Analysis_of_PM_Levels_in_Delhi

0.1 Analysis of Particulate Matter Levels in Delhi

0.2 1.Installing Required Libraries and Importing Datasets

[ ]:
```
# import the necessary libraries
import numpy as np
import pandas as pd
import os

# Visualisation libraries
import matplotlib.pyplot as plt
```

```
%matplotlib inline
import seaborn as sns
sns.set()

#Racing Bar Chart
!pip install bar_chart_race
import bar_chart_race as bcr
from IPython.display import HTML


# Increase the default plot size and set the color scheme
plt.rcParams['figure.figsize'] = 8, 5
plt.style.use("fivethirtyeight")# for pretty graphs


# Disable warnings
import warnings
warnings.filterwarnings('ignore')
```

[ ]:
```
# # Plotly libraries
import plotly.express as px
from plotly.offline import init_notebook_mode, iplot
import plotly.graph_objs as go
import plotly.offline as py
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot
!pip install chart_studio
import chart_studio.plotly as py
import cufflinks
cufflinks.go_offline()
cufflinks.set_config_file(world_readable=True, theme='pearl')
#py.init_notebook_mode(connected=True)
```

[379]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[380]:
```
# Dataset From CPCB
dfa = pd.read_csv(r"/content/drive/MyDrive/AP_dataset/AHMEDABAD_jUNE2019_22.
 ↪csv")
dfd = pd.read_csv(r"/content/drive/MyDrive/AP_dataset/DELHI_June2019.csv")
dfh = pd.read_csv(r"/content/drive/MyDrive/AP_dataset/HYDERABAD_June2019.csv")
dfk = pd.read_csv(r"/content/drive/MyDrive/AP_dataset/KOLKATA_JUNE19_22.csv")
dfm = pd.read_csv(r"/content/drive/MyDrive/AP_dataset/MUMBAI_JUNE_2019_22.csv")
```

The dataset collected from the CPCB website contains air quality data from June 2019 to November

2022 for five regions:

Delhi, Ahmedabad, Hyderabad, Mumbai, and Kolkata.

[381]:
```
data = [dfa, dfd, dfh,dfk,dfm]
```

0.3 2.Data Cleaning

[382]:
```
# for i in data:
#     print(i)
```

[383]:
```
for i in data:
    print(i.info)
```

```
<bound method DataFrame.info of                From Date           To Date  PM2.5
PM10     RH    WS
0       01-06-2019 01:00  01-06-2019 02:00  41.53  122.03     32  5.36
1       01-06-2019 02:00  01-06-2019 03:00  49.02  127.41     32  5.03
2       01-06-2019 03:00  01-06-2019 04:00  50.69  133.98     32  4.96
3       01-06-2019 04:00  01-06-2019 05:00  70.37   151.3     32  3.74
4       01-06-2019 05:00  01-06-2019 06:00  84.67  156.74     32  4.81
…                  …                 …    …     …    …   …
30595  29-11-2022 20:00  29-11-2022 21:00  38.31  106.09  54.53   0.2
30596  29-11-2022 21:00  29-11-2022 22:00   41.4  120.44  54.45  0.19
30597  29-11-2022 22:00  29-11-2022 23:00  38.02  125.03  53.99  0.48
30598  29-11-2022 23:00  30-11-2022 00:00  40.25  116.72  59.66  0.24
30599  30-11-2022 00:00  30-11-2022 01:00  39.71  122.15  57.11  0.66

[30600 rows x 6 columns]>
<bound method DataFrame.info of                From Date           To Date
WD     RH  BP   Temp    WS  \
0       01-06-2019 00:00  01-06-2019 01:00  195.85  36.93 NaN  29.98  0.53
1       01-06-2019 01:00  01-06-2019 02:00  240.57  34.87 NaN  30.37  0.51
2       01-06-2019 02:00  01-06-2019 03:00  216.42  30.23 NaN  31.48  0.69
3       01-06-2019 03:00  01-06-2019 04:00  142.53  32.35 NaN  30.91  0.58
4       01-06-2019 04:00  01-06-2019 05:00  180.31  38.29 NaN  29.06  0.55
…                  …                 …    …    …  ..    …   …
30596  29-11-2022 20:00  29-11-2022 21:00  117.48  35.92 NaN  14.82  0.71
30597  29-11-2022 21:00  29-11-2022 22:00  108.81  35.94 NaN  14.34  0.69
30598  29-11-2022 22:00  29-11-2022 23:00   69.74  35.94 NaN  13.86   0.7
30599  29-11-2022 23:00  30-11-2022 00:00  102.23  35.96 NaN  12.64  0.74
30600  30-11-2022 00:00  30-11-2022 01:00   125.7  35.96 NaN   12.2  0.74

       TOT-RF    RF    PM10   PM2.5
0         NaN     0    1000  127.06
1         NaN     0    1000  137.88
2         NaN     0    1000  172.98
```

```
3          NaN     0    1000   93.64
4          NaN     0    1000  105.42
…        …   …     …      …
30596      NaN  None  642.25   101.5
30597      NaN  None  800.25   129.5
30598      NaN  None     803  152.25
30599      NaN  None     811     172
30600      NaN  None  559.25  133.75

[30601 rows x 11 columns]>
<bound method DataFrame.info of               From Date           To Date  PM2.5
PM10   Temp     RH     AT
0      01-06-2019 00:00  01-06-2019 01:00  43.75  145.25  29.58     46   31.8
1      01-06-2019 01:00  01-06-2019 02:00  48.75     160  29.55  49.75  30.43
2      01-06-2019 02:00  01-06-2019 03:00   39.5   117.5  29.65  49.25  30.42
3      01-06-2019 03:00  01-06-2019 04:00     39  126.75  30.25   48.5  30.12
4      01-06-2019 04:00  01-06-2019 05:00   33.5   102.5  30.05  54.25  28.65
…                 …                 …    …     …    …    …    …
30596  29-11-2022 20:00  29-11-2022 21:00  82.25  168.25   31.2  65.25  22.27
30597  29-11-2022 21:00  29-11-2022 22:00   81.5  180.25  31.27   66.5  22.03
30598  29-11-2022 22:00  29-11-2022 23:00     77     169  31.23  69.33  21.43
30599  29-11-2022 23:00  30-11-2022 00:00  78.75  147.75   31.4     71  21.17
30600  30-11-2022 00:00  30-11-2022 01:00   84.5   138.5   31.8  71.75   20.9

[30601 rows x 7 columns]>
<bound method DataFrame.info of               From Date           To Date
PM2.5    PM10     RH    WS     AT
0      01-06-2019 01:00  01-06-2019 02:00    None    None   None  None   None
1      01-06-2019 02:00  01-06-2019 03:00    None    None   None  None   None
2      01-06-2019 03:00  01-06-2019 04:00    None    None   None  None   None
3      01-06-2019 04:00  01-06-2019 05:00    None    None   None  None   None
4      01-06-2019 05:00  01-06-2019 06:00    None    None   None  None   None
…                 …                 …     …     …    …   …    …
30595  29-11-2022 20:00  29-11-2022 21:00   96.53  194.09  65.85  0.13  24.19
30596  29-11-2022 21:00  29-11-2022 22:00  120.38  234.25  57.58  0.22  24.58
30597  29-11-2022 22:00  29-11-2022 23:00  119.63  229.78  62.07  0.07     24
30598  29-11-2022 23:00  30-11-2022 00:00  129.94  237.81  69.91  0.06  23.84
30599  30-11-2022 00:00  30-11-2022 01:00  130.48  252.19   76.5   0.1  23.43

[30600 rows x 7 columns]>
<bound method DataFrame.info of               From Date           To Date  PM2.5
PM10     RH    WS     AT
0      01-06-2019 01:00  01-06-2019 02:00   None    None   None  None   None
1      01-06-2019 02:00  01-06-2019 03:00   None    None   None  None   None
2      01-06-2019 03:00  01-06-2019 04:00   None    None   None  None   None
3      01-06-2019 04:00  01-06-2019 05:00   None    None   None  None   None
4      01-06-2019 05:00  01-06-2019 06:00   None    None   None  None   None
…                 …                 …    …     …    …   …    …
```

```
30595  29-11-2022 20:00  29-11-2022 21:00  96.32  155.51  87.62  0.79  21.79
30596  29-11-2022 21:00  29-11-2022 22:00  75.92  106.69  84.06  0.87  21.24
30597  29-11-2022 22:00  29-11-2022 23:00  53.11   68.22  83.75  0.72  20.97
30598  29-11-2022 23:00  30-11-2022 00:00  62.84  106.67  89.77  0.77  20.77
30599  30-11-2022 00:00  30-11-2022 01:00  73.11     106  88.49  0.69  20.28

[30600 rows x 7 columns]>
```

