
```

In [273]:
```
Air.head() #Shows only the top 5 values
```

Out[273]:

	Ozone	Solar.R	Wind	Temp
1	41.0	190.0	7.4	67.0
2	36.0	118.0	8.0	72.0
3	12.0	149.0	12.6	74.0
4	18.0	313.0	NaN	62.0
5	NaN	NaN	NaN	56.0

In [274]:
```
# Calculate the mean values for each column
Ozone_mean = Air['Ozone'].mean()
Solar_mean = Air['Solar.R'].mean()
Wind_mean = Air['Wind'].mean()
Temp_mean = Air['Temp'].mean()
```

In [275]:
```
Ozone_mean
```

Out[275]: 42.12931034482759

In [276]:
```
Solar_mean
```

Out[276]: 185.95070422535213

In [277]:
```
Wind_mean
```

Out[277]: 9.806164383561642

In [278]:
```
Temp_mean
```

Out[278]: 77.859375

In [279]:
```
# Next fill each corresponding column with the corresponding mean value
```

In [280]:
```
Air.fillna(value = {"Ozone": Ozone_mean, 'Solar.R': Solar_mean, 'Wind': Wind_m
ean, 'Temp': Temp_mean }, inplace = True)
```

Out[280]:

	Ozone	Solar.R	Wind	Temp
1	41.00000	190.000000	7.400000	67.000000
2	36.00000	118.000000	8.000000	72.000000
3	12.00000	149.000000	12.600000	74.000000
4	18.00000	313.000000	9.806164	62.000000
5	42.12931	185.950704	9.806164	56.000000
6	28.00000	185.950704	9.806164	77.859375
7	23.00000	299.000000	9.806164	77.859375
8	19.00000	99.000000	9.806164	59.000000
9	8.00000	19.000000	9.806164	61.000000
10	42.12931	194.000000	9.806164	77.859375
11	7.00000	185.950704	6.900000	77.859375
12	16.00000	256.000000	9.700000	77.859375
13	11.00000	290.000000	9.200000	77.859375
14	14.00000	274.000000	10.900000	77.859375
15	18.00000	65.000000	13.200000	77.859375
16	14.00000	334.000000	11.500000	64.000000
17	34.00000	307.000000	12.000000	66.000000
18	6.00000	78.000000	18.400000	57.000000
