+ Code + Text

Data Uploding

```
import pandas as pd
x=pd.read_csv("/content/cpcb_dly_aq_tamil_nadu-2014.csv")
```

```
x
```

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type o Locatio
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industri Are
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industri Are
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industri Are
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industri Are
4	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industri Are
...	...	...	...	...	...	...	
2874	773	12-03-14	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residentia Rural an other Area
						Tamilnadu	

Data Pre-Processing

Checking for null values

```
x.isnull().sum()
```

```
Stn Code                          0
Sampling Date                     0
State                             0
City/Town/Village/Area            0
Location of Monitoring Station    0
Agency                            0
Type of Location                  0
SO2                              11
NO2                              13
RSPM/PM10                         4
PM 2.5                         2879
dtype: int64
```

```
x["SO2"]
```

```
0       11.0
1       13.0
2       12.0
3       15.0
4       13.0
```

```
        ...
2874    15.0
2875    12.0
2876    19.0
2877    15.0
2878    14.0
Name: SO2, Length: 2879, dtype: float64
```

```
import numpy as np
```

Replacing null value with mean

```
x["SO2"]=x["SO2"].fillna(x["SO2"].mean())
```

```
x["SO2"]
```

```
0       11.0
1       13.0
2       12.0
3       15.0
4       13.0
        ...
2874    15.0
2875    12.0
2876    19.0
2877    15.0
2878    14.0
Name: SO2, Length: 2879, dtype: float64
```

```
x["NO2"]
```

```
0       17.0
1       17.0
2       18.0
3       16.0
4       14.0
        ...
2874    18.0
2875    14.0
2876    22.0
2877    17.0
2878    16.0
Name: NO2, Length: 2879, dtype: float64
```

```
x["NO2"]=x["NO2"].fillna(x["NO2"].mean())
```

```
x.isnull().sum()
```

```
Stn Code                            0
Sampling Date                       0
State                               0
City/Town/Village/Area              0
Location of Monitoring Station      0
Agency                              0
Type of Location                    0
SO2                                 0
NO2                                 0
RSPM/PM10                           4
PM 2.5                           2879
dtype: int64
```

```
x["RSPM/PM10"]
```

```
0        55.0
1        45.0
2        50.0
3        46.0
4        42.0
        ...
2874    102.0
2875     91.0
2876    100.0
2877     95.0
2878     94.0
Name: RSPM/PM10, Length: 2879, dtype: float64
```

```
x["RSPM/PM10"]=x["RSPM/PM10"].fillna(x["RSPM/PM10"].mean())
```

```
x.drop("PM 2.5",axis=1,inplace=True)
```

Standard scalling

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
x[['SO2', 'NO2', 'RSPM/PM10']] = scaler.fit_transform(x[['SO2', 'NO2', 'RSPM/PM10']])
```

Feature Engineering

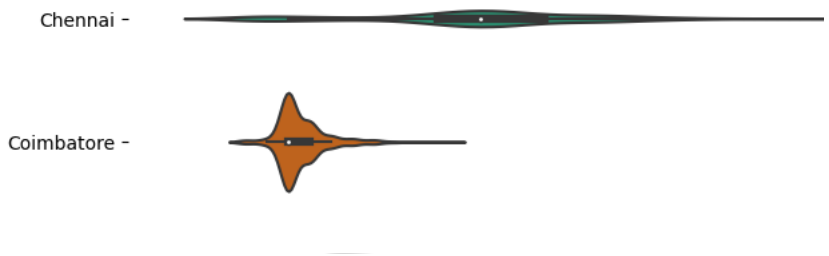
```
x['Sampling Date'] = pd.to_datetime(x['Sampling Date'], format='%d-%m-%y')
```

```
x['Day'] = x['Sampling Date'].dt.day
x['Month'] = x['Sampling Date'].dt.month
x['Year'] = x['Sampling Date'].dt.year
```

Data Visualization

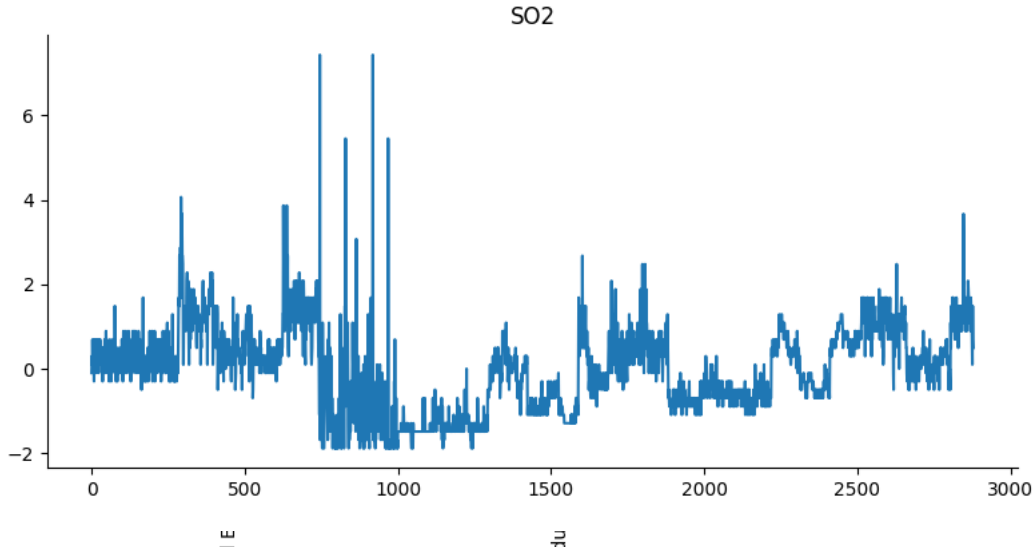
```
import matplotlib.pyplot as plt
```