[384]:
```
def clean(df):

  # 1 Replace Null Values
  df.replace('None', np.nan, inplace=True)
  # 2 Change Data types of Columns
  df= df.astype({ 'PM10': 'float','PM2.5': 'float','RH': 'float',})
  # Rename Date Column
  df.rename(columns = {'From Date':'Date'}, inplace = True)
  #  Change Date Format
  df['Date'] = pd.to_datetime(df['Date'], format='%d-%m-%Y %H:%M')
  #Choose only Required Column
  df = df[['Date','PM10']]


  return(df)
```

[385]:
```
# Apply function
data = [clean(i) for i in data]
```

[386]:
```
data[0].rename(columns = {'PM10':'Ahmedabad'}, inplace = True)
data[1].rename(columns = {'PM10':'Delhi'}, inplace = True)
data[2].rename(columns = {'PM10':'Hyderabad'}, inplace = True)
data[3].rename(columns = {'PM10':'Kolkata'}, inplace = True)
data[4].rename(columns = {'PM10':'Mumbai'}, inplace = True)
```

[387]:
```
for i in data:
    print(i.isna().sum())
```

```
Date            0
Ahmedabad    2440
dtype: int64
Date        0
Delhi    2102
dtype: int64
Date            0
Hyderabad    2054
dtype: int64
Date          0
Kolkata    1116
dtype: int64
```

```
Date         0
Mumbai    3676
dtype: int64
```

Resampling the data for evaluating Monthly Average values

[388]:
```
def resamp(df):
  df.set_index('Date', inplace = True)
  df= df.resample("M").mean()
  # Change Date text for better visualisation
  df.reset_index(inplace = True)
  df['Date'] = df['Date'].dt.strftime('%Y-%B')
  df.set_index('Date', inplace = True)


  return(df)
```

[389]:
```
# Apply function
data = [resamp(i) for i in data]
```

[390]:
```
# Displaying Effect
# for i in data:
#     print(i)
```

0.4 3.Joining the Datasets

[391]:
```
# Merging Data frames on the basis of index

dfa, dfd, dfh,dfk,dfm = data[0],data[1],data[2],data[3],data[4]
result = pd.concat([dfa, dfd, dfh,dfk,dfm], axis=1)
```

[392]:
```
dfa, dfd, dfh,dfk,dfm
```

[392]:
```
(                Ahmedabad
 Date
 2019-June       115.904420
 2019-July       113.729615
 2019-August     136.844570
 2019-September   79.160353
 2019-October    139.174492
 2019-November   111.788395
 2019-December   109.309060
 2020-January    113.164767
 2020-February   144.323848
 2020-March      107.254686
 2020-April       81.454837
```

```
2020-May         76.672898
2020-June        97.133046
2020-July        68.307889
2020-August      69.559064
2020-September   71.612066
2020-October    122.226003
2020-November   158.602386
2020-December   162.585100
2021-January    164.614693
2021-February   197.737352
2021-March      205.953967
2021-April      156.854825
2021-May         89.941798
2021-June       150.333230
2021-July        85.809521
2021-August      99.566449
2021-September   91.496239
2021-October    149.148317
2021-November   197.033153
2021-December   176.712712
2022-January    161.626132
2022-February   207.258397
2022-March      184.157873
2022-April      140.804658
2022-May        154.042794
2022-June       112.125554
2022-July        78.750448
2022-August      82.895105
2022-September   73.462475
2022-October    157.962710
2022-November   134.411298,                    Delhi
Date
2019-June       228.933644
2019-July       147.392547
2019-August      92.756749
2019-September  108.419528
2019-October    277.654957
2019-November   320.299060
2019-December   317.970156
2020-January    269.091009
2020-February   277.513435
2020-March      162.335952
2020-April      124.148508
2020-May        175.428385
2020-June       150.068262
2020-July        99.655530
2020-August      72.467049
```

```
2020-September  178.273118
2020-October    335.321131
2020-November   350.381172
2020-December   315.399795
2021-January    282.511455
2021-February   294.233164
2021-March      285.616785
2021-April      244.610627
2021-May        149.478992
2021-June       165.702550
2021-July       112.691686
2021-August     108.806137
2021-September   67.215100
2021-October    160.062843
2021-November   326.228076
2021-December   281.756142
2022-January    196.684442
2022-February   202.383306
2022-March      223.415871
2022-April      320.817577
2022-May        251.464283
2022-June       222.076576
2022-July        79.474151
2022-August      89.230014
2022-September   88.376566
2022-October    209.329787
2022-November   365.068165,                    Hyderabad
Date
2019-June        74.881139
2019-July        68.521969
2019-August      73.538122
2019-September   55.972651
2019-October     72.505842
2019-November   132.347808
2019-December   124.056399
2020-January    100.057459
2020-February   100.984884
2020-March       80.100780
2020-April       57.734145
2020-May         82.685082
2020-June        63.162530
2020-July        41.088595
2020-August      40.246173
2020-September   54.528092
2020-October    107.847014
2020-November   106.932433
2020-December   131.983896
```

```
2021-January    120.438847
2021-February   127.028169
2021-March      127.278696
2021-April      116.649291
2021-May         72.228738
2021-June        57.346364
2021-July        41.010299
2021-August      59.663914
2021-September   49.040611
2021-October     99.517000
2021-November    84.778241
2021-December   125.505119
2022-January    107.719512
2022-February   119.543660
2022-March      127.108939
2022-April      119.036127
2022-May        124.122779
2022-June        90.529626
2022-July        57.571601
2022-August      64.149154
2022-September   73.356186
2022-October     86.922330
2022-November   137.416948,                    Kolkata
Date
2019-June        49.826096
2019-July        34.264612
2019-August      31.044552
2019-September   22.466880
2019-October     86.560286
2019-November   159.993764
2019-December   156.445723
2020-January    144.981518
2020-February   154.010843
2020-March       91.915266
2020-April       47.992601
2020-May         34.616976
2020-June        31.360172
2020-July        30.681875
2020-August      30.993393
2020-September   32.889733
2020-October     68.816164
2020-November   158.473955
2020-December   240.176792
2021-January    234.260630
2021-February   204.318511
2021-March      146.029972
2021-April       84.594808
```

```
2021-May         46.717781
2021-June        49.677069
2021-July        46.940081
2021-August      50.658816
2021-September   45.230263
2021-October     94.143696
2021-November   160.974879
2021-December   191.048305
2022-January    185.708807
2022-February   151.661736
2022-March      131.070749
2022-April       49.686261
2022-May         59.351777
2022-June        52.581085
2022-July        31.729714
2022-August      41.693415
2022-September   46.426728
2022-October     71.850392
2022-November   171.633585,                    Mumbai
Date
2019-June        50.836151
2019-July        32.879381
2019-August      39.356961
2019-September   30.161610
2019-October     62.466375
2019-November    90.285724
2019-December   119.038412
2020-January    115.208565
2020-February   126.484126
2020-March       88.747078
2020-April       69.204665
2020-May         39.893071
2020-June        27.945987
2020-July        24.209493
2020-August      24.556521
2020-September   39.070976
2020-October     72.409709
2020-November   103.141897
2020-December   117.629586
2021-January    143.068129
2021-February   108.833896
2021-March      119.156640
2021-April      106.213530
2021-May         63.297273
2021-June        48.954415
2021-July        43.410292
2021-August      42.130371
```

```
2021-September   39.071550
2021-October     78.171457
2021-November   109.631455
2021-December   134.123165
2022-January    152.944593
2022-February   135.004614
2022-March      146.256137
2022-April      154.486501
2022-May        108.241049
2022-June        39.979098
2022-July        35.177680
2022-August      36.674151
2022-September   35.719927
2022-October     64.714047
2022-November   118.534275)
```

0.5 4.Animated Bar Graph

[393]:
```
#Racing Bar Chart
!pip install bar_chart_race
import bar_chart_race as bcr
from IPython.display import HTML
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: bar_chart_race in /usr/local/lib/python3.8/dist-
packages (0.1.0)
Requirement already satisfied: pandas>=0.24 in /usr/local/lib/python3.8/dist-
packages (from bar_chart_race) (1.3.5)
Requirement already satisfied: matplotlib>=3.1 in /usr/local/lib/python3.8/dist-
packages (from bar_chart_race) (3.2.2)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib/python3.8/dist-
packages (from matplotlib>=3.1->bar_chart_race) (1.21.6)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.8/dist-
packages (from matplotlib>=3.1->bar_chart_race) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib>=3.1->bar_chart_race)
(1.4.4)
Requirement already satisfied: python-dateutil>=2.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib>=3.1->bar_chart_race)
(2.8.2)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.8/dist-packages (from matplotlib>=3.1->bar_chart_race)
(3.0.9)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.8/dist-
packages (from pandas>=0.24->bar_chart_race) (2022.6)
```

```
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.8/dist-
packages (from python-dateutil>=2.1->matplotlib>=3.1->bar_chart_race) (1.15.0)
```