19	30.00000	322.000000	11.500000	68.000000
20	11.00000	44.000000	9.700000	62.000000
21	1.00000	8.000000	9.700000	59.000000
22	11.00000	320.000000	16.600000	73.000000
23	4.00000	25.000000	9.700000	61.000000
24	32.00000	92.000000	12.000000	61.000000
25	42.12931	66.000000	16.600000	57.000000
26	42.12931	266.000000	14.900000	58.000000
27	42.12931	185.950704	8.000000	57.000000
28	23.00000	13.000000	12.000000	67.000000
29	45.00000	252.000000	14.900000	81.000000
30	115.00000	223.000000	5.700000	79.000000
...	...	...	...	...
124	96.00000	167.000000	6.900000	91.000000

	Ozone	Solar.R	Wind	Temp
125	78.00000	197.000000	5.100000	92.000000
126	73.00000	183.000000	2.800000	93.000000
127	91.00000	189.000000	4.600000	93.000000
128	47.00000	95.000000	7.400000	87.000000
129	32.00000	92.000000	15.500000	84.000000
130	20.00000	252.000000	10.900000	80.000000
131	23.00000	220.000000	10.300000	78.000000
132	21.00000	230.000000	10.900000	75.000000
133	24.00000	259.000000	9.700000	73.000000
134	44.00000	236.000000	14.900000	81.000000
135	21.00000	259.000000	15.500000	76.000000
136	28.00000	238.000000	6.300000	77.000000
137	9.00000	24.000000	10.900000	71.000000
138	13.00000	112.000000	11.500000	71.000000
139	46.00000	237.000000	6.900000	78.000000
140	18.00000	224.000000	13.800000	67.000000
141	13.00000	27.000000	10.300000	76.000000
142	24.00000	238.000000	10.300000	68.000000
143	16.00000	201.000000	8.000000	82.000000
144	13.00000	238.000000	12.600000	64.000000
145	23.00000	14.000000	9.200000	71.000000
146	36.00000	139.000000	10.300000	81.000000
147	7.00000	49.000000	10.300000	69.000000
148	14.00000	20.000000	16.600000	63.000000
