## **NOISE POLLUTION MONITORING**

------------------------------------------------------------------------------------------------------------------------------

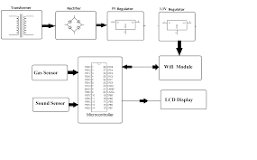
**Phase 3:Development part 1**

**Topic: Section begin building your project by loading and preprocessing the dataset.**

**Introduction:**

Greetings from the Kaggle bot! This is an automatically-generated kernel with starter code demonstrating how to read in the data and begin exploring. If you're inspired to dig deeper, click the blue "Fork Notebook" button at the top of this kernel to begin editing.

**Monitor noise pollution in IoT**



System uses air sensors to sense presence of harmful gases/compounds in the air and constantly transmit this data to microcontroller. Also system keeps measuring sound level and reports it to the online server over IOT. The sensors interact with microcontroller which processes this data and transmits it over internet.

## **Exploratory Analysis**

To begin this exploratory analysis, first import libraries and define functions for plotting the data using matplotlib. Depending on the data, not all plots will be made. (Hey, I'm just a simple kerneling bot, not a Kaggle Competitions Grandmaster!)

In [1]:

from mpl\_toolkits.mplot3d import Axes3D  
from sklearn.preprocessing import StandardScaler  
import matplotlib.pyplot as plt *# plotting*  
*import* numpy as np *# linear algebra*  
*import* os *# accessing directory structure*  
*import* pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

There are 2 csv files in the current version of the dataset:

In [2]:

for dirname, \_, filenames **in** os.walk('/kaggle/input'):  
 for filename **in** filenames:  
 print(os.path.join(dirname, filename))

/kaggle/input/stations.csv  
/kaggle/input/station\_month.csv

The next hidden code cells define functions for plotting data. Click on the "Code" button in the published kernel to reveal the hidden code.

unfold\_moreShow hidden code

unfold\_moreShow hidden code

unfold\_moreShow hidden code

Now you're ready to read in the data and use the plotting functions to visualize the data.

### Let's check 1st file: /kaggle/input/station\_month.csv

In [6]:

nRowsRead = 1000 *# specify 'None' if want to read whole file*  
*# station\_month.csv may have more rows in reality, but we are only loading/previewing the first 1000 rows*  
*df1* = pd.read\_csv('/kaggle/input/station\_month.csv', delimiter=',', nrows = nRowsRead)  
df1.dataframeName = 'station\_month.csv'  
nRow, nCol = df1.shape  
print(f'There are **{nRow}** rows and **{nCol}** columns')

There are 840 rows and 7 columns

Let's take a quick look at what the data looks like:

In [7]:

df1.head(5)