```
import numpy as np
from google.colab import autoviz

def violin_plot(df, value_colname, facet_colname, figscale=1, mpl_palette_name='Dark2', **kwargs):
  from matplotlib import pyplot as plt
  import seaborn as sns
  figsize = (12 * figscale, 1.2 * figscale * len(df[facet_colname].unique()))
  plt.figure(figsize=figsize)
  sns.violinplot(df, x=value_colname, y=facet_colname, palette=mpl_palette_name, **kwargs)
  sns.despine(top=True, right=True, bottom=True, left=True)
  return autoviz.MplChart.from_current_mpl_state()

chart = violin_plot(x, *['SO2', 'City/Town/Village/Area'], **{'inner': 'box'})
chart
```

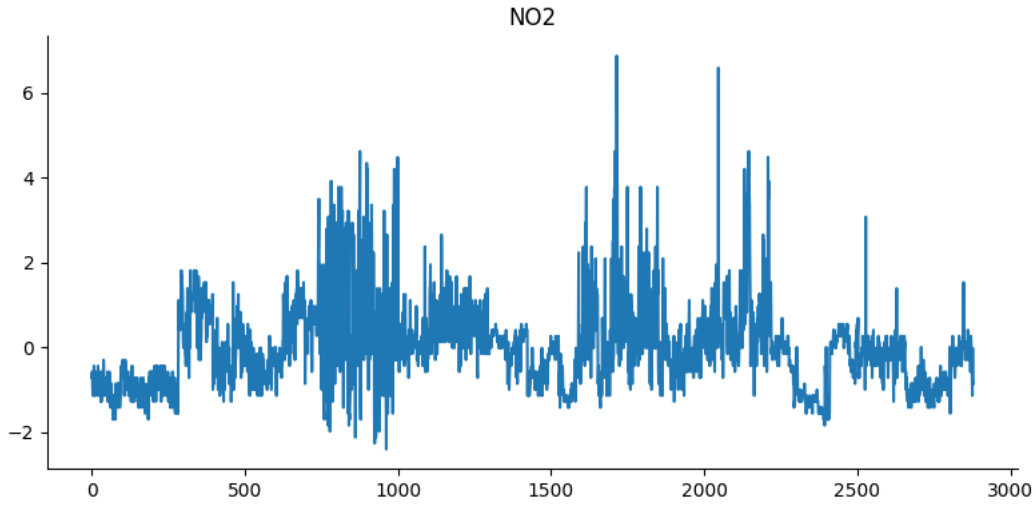



```
import numpy as np
from google.colab import autoviz

def heatmap(df, x_colname, y_colname, figscale=1, mpl_palette_name='viridis'):
  from matplotlib import pyplot as plt
  import seaborn as sns
  import pandas as pd
  plt.subplots(figsize=(8 * figscale, 8 * figscale))
  df_2dhist = pd.DataFrame({
      x_label: grp[y_colname].value_counts()
      for x_label, grp in df.groupby(x_colname)
  })
  sns.heatmap(df_2dhist, cmap=mpl_palette_name)
  plt.xlabel(x_colname)
  plt.ylabel(y_colname)
  return autoviz.MplChart.from_current_mpl_state()