0.5.1 Animated Bar graph representing the levels of P.M.10 in 5 Indian cities.

[394]:
```
#Source code for racing barchart: https://github.com/dexplo/bar_chart_race

race_chart = bcr.bar_chart_race(df=result, filename=None, steps_per_period=1,␣
↪period_length=1000,orientation='h',figsize=(5, 3.
↪5),bar_label_size=10,fixed_max=True, tick_label_size=10,title='Particulate␣
↪Matter P.M.  (µg/m³)',title_size='smaller', cmap='set1',period_label={'x': .
↪90, 'y': .30, 'ha': 'right', 'color': 'blue'},dpi=150, bar_size=.5, ␣
↪bar_kwargs={'alpha': .8, 'ec': 'black', 'lw': 1})
```

[395]:
```
race_chart
```

[395]:
```
<IPython.core.display.HTML object>
```

0.6 Data Preparation for charts

New Analysis for plotting other parameters of DELHI

In this i have taken the data of Delhi from Jan 2017 to 2022 Nov

[396]:
```
dfdelhi = pd.read_csv(r"/content/drive/MyDrive/AP_dataset/cpcb_airdataset.csv")
```

[397]:
```
dfdelhi
```

[397]:

	From Date	To Date	WD	RH	BP	Temp	WS	TOT-RF	RF	PM10	PM2.5
0	01-01-2017 01:00	01-01-2017 02:00	99.25	88.6	NaN	19.2	0.15	NaN	None	NaN	245.75
1	01-01-2017 02:00	01-01-2017 03:00	321.75	88.6	NaN	19.05	0.25	NaN	None	NaN	220.75
2	01-01-2017 03:00	01-01-2017 04:00	292.25	88.52	NaN	18.53	0.3	NaN	None	NaN	267
3	01-01-2017 04:00	01-01-2017 05:00	344.25	88.45	NaN	18.48	0.3	NaN	None	NaN	319
4	01-01-2017 05:00	01-01-2017 06:00	87.5	88.4	NaN	18.38	0.3				
…	…	…	…	…	..	…	…				
51711	30-11-2022 16:00	30-11-2022 17:00	55.19	35.83	NaN	22.44	0.72				
51712	30-11-2022 17:00	30-11-2022 18:00	128.04	35.89	NaN	17.73	0.74				
51713	30-11-2022 18:00	30-11-2022 19:00	103.35	35.91	NaN	15.59	0.75				
51714	30-11-2022 19:00	30-11-2022 20:00	115.34	35.92	NaN	14.62	0.76				
51715	30-11-2022 20:00	30-11-2022 21:00	113.93	35.93	NaN	13.92	0.73				

```
4          NaN  None     NaN  289.75
…        …   …     …      …
51711      NaN  None  176.25   60.25
51712      NaN  None  220.25      76
51713      NaN  None     398    82.5
51714      NaN  None     537  112.75
51715      NaN  None     648     149

[51716 rows x 11 columns]
```

Cleaning of Data

[398]:
```
# 1 Replace Null Values
dfdelhi.replace('None', np.nan, inplace=True)
# 2 Change Data types of Columns
dfdelhi= dfdelhi.astype({ 'PM10': 'float','PM2.5': 'float','RH': 'float','WS':
↪ 'float','Temp': 'float','TOT-RF':'float','RF':'float'})
# Rename Date Column
dfdelhi.rename(columns = {'From Date':'Date'}, inplace = True)
#  Change Date Format
dfdelhi['Date'] = pd.to_datetime(dfdelhi['Date'], format='%d-%m-%Y %H:%M')
#Choose only Required Column
dfdelhi2 = dfdelhi[['Date','PM10','PM2.5','WS','Temp','RH']]
```

Cleaning is Done - Now Make Dataset of Daily Avg , Monthly avg

[399]:
```
dfdelhi2.set_index('Date', inplace = True)
dfdelhiM= dfdelhi2.resample("M").mean()
#use MS for starting date index
dfdelhiD= dfdelhi2.resample("D").mean()
```

[400]:
```
dfdelhiH=dfdelhi2.copy()
```

Now we have 3 dataset dfdelhiH, dfdelhiD, dfdelhiM for Hourly, Daily, Monthly averages value of Delhi's air quality.

[401]:
```
# Now we have 4 dataset
# dfdelhiH,dfdelhiD,dfdelhiM,for Hourly, Daily, Monthly
dfdelhiH,dfdelhiD,dfdelhiM
```

[401]:
```
(                     PM10    PM2.5    WS   Temp     RH
 Date
 2017-01-01 01:00:00     NaN  245.75  0.15  19.20  88.60
 2017-01-01 02:00:00     NaN  220.75  0.25  19.05  88.60
 2017-01-01 03:00:00     NaN  267.00  0.30  18.53  88.52
 2017-01-01 04:00:00     NaN  319.00  0.30  18.48  88.45
 2017-01-01 05:00:00     NaN  289.75  0.30  18.38  88.40
 …                     …      …    …     …      …
```

```
2022-11-30 16:00:00  176.25   60.25  0.72  22.44  35.83
2022-11-30 17:00:00  220.25   76.00  0.74  17.73  35.89
2022-11-30 18:00:00  398.00   82.50  0.75  15.59  35.91
2022-11-30 19:00:00  537.00  112.75  0.76  14.62  35.92
2022-11-30 20:00:00  648.00  149.00  0.73  13.92  35.93

[51716 rows x 5 columns],
                  PM10       PM2.5        WS       Temp         RH
Date
2017-01-01         NaN  230.195652  0.395217  20.889130  79.121739
2017-01-02         NaN  203.468750  0.310417  21.882917  72.740833
2017-01-03         NaN  287.739583  0.317500  21.163333  78.483750
2017-01-04         NaN  348.770833  0.315417  22.230833  68.847917
2017-01-05         NaN  239.697917  0.284167  22.038333  72.214583
…                  …          …       …         …          …
2022-11-26  361.947917   78.406250  0.592500  12.177083  10.880833
2022-11-27  332.135417   76.458333  0.627083  11.917083  10.885833
2022-11-28  369.899167   75.947917  0.613333  11.112083  10.890833
2022-11-29  543.208333  107.395833  0.642083  13.784167  19.087917
2022-11-30  426.250000  101.308824  0.639048  19.210476  35.874762

[2160 rows x 5 columns],
                  PM10       PM2.5        WS       Temp         RH
Date
2017-01-31         NaN  206.198649  0.399176  19.075324  73.314041
2017-02-28         NaN  159.133502  0.579717  18.126622  58.506964
2017-03-31         NaN  113.647226  0.649556  20.219597  39.865806
2017-04-30         NaN  107.611679  0.658457  30.741965  21.642434
2017-05-31         NaN  100.236213  0.490743  33.474263  39.831525
…                  …          …       …         …          …
2022-07-31   79.474151   40.183093  0.578187  22.931840  73.658457
2022-08-31   89.230014   38.715691  0.592938  22.404111  63.522588
2022-09-30   88.376566   35.705973  0.583648  24.282017  41.031202
2022-10-31  209.329787   92.306764  0.564274  26.930290  10.763181
2022-11-30  366.431930   93.495051  0.583492  12.244332  11.847430

[71 rows x 5 columns])
```

[402]:
```
print(dfdelhiH.isna().sum())
print(dfdelhiD.isna().sum())
print(dfdelhiM.isna().sum())
```

```
PM10     13451
PM2.5     4760
WS        2445
Temp      1882
RH        2400
```

```
dtype: int64
PM10     480
PM2.5     52
WS        58
Temp      30
RH        40
dtype: int64
PM10     14
PM2.5     0
WS        0
Temp      0
RH        0
dtype: int64
```

0.7 5.Clustered Bar chart

For clusted chart -

Data is taken which have concentration of PM10 and PM2.5 from duration of Jan 2022 to November 2022

name of dataframe is cc

cc abbreviation used for clustered chart

[403]:
```
# cc abbreviation is for clustered chart

cc = dfdelhiM.loc['2022-01-01':'2022-12-01']
```

[404]:
```
# Import the necessaries libraries
import plotly.offline as pyo
# Set notebook mode to work in offline
pyo.init_notebook_mode()
```

0.7.1 Delhi's P.M. 10 and P.M. 2.5 concentration this year

[405]:
```
import plotly.graph_objects as go

years = ['January', 'February', 'March', 'April','May', 'June', 'July',␣
 ↪'August', 'September', 'October', 'November']

fig = go.Figure()
fig.add_trace(go.Bar(x=years,
                y=cc['PM10'],
                name='P.M.  (µg/m³)',
                marker_color='rgb(55, 83, 109)'
                ))
```

```
fig.add_trace(go.Bar(x=years,
                y=cc['PM2.5'],
                name='P.M. .  (µg/m³)',
                marker_color='rgb(26, 118, 255)'
                ))

fig.update_layout(
    title='PARTICULATE MATTER - P.M.   & P.M .  concentration in 2022',
    xaxis_tickfont_size=14,
    yaxis=dict(
        title='P.M.   & P.M .  in (µg/m³)',
        titlefont_size=16,
        tickfont_size=14,
    ),
    legend=dict(
        x=0,
        y=1.0,
        bgcolor='rgba(255, 255, 255, 0)',
        bordercolor='rgba(255, 255, 255, 0)',
        font=dict(size= 20)
    ),
    barmode='group',
    bargap=0.15, # gap between bars of adjacent location coordinates.
    bargroupgap=0.1 # gap between bars of the same location coordinate.
)

fig.show(renderer="colab")
```