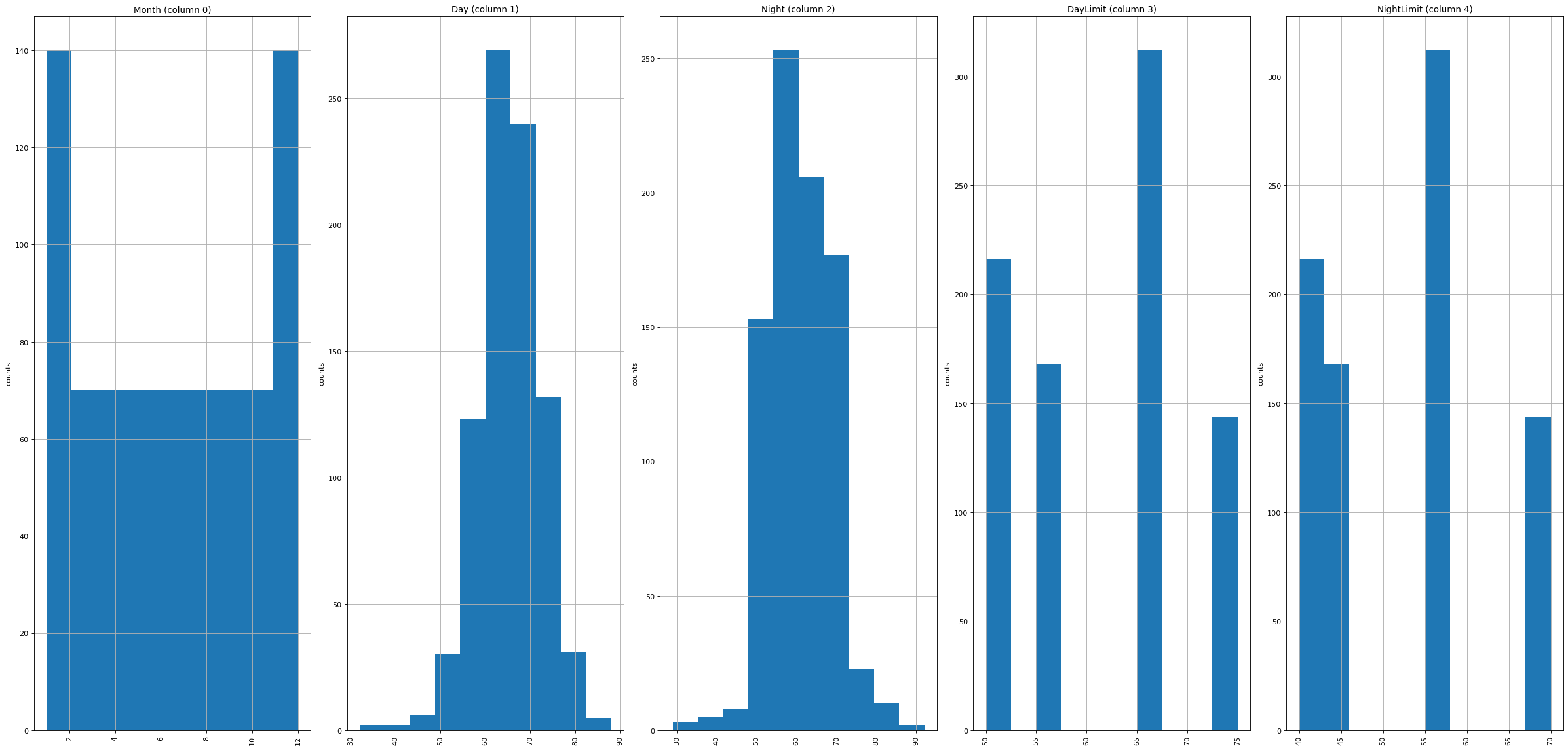
Out[7]:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Station | Year | Month | Day | Night | DayLimit | NightLimit |
| 0 | DEL01 | 2018 | 1 | 53 | 52 | 50 | 40 |
| 1 | DEL01 | 2018 | 2 | 53 | 53 | 50 | 40 |
| 2 | DEL01 | 2018 | 3 | 53 | 53 | 50 | 40 |
| 3 | DEL01 | 2018 | 4 | 55 | 58 | 50 | 40 |
| 4 | DEL01 | 2018 | 5 | 55 | 57 | 50 | 40 |

**Distribution graphs (histogram/bar graph) of sampled columns:**

In [8]:

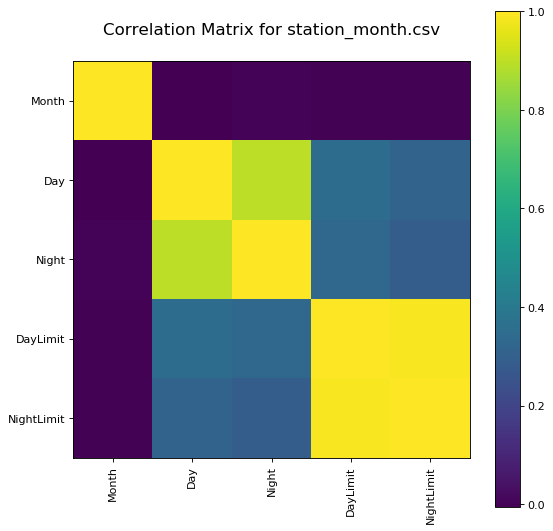
plotPerColumnDistribution(df1, 10, 5)



Correlation matrix:

In [9]:

