```
!pip install pandas matplotlib seaborn
```

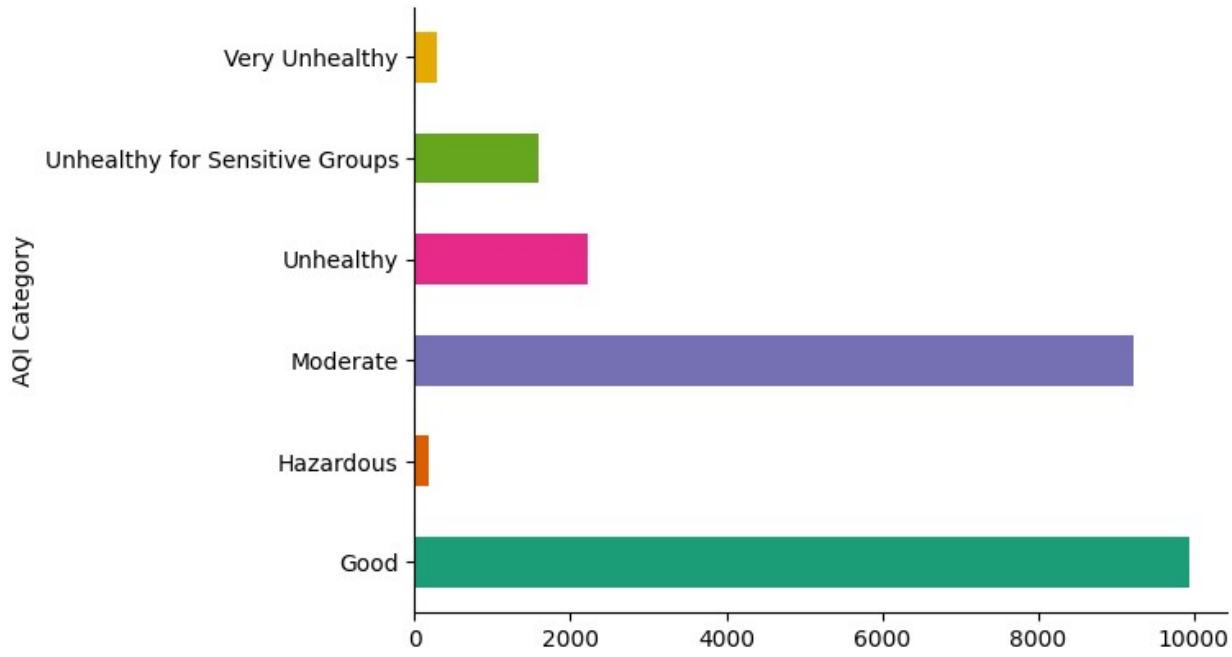
```
Requirement already satisfied: pandas in
/usr/local/lib/python3.10/dist-packages (2.0.3)
Requirement already satisfied: matplotlib in
/usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: seaborn in
/usr/local/lib/python3.10/dist-packages (0.13.1)
Requirement already satisfied: python-dateutil>=2.8.2 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: tzdata>=2022.1 in
/usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: numpy>=1.21.0 in
/usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (4.53.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.5)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (24.1)
Requirement already satisfied: pillow>=6.2.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2-
>pandas) (1.16.0)
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objects as go
from plotly.subplots import make_subplots

weather_data = pd.read_csv("/Weater_Data.csv")

pollution_data = pd.read_csv("/global air pollution dataset.csv")

pollution_data
```