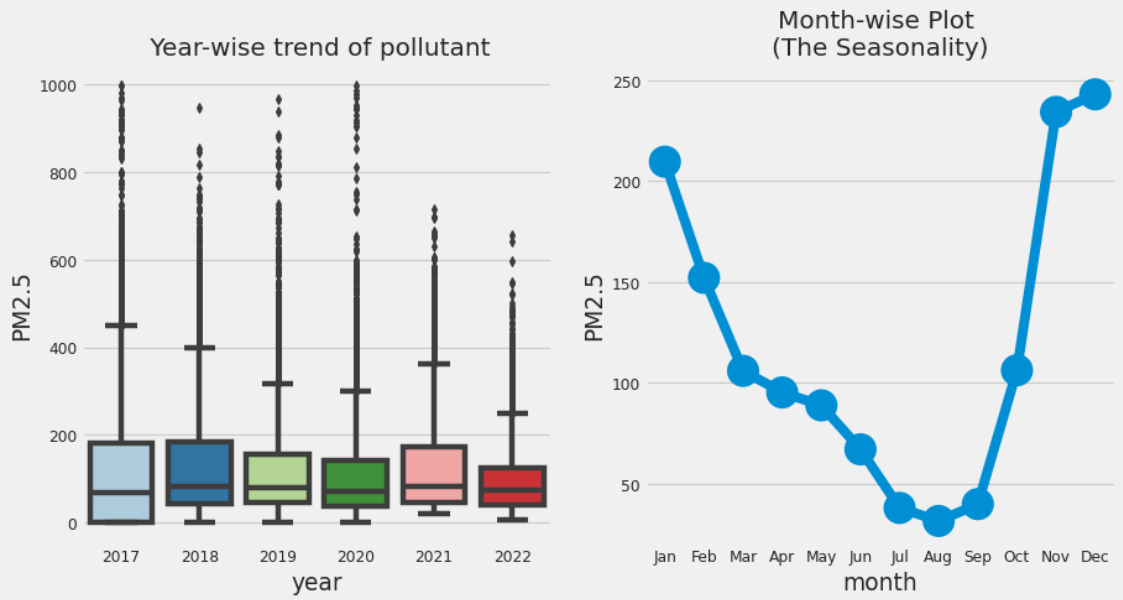
Note:

- In Delhi, atmospheric inversion is more likely to occur during the winter months, when the temperature difference between the ground and the air above is greater.
- This can lead to higher concentrations of P.M.10 and P.M.2.5 in the air, which can have negative health effects for residents.
- Monsoon season can also affect the concentration of P.M.10 and P.M.2.5 in the air. Rain can help to clear the air by washing away pollutants and increasing the dispersion of PM particles.
- This can lead to lower levels of P.M.10 and P.M.2.5 in the air during the monsoon season.
- Overall, the variation in P.M.10 and P.M.2.5 levels in Delhi is likely to be influenced by seasonal and monthly factors, such as atmospheric inversion, rain, and other meteorological conditions.

0.8 6.Year Wise Box plot & Seasonal Wise Plot

[406]:
```
dfdelhiH.reset_index(inplace =True)
```

[407]:
```
# Prepare data
dfdelhiH['year'] = [d.year for d in dfdelhiH.Date]
dfdelhiH['month'] = [d.strftime('%b') for d in dfdelhiH.Date]
years = dfdelhiH['year'].unique()

    # Draw Plot
fig, axes = plt.subplots(1, 2, figsize=(12,6), dpi=80)
sns.boxplot(x='year', y = dfdelhiH['PM2.5'], data=dfdelhiH,␣
 ↪ax=axes[0],palette="Paired")
sns.pointplot(x='month', y = dfdelhiH['PM2.5'], data=dfdelhiH)

    # Set Title
axes[0].set_title(' Year-wise trend of pollutant', fontsize=18);
axes[1].set_title('Month-wise Plot \n(The Seasonality)', fontsize=18)

    #Setting the background color of the plot
# using set_facecolor() method
# ax.set_facecolor("gray")
# axes.set_facecolor("yellow")
plt.show()
```



Note:

- Boxplot could be used to show the variation in P.M.2.5 levels over time.

- Boxplot showed that the median P.M.2.5 level in Delhi was 100, with a first quartile of 90 and a third quartile of 110, it would indicate that 50% of the P.M.2.5 levels in Delhi were between 90 and 110.
- The whiskers extending from the box might show that the minimum P.M.2.5 level was 80 and the maximum level was 120. This would suggest that P.M.2.5 levels in Delhi can vary greatly, but tend to be concentrated within a relatively narrow range.
- In the above figure, the pointplot showed a cluster of points near the bottom of the graph for the months of June, July, and August, it is indicating that P.M.2.5 levels were lower during those months.
- Conversely,the pointplot showed a cluster of points near the top of the graph for the months of December, January, and February, it is indicating that P.M.2.5 levels were higher during those months.

rectifying the hourly data

[408]:
```
dfdelhiH.set_index('Date', inplace = True)
```

[409]:
```
del dfdelhiH['year']
del dfdelhiH['month']
```

0.9 7.Top 10 Polluted Days of 2022

Using daily AVG data from Delhi from January to November 2022,

with name TopDays

[410]:
```
TopDays =dfdelhiD.loc['2022-01-01':'2022-12-01']
```

[411]:
```
def max_polluted_day(value):
 x1 =TopDays.sort_values(by=value,ascending=False)
 x1 = x1[[value]]
 x1 = round(x1,2)
 x1.reset_index(inplace = True)
 x1['Date'] = x1['Date'].dt.strftime('%d-%B')
 x1.set_index('Date', inplace = True)
 x1 =x1.round(1)
 x1 = x1[:10]
# set style

 return x1.style.background_gradient(cmap='OrRd')
```

[412]:
```
pm2_5 = max_polluted_day('PM2.5')
pm10 = max_polluted_day('PM10')
```

0.9.1 In 2022, Delhi had the highest PM2.5(g/m3) & PM10(g/m3) level in these 10 days.

[413]:
```
display(pm2_5)
display(pm10)
```

```
<pandas.io.formats.style.Styler at 0x7f39893d45b0>


<pandas.io.formats.style.Styler at 0x7f398a1a69d0>
```

Note:

- It can easily be observed that the most polluted days tend to fall in the months of November and January.
- This is likely due to the phenomenon of atmospheric inversion, which is more common in the winter months.
- During an atmospheric inversion, a layer of warmer air traps cooler air near the surface, preventing the dispersion of pollutants and leading to an accumulation of pollutants in the lower atmosphere.

0.10 8.Correlation Statistics for the Heatmap:

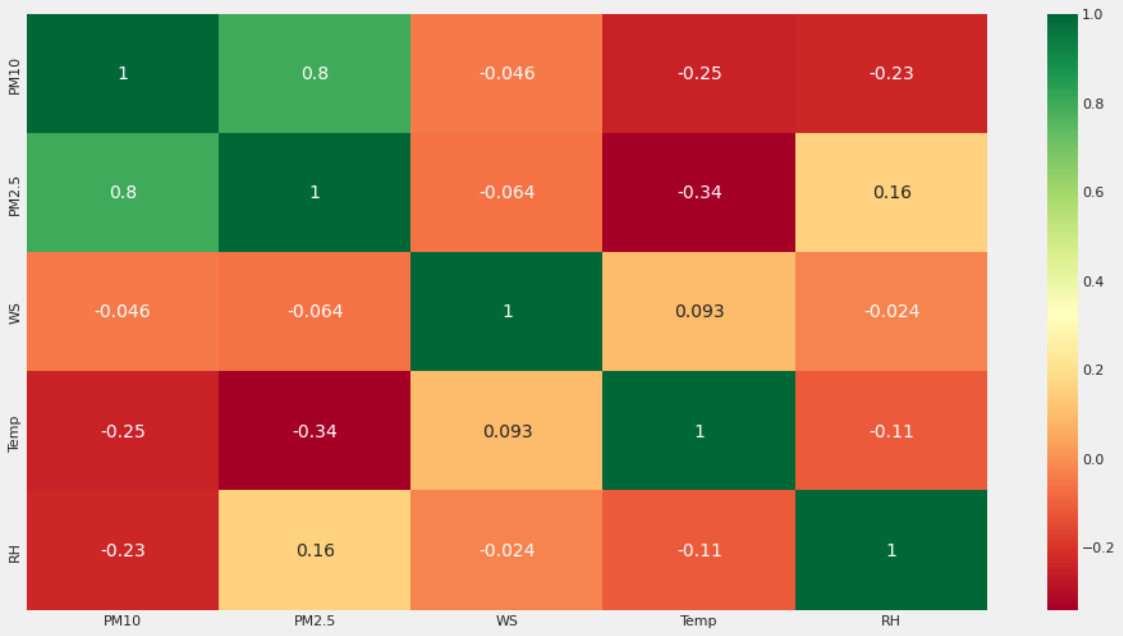
Representing the relationship between P.M. and Meterological parameters.