chart = heatmap(x, *['Agency', 'Type of Location'], **{})
chart
```

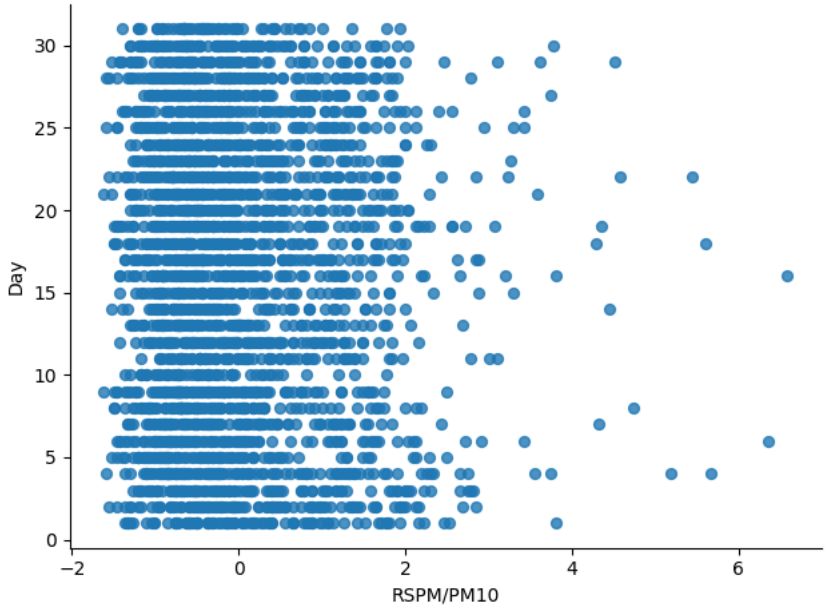

```
import numpy as np
from google.colab import autoviz

def value_plot(df, y, figscale=1):
  from matplotlib import pyplot as plt
  df[y].plot(kind='line', figsize=(8 * figscale, 4 * figscale), title=y)
  plt.gca().spines[['top', 'right']].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = value_plot(x, *['SO2'], **{})
chart
```




```
import numpy as np
from google.colab import autoviz

def value_plot(df, y, figscale=1):
  from matplotlib import pyplot as plt
  df[y].plot(kind='line', figsize=(8 * figscale, 4 * figscale), title=y)
  plt.gca().spines[['top', 'right']].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = value_plot(x, *['NO2'], **{})
chart
```

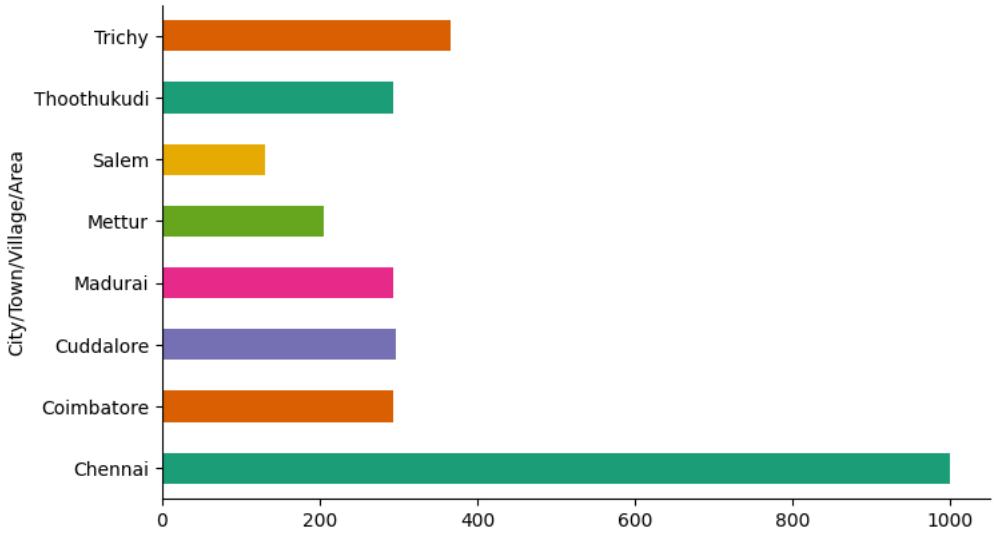