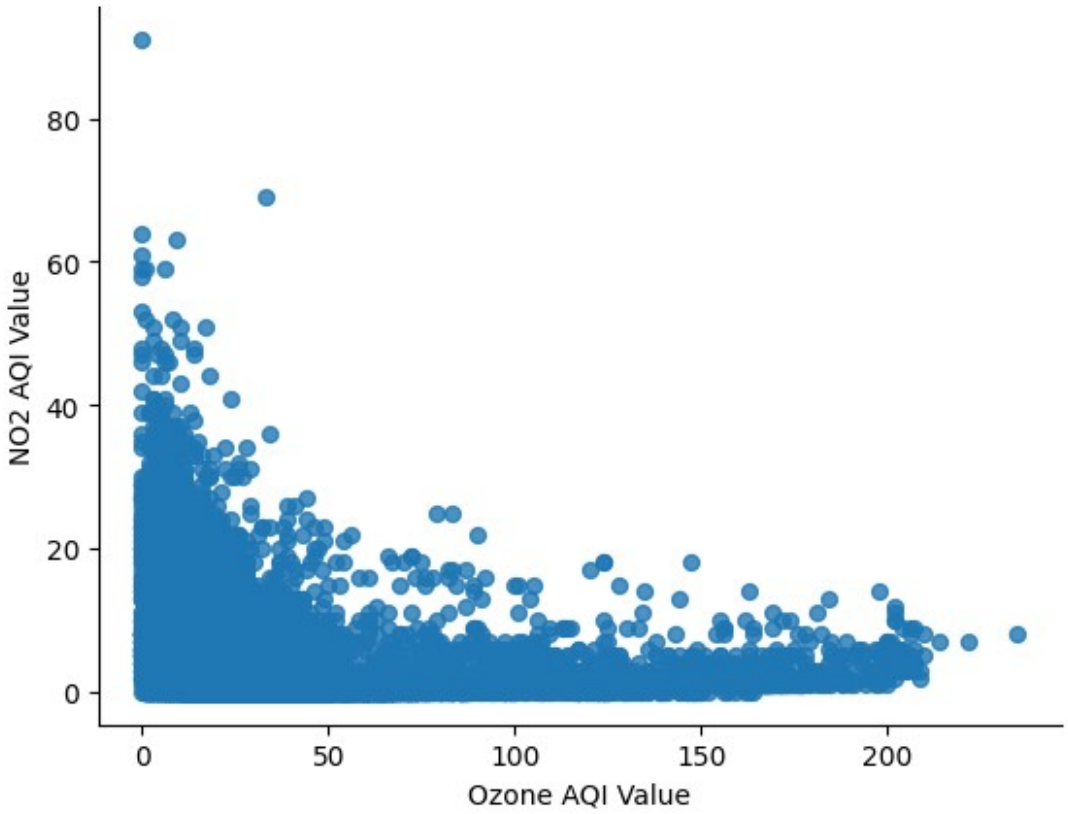
```
{"summary":"{\n  \"name\": \"pollution_data\",\n  \"rows\": 23463,\n
\"fields\": [\n    {\n      \"column\": \"Country\",\n
\"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 175,\n        \"samples\": [\n
```

```
\"Comoros\",\n          \"Turkmenistan\",\n          \"Honduras\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"City\",\n      \"properties\":
{\n        \"dtype\": \"string\",\n        \"num_unique_values\":
23462,\n        \"samples\": [\n          \"Fonte Boa\",\n
\"Vidin\",\n          \"Kunda\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"AQI Value\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
56,\n        \"min\": 6,\n        \"max\": 500,\n
\"num_unique_values\": 347,\n        \"samples\": [\n          250,\n
222,\n          348\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"AQI
Category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 6,\n        \"samples\":
[\n          \"Moderate\",\n          \"Good\",\n
\"Hazardous\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"CO
AQI Value\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 1,\n        \"min\": 0,\n        \"max\": 133,\n
\"num_unique_values\": 34,\n        \"samples\": [\n          27,\n
13,\n          67\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"CO
AQI Category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 3,\n        \"samples\":
[\n          \"Good\",\n          \"Unhealthy for Sensitive Groups\",\
n          \"Moderate\"\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Ozone AQI Value\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 28,\n        \"min\": 0,\n
\"max\": 235,\n        \"num_unique_values\": 213,\n
\"samples\": [\n          89,\n          189,\n          138\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Ozone AQI Category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 5,\n        \"samples\":