Taking daily dataset which will have

only appropriate parameters for heatmap

[414]:
```
relation= dfdelhiD.corr()
```

[415]:
```
top_corr_features = relation.index
plt.figure(figsize=(15,8))
#plot heat map
g=sns.heatmap(dfdelhiD[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```



Note:

- In this heatmap, the cell with a value of -0.25 between PM10 and temperature is also likely colored a shade of red, indicating a relatively low value.
- This suggests that there is a negative correlation between PM10 levels and either relative humidity or temperature, which is also being verified by the fact of winter inversion.

0.11 9.Wind Speed vs particulate Matter

Taking Delhi data of this year having Monthly AVG values with 2 column pm2.5 and Wind Speed with dataset of name - df_WS

[416]:
```
df_WS = dfdelhiM[['PM2.5','WS']]
```

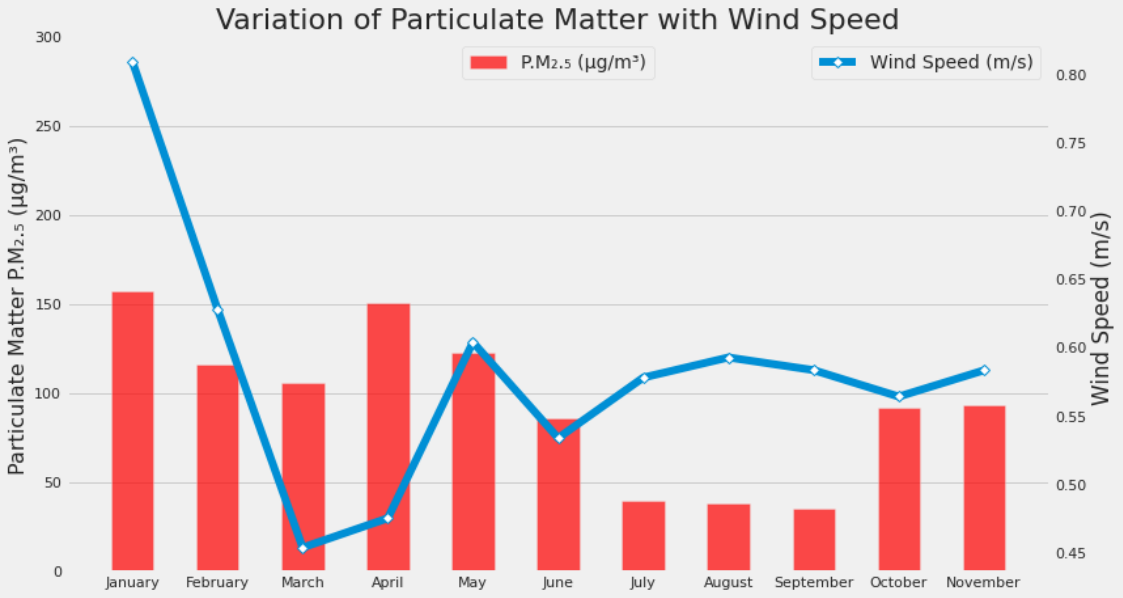
[417]:
```
df_WS=df_WS['2022-01-01':'2022-12-01']
df_WS
```

[417]:

Date	PM2.5	WS
2022-01-31	157.151953	0.809641
2022-02-28	116.574419	0.628441
2022-03-31	106.154023	0.453762
2022-04-30	150.764453	0.475240
2022-05-31	123.239931	0.604066
2022-06-30	86.030569	0.533944
2022-07-31	40.183093	0.578187

```
2022-08-31   38.715691  0.592938
2022-09-30   35.705973  0.583648
2022-10-31   92.306764  0.564274
2022-11-30   93.495051  0.583492
```

[418]:
```
# Create figure and axis #1
plt.style.use("fivethirtyeight")
fig, ax = plt.subplots(figsize=(12,7))

x= np.array(['January', 'February', 'March', 'April','May', 'June', 'July',␣
 ↪'August', 'September', 'October', 'November'])
ax.set_title("Variation of Particulate Matter with Wind Speed", fontsize=22)

ax.bar(x,
        df_WS['PM2.5'],
        label="P.M .  (µg/m³)",
        color='red',
       width=0.5,
       alpha=0.7)
ax.set_ylim(0, 300)
ax.set_ylabel('Particulate Matter P.M .  (µg/m³)', fontsize=16)
ax.legend(loc="upper center",fontsize=14)
ax.xaxis.grid(False)

# second Chart
ax2 = ax.twinx()

ax2.plot(x,
        df_WS['WS'],
         linewidth=6,
        label="Wind Speed (m/s)",
        marker='D',mfc='white')

ax2.set_ylabel('Wind Speed (m/s) ')
ax2.legend(loc="upper right",fontsize=14)
ax2.grid(False)

plt.savefig("PM.jpeg" , dpi = 180)