```
import numpy as np
from google.colab import autoviz

def scatter_plot(df, x_colname, y_colname, figscale=1, alpha=.8):
  from matplotlib import pyplot as plt
  plt.figure(figsize=(6 * figscale, 6 * figscale))
  df.plot(kind='scatter', x=x_colname, y=y_colname, s=(32 * figscale), alpha=alpha)
  plt.gca().spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = scatter_plot(x, *['RSPM/PM10', 'Day'], **{})
chart
```

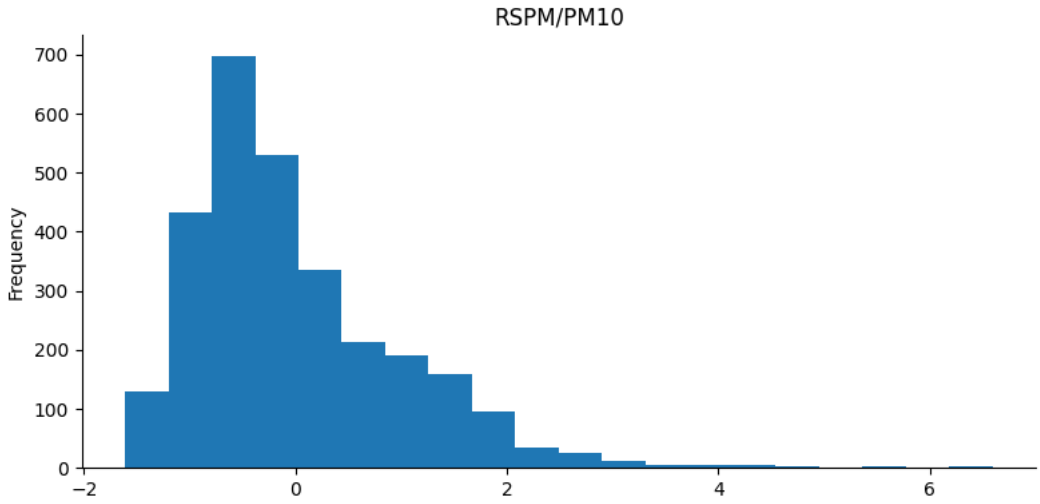



```
<Figure size 600x600 with 0 Axes>
```

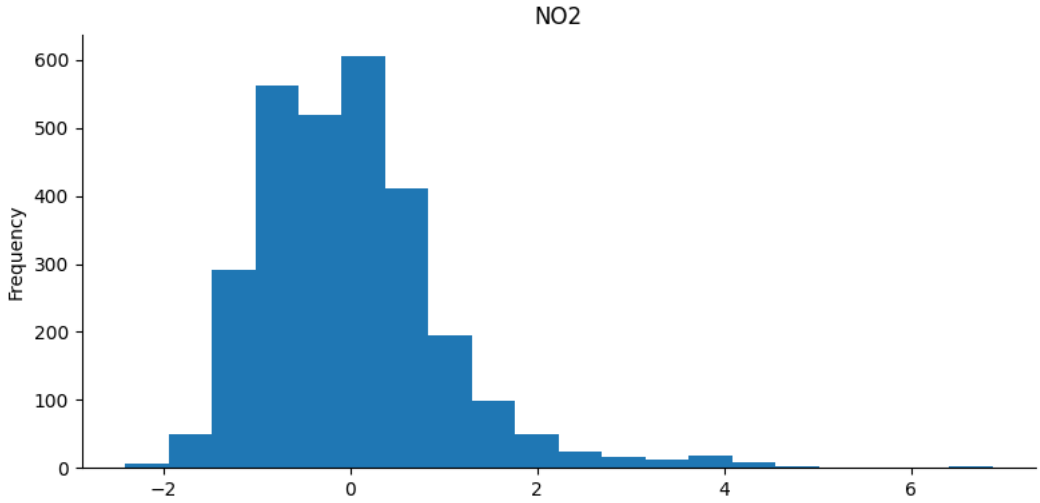
```
import numpy as np
from google.colab import autoviz

def categorical_histogram(df, colname, figscale=1, mpl_palette_name='Dark2'):
  from matplotlib import pyplot as plt
  import seaborn as sns
  df.groupby(colname).size().plot(kind='barh', color=sns.palettes.mpl_palette(mpl_palette_name), figsize=(8*figscale, 4
  plt.gca().spines[['top', 'right',]].set_visible(False)
  return autoviz.MplChart.from_current_mpl_state()

chart = categorical_histogram(x, *['City/Town/Village/Area'], **{})
chart
```



```
import numpy as np
from google.colab import autoviz

def histogram(df, colname, num_bins=20, figscale=1):
  from matplotlib import pyplot as plt
  df[colname].plot(kind='hist', bins=num_bins, title=colname, figsize=(8*figscale, 4*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = histogram(x, *['RSPM/PM10'], **{})
chart
```




```
import numpy as np
from google.colab import autoviz

def histogram(df, colname, num_bins=20, figscale=1):
  from matplotlib import pyplot as plt
  df[colname].plot(kind='hist', bins=num_bins, title=colname, figsize=(8*figscale, 4*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
  plt.tight_layout()
  return autoviz.MplChart.from_current_mpl_state()

chart = histogram(x, *['NO2'], **{})
chart
```




```
import numpy as np
from google.colab import autoviz
import matplotlib.pyplot as plt

def histogram(df, colname, num_bins=20, figscale=1):
  from matplotlib import pyplot as plt
  df[colname].plot(kind='hist', bins=num_bins, title=colname, figsize=(8*figscale, 4*figscale))
  plt.gca().spines[['top', 'right',]].set_visible(False)
```

```
    plt.tight_layout()
    return autoviz.MplChart.from_current_mpl_state()

chart = histogram(x, *['SO2'], **{})
chart
```