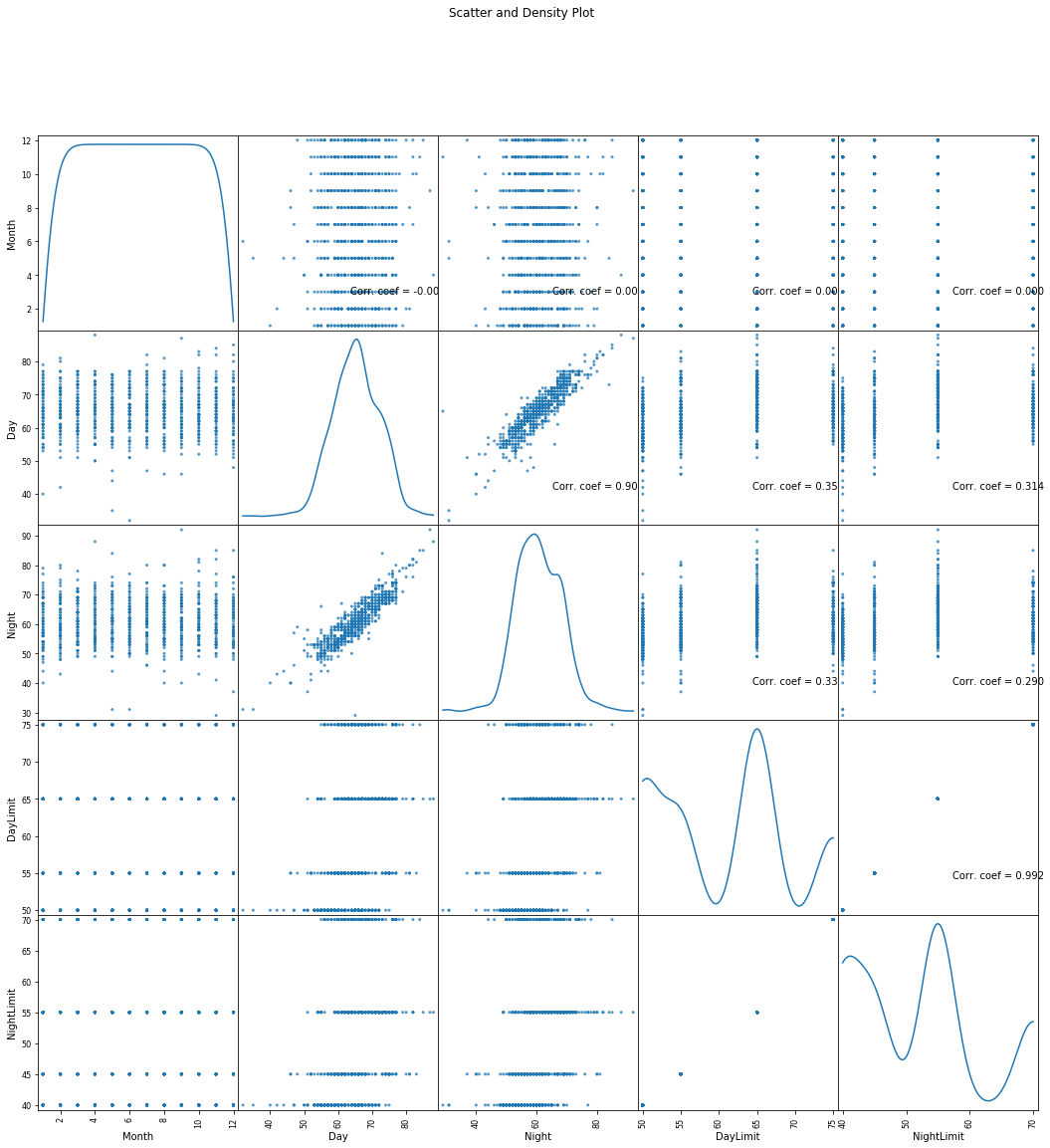
plotCorrelationMatrix(df1, 8)



Scatter and density plots:

In [10]:

plotScatterMatrix(df1, 18, 10)



### Let's check 2nd file: /kaggle/input/stations.csv

In [11]:

nRowsRead = 1000 *# specify 'None' if want to read whole file*  
*# stations.csv may have more rows in reality, but we are only loading/previewing the first 1000 rows*  
*df2* = pd.read\_csv('/kaggle/input/stations.csv', delimiter=',', nrows = nRowsRead)  
df2.dataframeName = 'stations.csv'  
nRow, nCol = df2.shape  
print(f'There are **{nRow}** rows and **{nCol}** columns')

There are 70 rows and 5 columns

Let's take a quick look at what the data looks like:

In [12]:

df2.head(5)