plt.show()
```



Note:

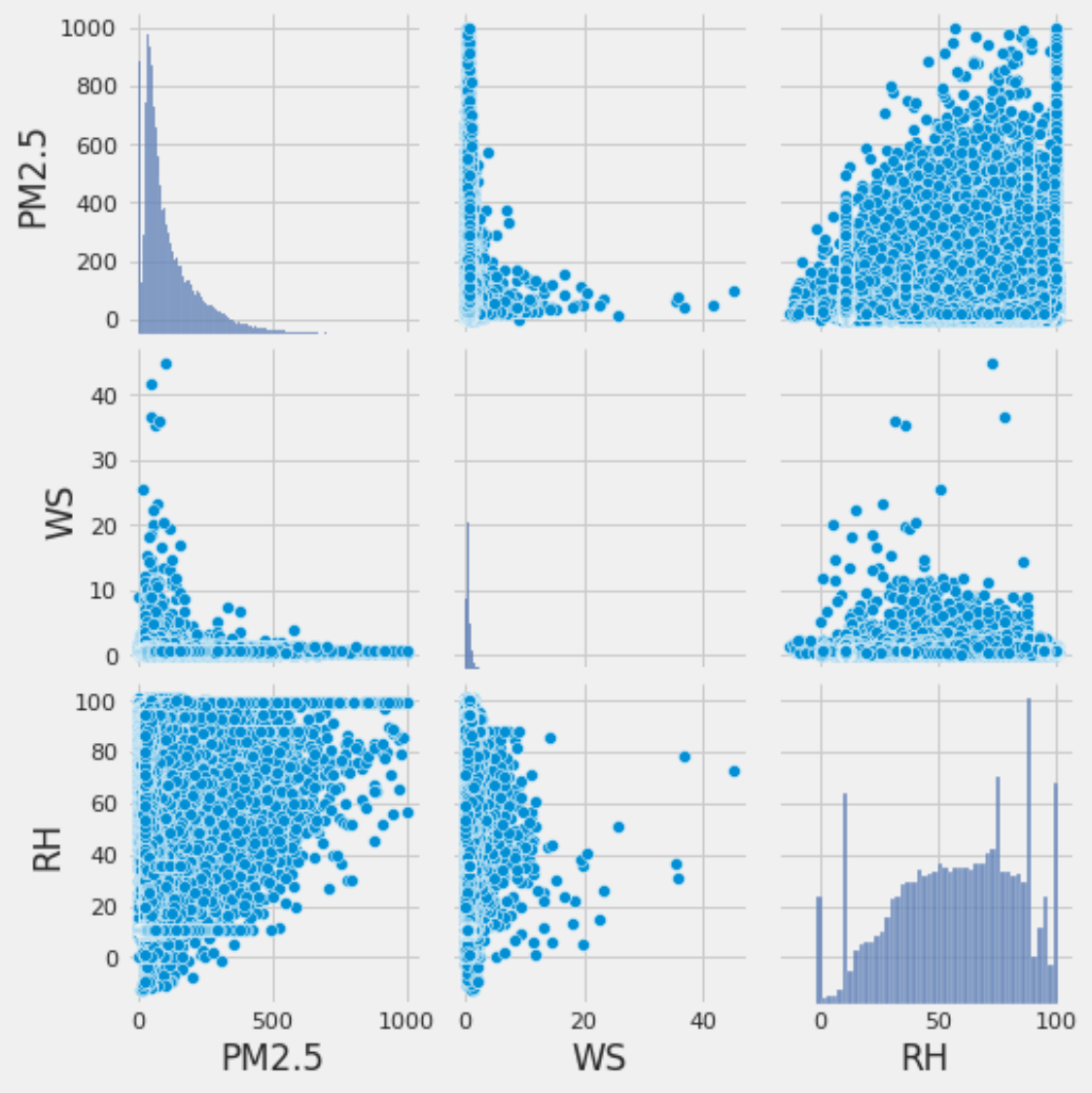
- During the months of July to September, the wind velocity in Delhi is generally higher, which helps to disperse the PM2.5 particles and reduce their concentration in the air.
- As a result, the PM2.5 levels tend to decrease during this period.
- On the other hand, in April, the wind velocity is generally lower, which means that the PM2.5 particles are not as effectively dispersed and their levels tend to increase.

0.12 10.Pair_Plot

Taking Hourly data dfdelhiH having data from 2017 Jan to 2022 Nov

[419]:
```
sns.pairplot(dfdelhiH[['PM2.5','WS','RH']])
```

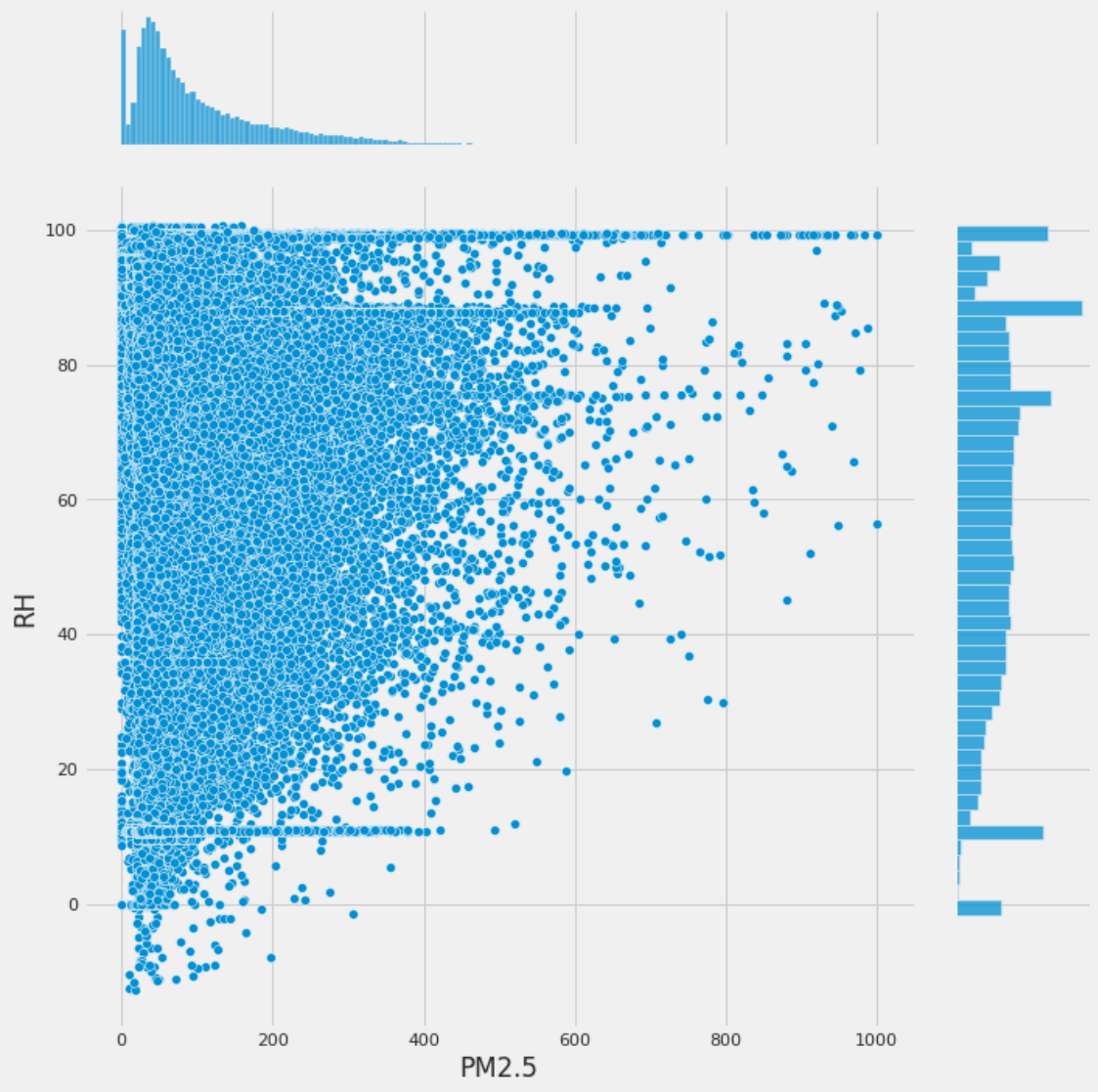