149	30.00000	193.000000	6.900000	70.000000
150	42.12931	145.000000	13.200000	77.000000
151	14.00000	191.000000	14.300000	75.000000
152	18.00000	131.000000	8.000000	76.000000
153	20.00000	223.000000	11.500000	68.000000

153 rows × 4 columns

In [281]:
```
Air.head() # Notice the changes in the row with index 5
```

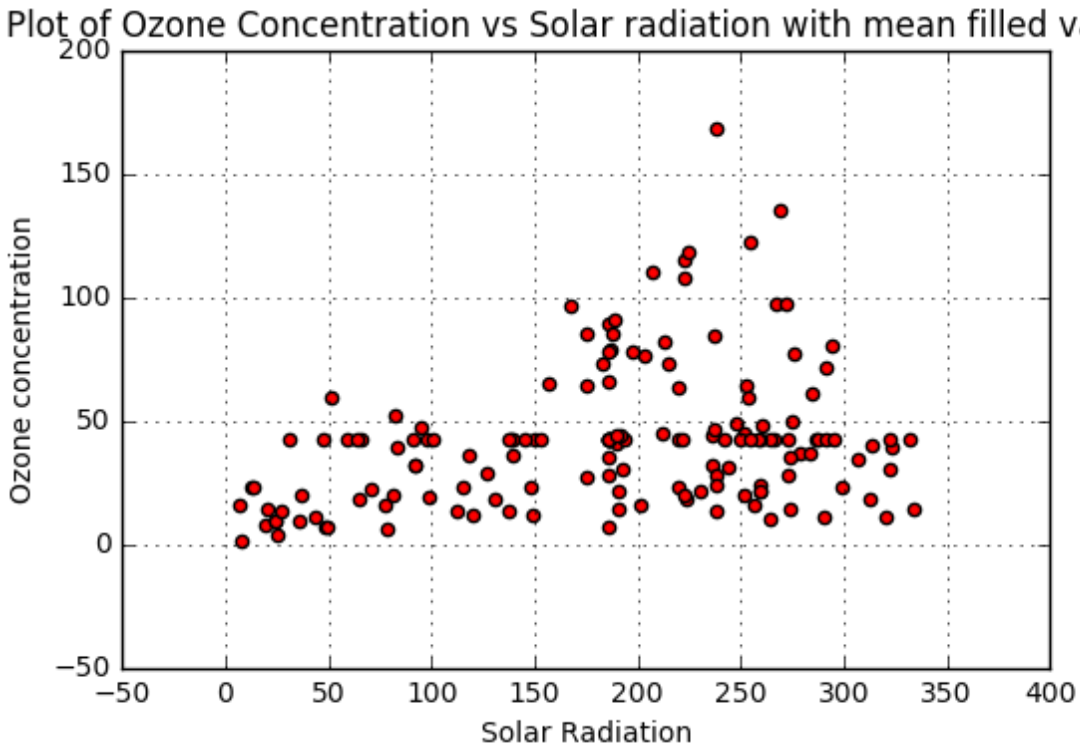
Out[281]:

	Ozone	Solar.R	Wind	Temp
1	41.00000	190.000000	7.400000	67.0
2	36.00000	118.000000	8.000000	72.0
3	12.00000	149.000000	12.600000	74.0
4	18.00000	313.000000	9.806164	62.0
5	42.12931	185.950704	9.806164	56.0

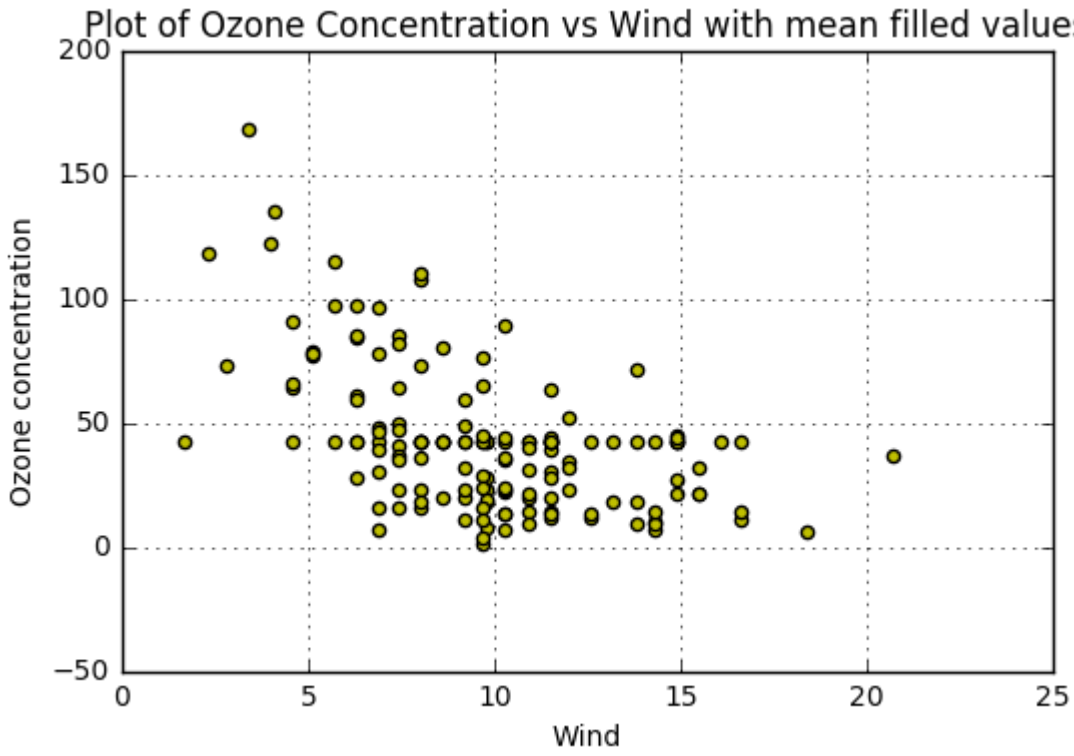
In [282]:
```
Air.isnull().sum()  # SUms up the number of rows with a Nan value
```

Out[282]:
```
Ozone      0
Solar.R    0
Wind       0
Temp       0
dtype: int64
```

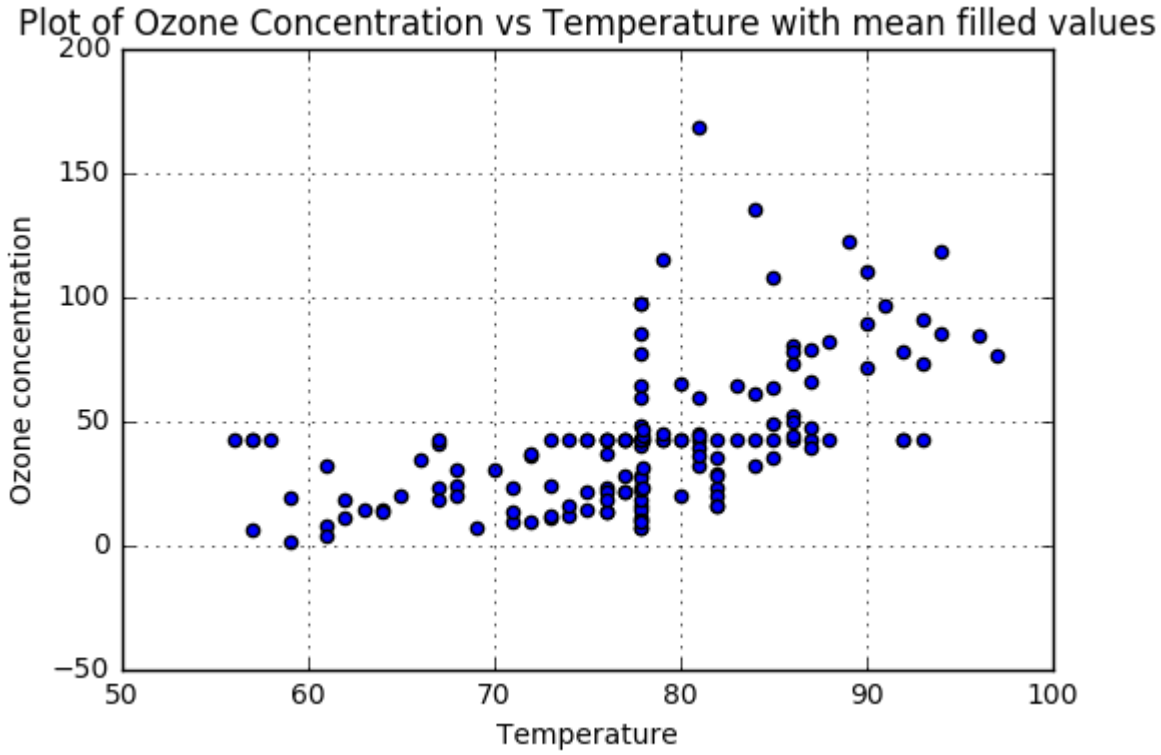
In [283]:
```
plt.scatter( Air['Solar.R'],Air['Ozone'],c = 'r', marker ='o')
plt.ylabel('Ozone concentration')
plt.xlabel('Solar Radiation')
plt.title('Plot of Ozone Concentration vs Solar radiation with mean filled val
ues')
plt.grid()
plt.show()
```



In [284]:
```
plt.scatter(Air['Wind'],Air['Ozone'],c = 'y',marker ='o')
plt.ylabel('Ozone concentration')
plt.xlabel('Wind')
plt.title('Plot of Ozone Concentration vs Wind with mean filled values')
plt.grid()
plt.show()
```



In [285]:
```
plt.scatter(Air['Temp'],Air['Ozone'],marker ='o')
plt.ylabel('Ozone concentration')
plt.xlabel('Temperature')
plt.title('Plot of Ozone Concentration vs Temperature with mean filled
values')
plt.grid()
plt.show()
```



In [ ]:
```

```

In [286]:
```
print('This is the end')
print("bows")
```

```
This is the end
bows
```

In [ ]:
```

```