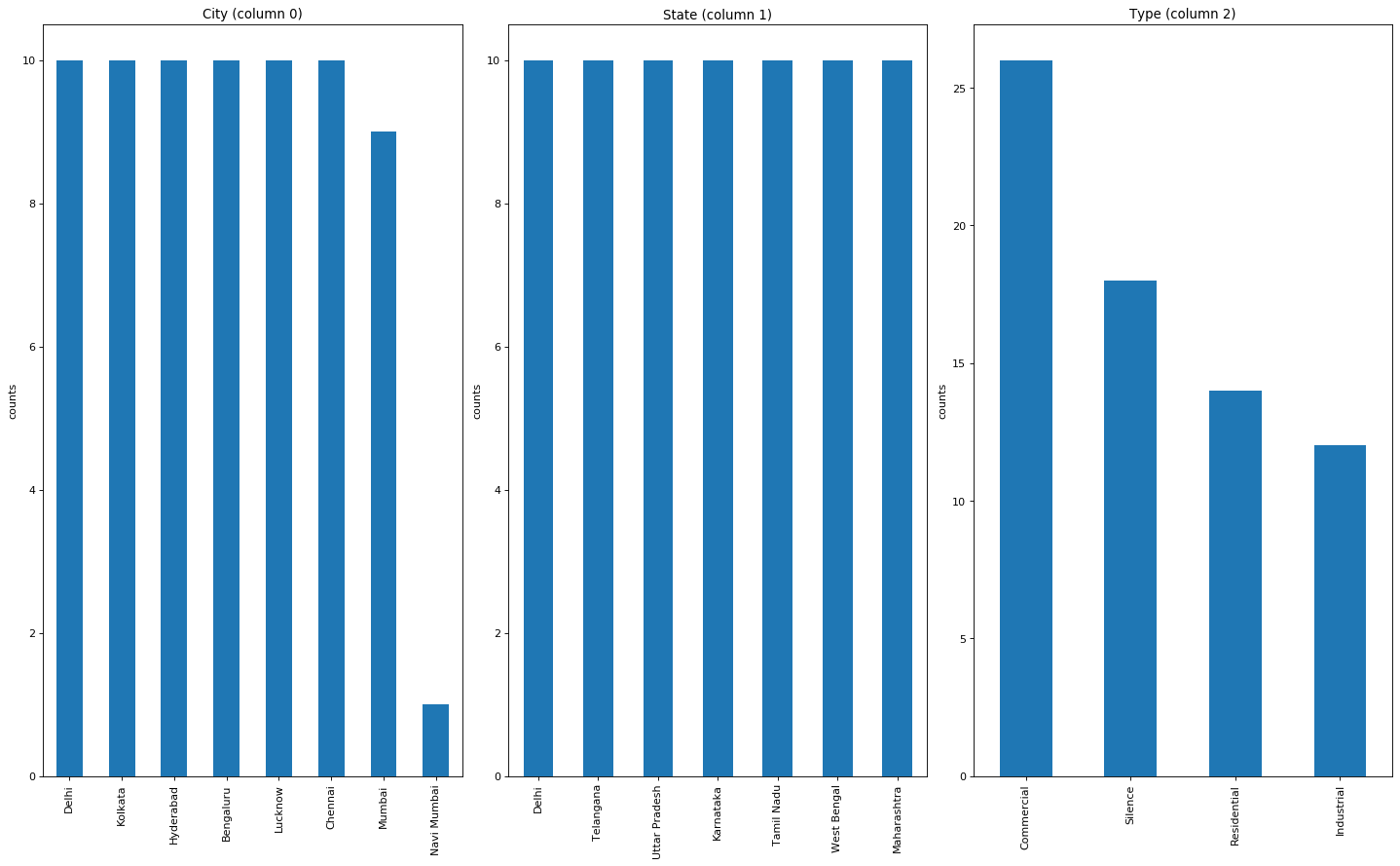
Out[12]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Station | Name | City | State | Type |
| 0 | DEL01 | Dilshad Garden | Delhi | Delhi | Silence |
| 1 | DEL02 | CPCB, HQ | Delhi | Delhi | Commercial |
| 2 | DEL03 | DCE, Bawana | Delhi | Delhi | Silence |
| 3 | DEL04 | ITO | Delhi | Delhi | Commercial |
| 4 | DEL05 | NSIT, Dwarka | Delhi | Delhi | Silence |

Distribution graphs (histogram/bar graph) of sampled columns:

In [13]:

plotPerColumnDistribution(df2, 10, 5)



Code :

from machine import Pin, ADC

from time import sleep

pot = ADC(Pin(2))

pot.atten(ADC.ATTN\_11DB) #Full range: 3.3v

#ADC.ATTN\_0DB: Maximum voltage of 1.2V

#ADC.ATTN\_2\_5DB: Maximum voltage of 1.5V

#ADC.ATTN\_6DB: Maximum voltage of 2.0V

#ADC.ATTN\_11DB: Maximum voltage of 3.3V

while True:

pot\_value = pot.read()

print(pot\_value)

sleep(0.1)

'''

import machine, time

a = machine.ADC(machine.Pin(32))

while True:

sample = a.read() # we want 16 bits, a.read() returns 10 bits

print(sample)

time.sleep(1/44100)

## **Conclusion:**

This concludes your starter analysis! To go forward from here, click the blue "Fork Notebook" button at the top of this kernel. This will create a copy of the code and environment for you to edit. Delete, modify, and add code as you please. Happy Kaggling!