[419]:
```
<seaborn.axisgrid.PairGrid at 0x7f39852c4370>
```



0.13 11.Jointplot

[420]:
```
sns.jointplot( x='PM2.5',
               y='RH',
               data= dfdelhiH,
               height=10, space=0.3
)
```

[420]: <seaborn.axisgrid.JointGrid at 0x7f3986391970>



0.14 12 Distplot

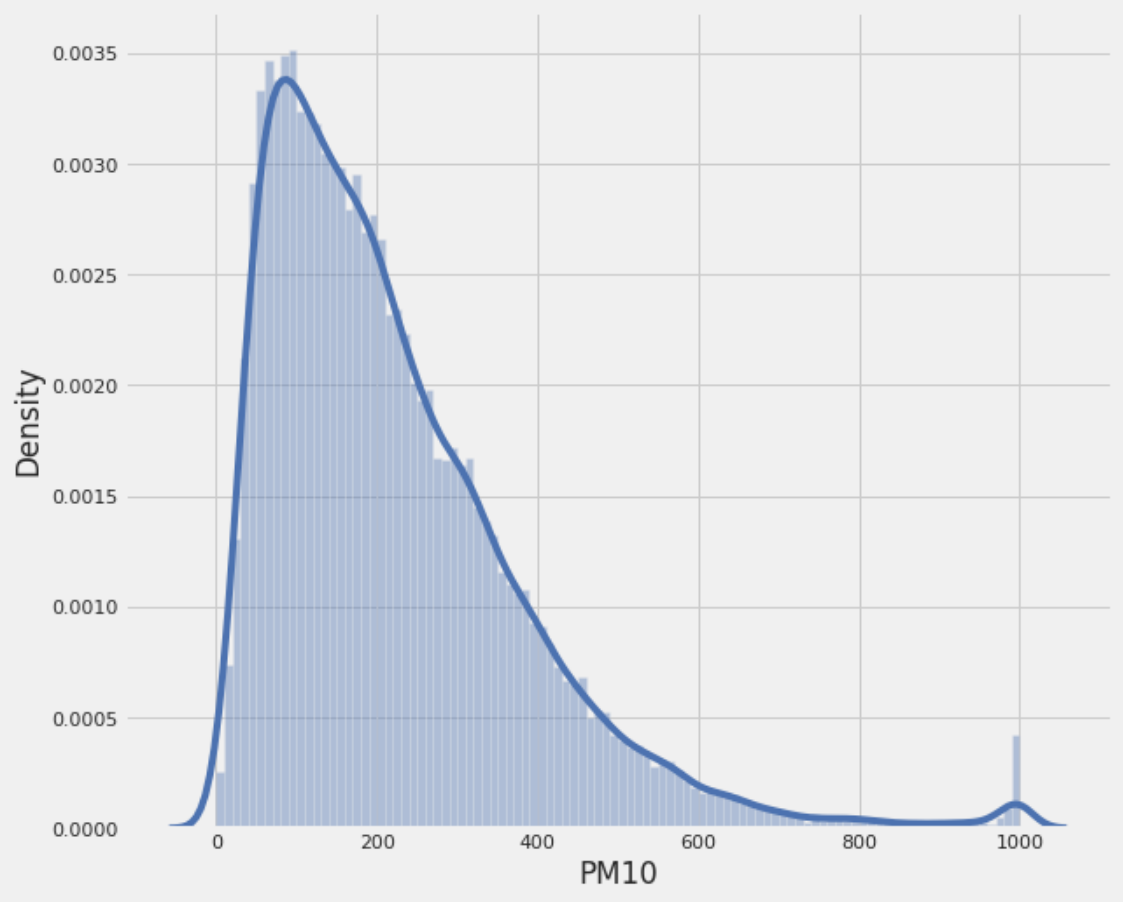
[421]:
```
col= dfdelhiH.columns
```

[422]:
```
col
```

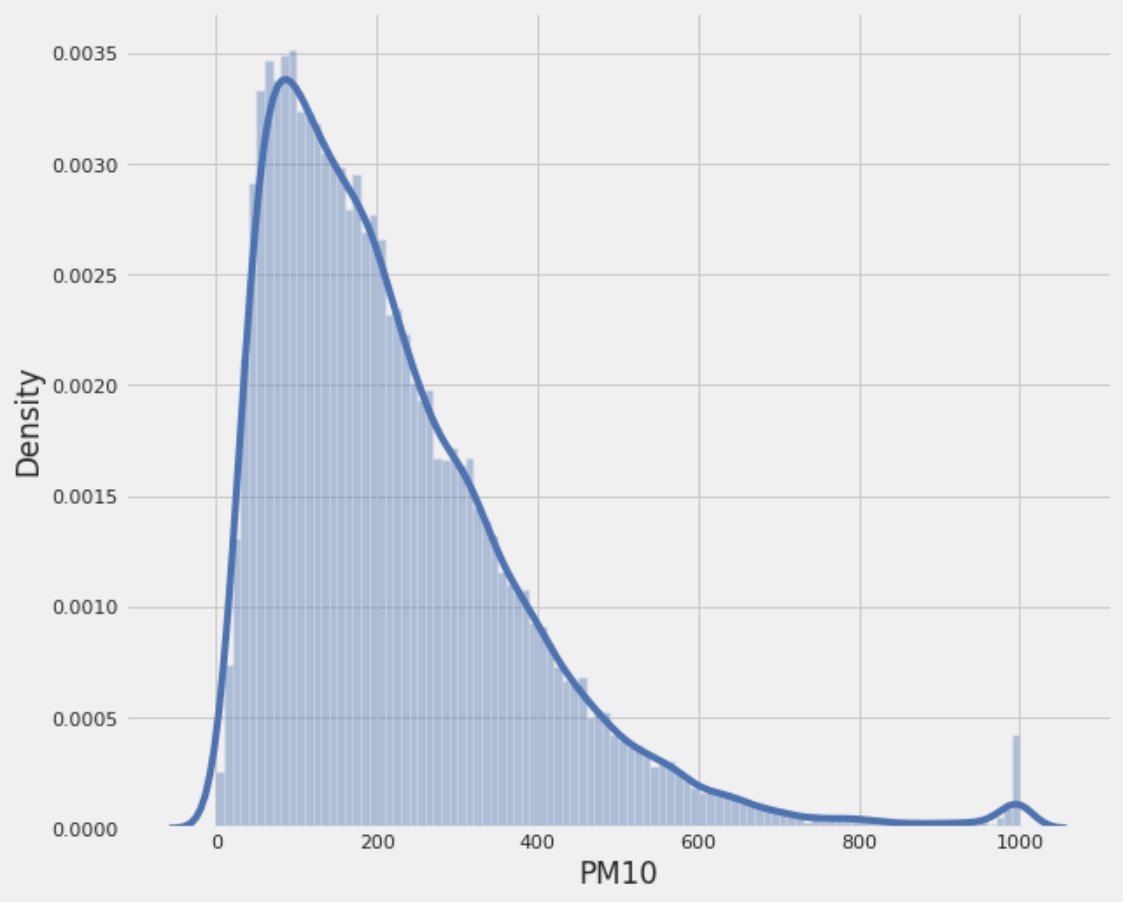
[422]:
```
Index(['PM10', 'PM2.5', 'WS', 'Temp', 'RH'], dtype='object')
```

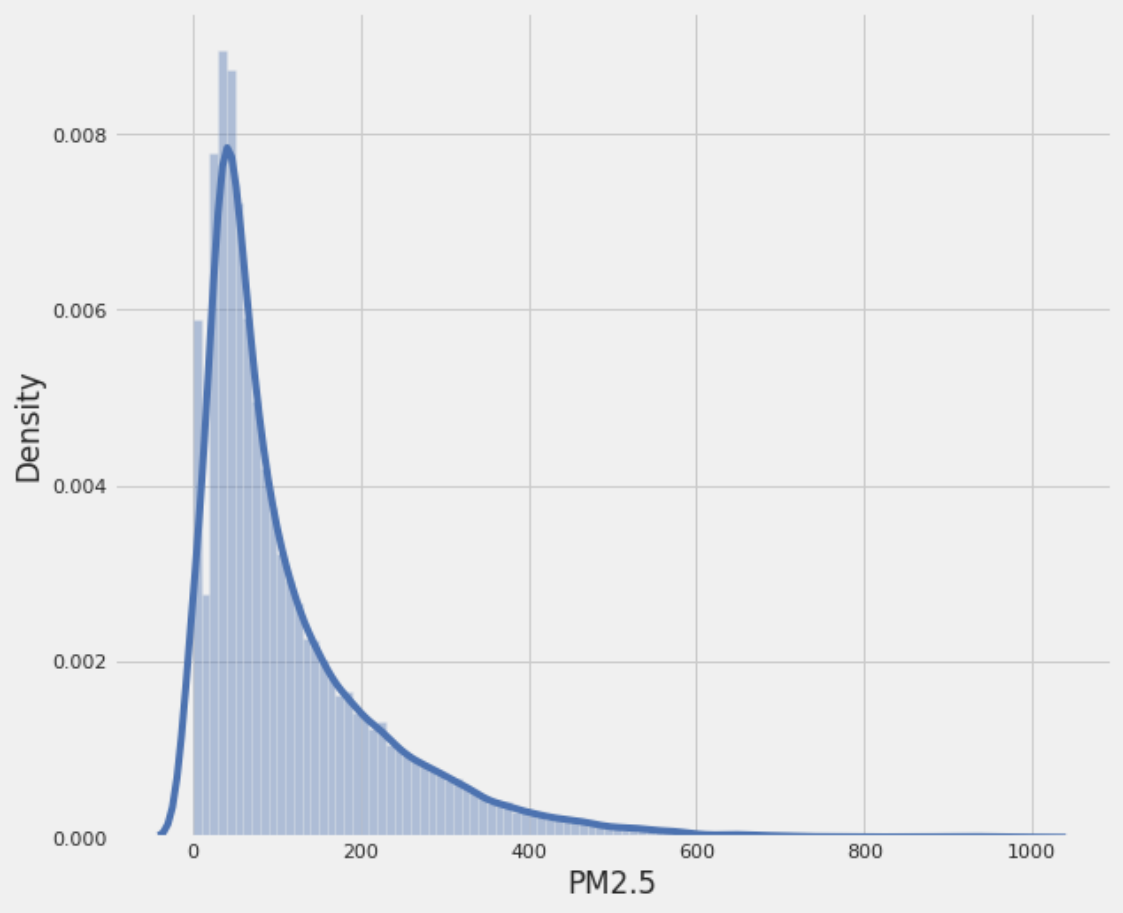
single parameters distplot

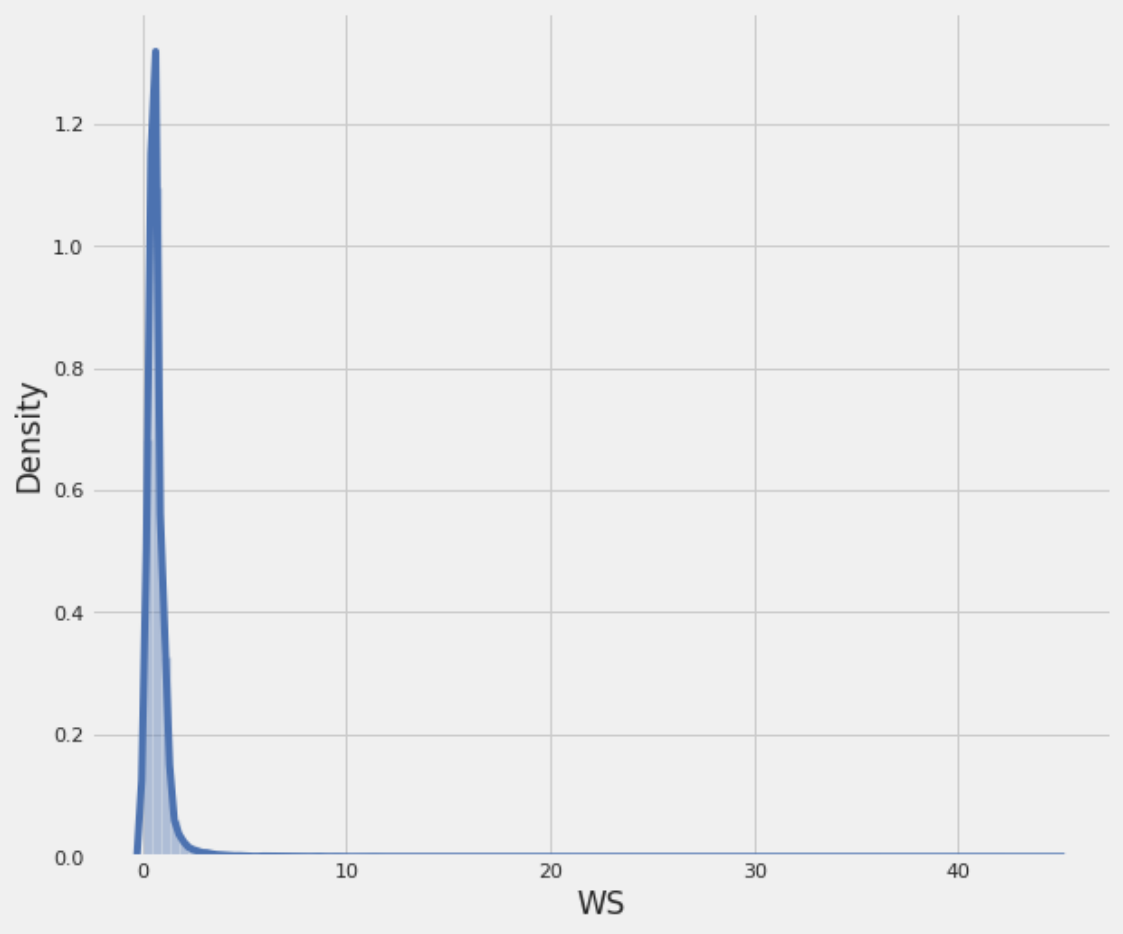
[423]:
```
plt.figure(figsize=(9, 8))
sns.distplot(dfdelhiH['PM10'], color='b', bins=100, hist_kws={'alpha': 0.4});
```

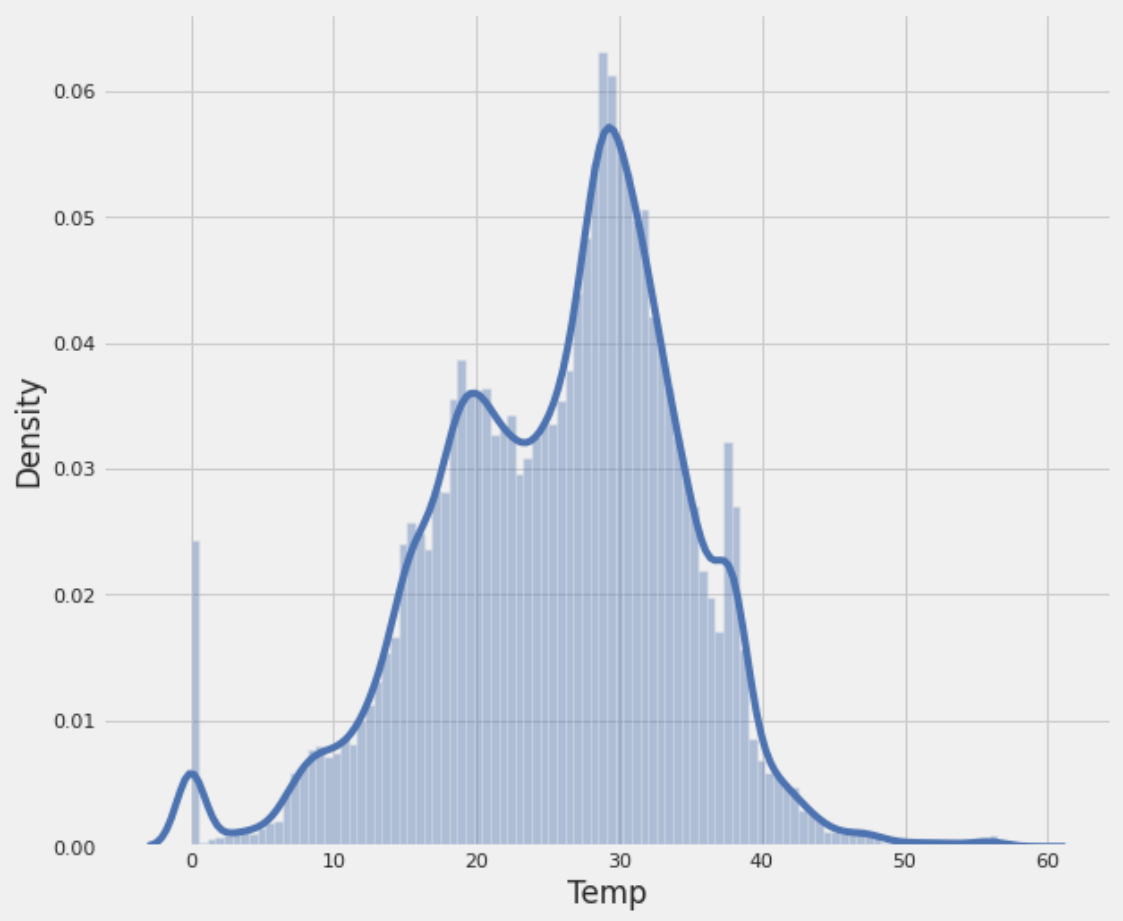


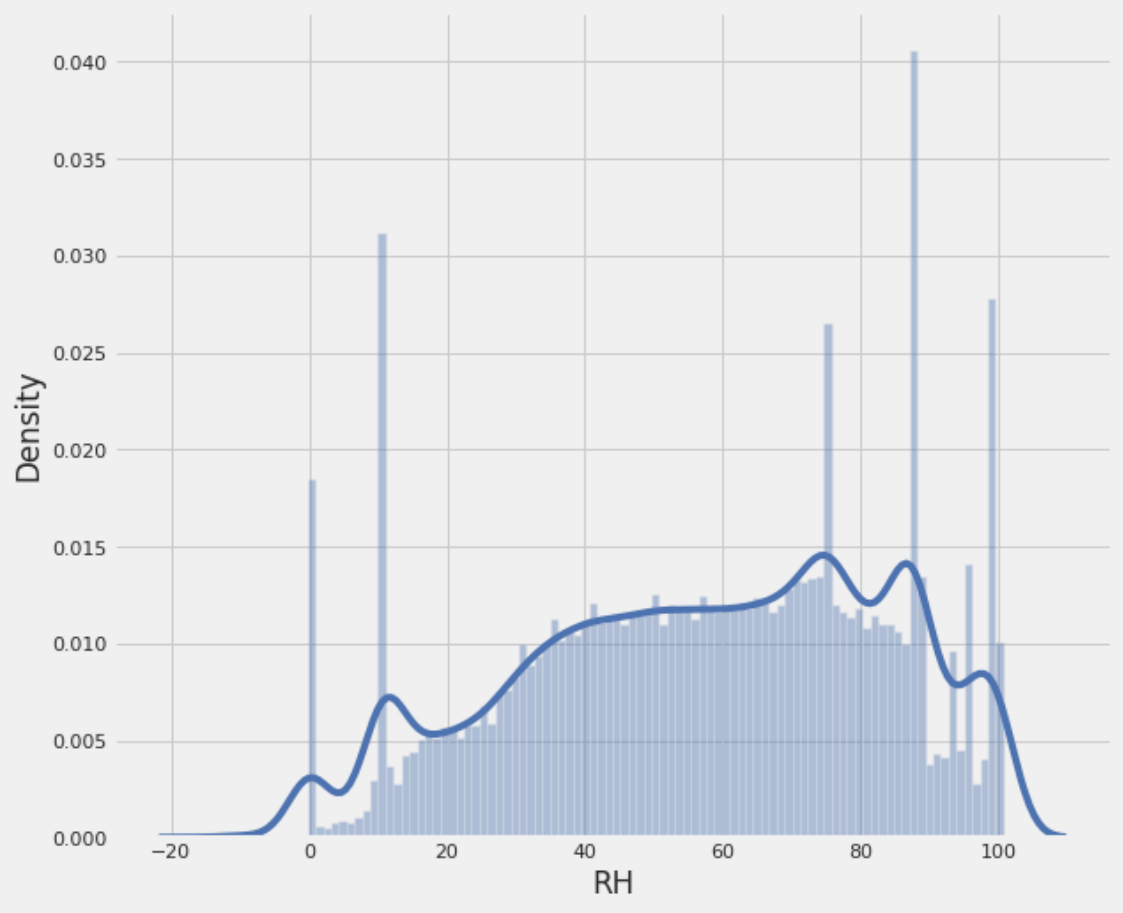
[424]:
```
for i in col:
    plt.figure(figsize=(9, 8))
    sns.distplot(dfdelhiH[i], color='b', bins=100, hist_kws={'alpha': 0.4})
```


0.0035
0.0030
0.0025
0.0020
0.0015
0.0010
0.0005
0.0000
Density
0
200
400
600
800
1000
PM10


0.008
0.006
0.004
0.002
0.000
Density
0
200
400
600
800
1000
PM2.5


Density
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
WS


0.06
0.05
0.04
0.03
0.02
0.01
0.00
Density
0
10
20
30
40
50
60
Temp


0.040
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Density
−20
0
20
40
60
80
100
RH