[\n          \"Moderate\",\n          \"Very Unhealthy\",\n
\"Unhealthy for Sensitive Groups\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"NO2 AQI Value\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5,\n        \"min\": 0,\n        \"max\": 91,\n
\"num_unique_values\": 59,\n        \"samples\": [\n          0,\n
7,\n          44\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"NO2
AQI Category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 2,\n        \"samples\":
[\n          \"Moderate\",\n          \"Good\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

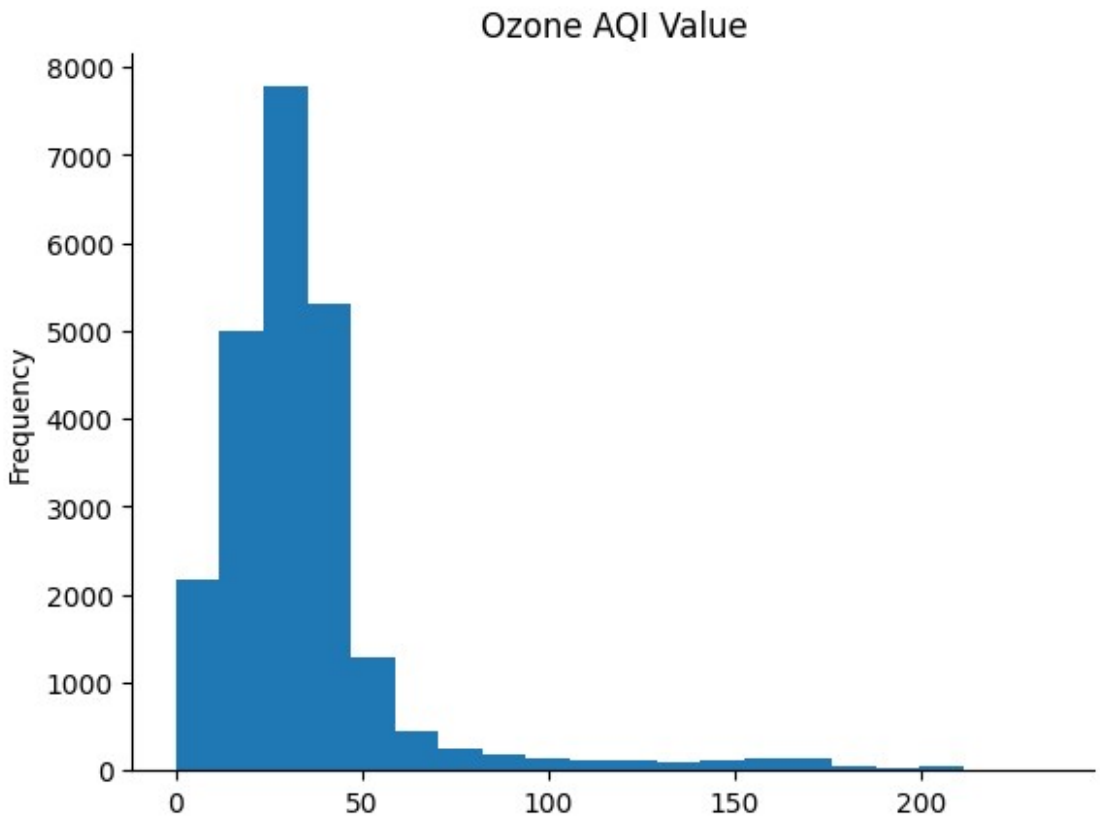
```
n    },\n    {\n      \"column\": \"PM2.5 AQI Value\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
54,\n        \"min\": 0,\n        \"max\": 500,\n
\"num_unique_values\": 383,\n        \"samples\": [\n          136,\n
239\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PM2.5 AQI Category\",\n      \"properties\": {\n        \"dtype\":
\"category\",\n        \"num_unique_values\": 6,\n        \"samples\":
[\n          \"Moderate\",\n          \"Good\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"pollution_data"}

# @title AQI Category

from matplotlib import pyplot as plt
import seaborn as sns
pollution_data.groupby('AQI Category').size().plot(kind='barh',
color=sns.palettes.mpl_palette('Dark2'))
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
# @title Ozone AQI Value vs NO2 AQI Value

from matplotlib import pyplot as plt
pollution_data.plot(kind='scatter', x='Ozone AQI Value', y='NO2 AQI
Value', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
# @title Ozone AQI Value

from matplotlib import pyplot as plt
pollution_data['Ozone AQI Value'].plot(kind='hist', bins=20,
title='Ozone AQI Value')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



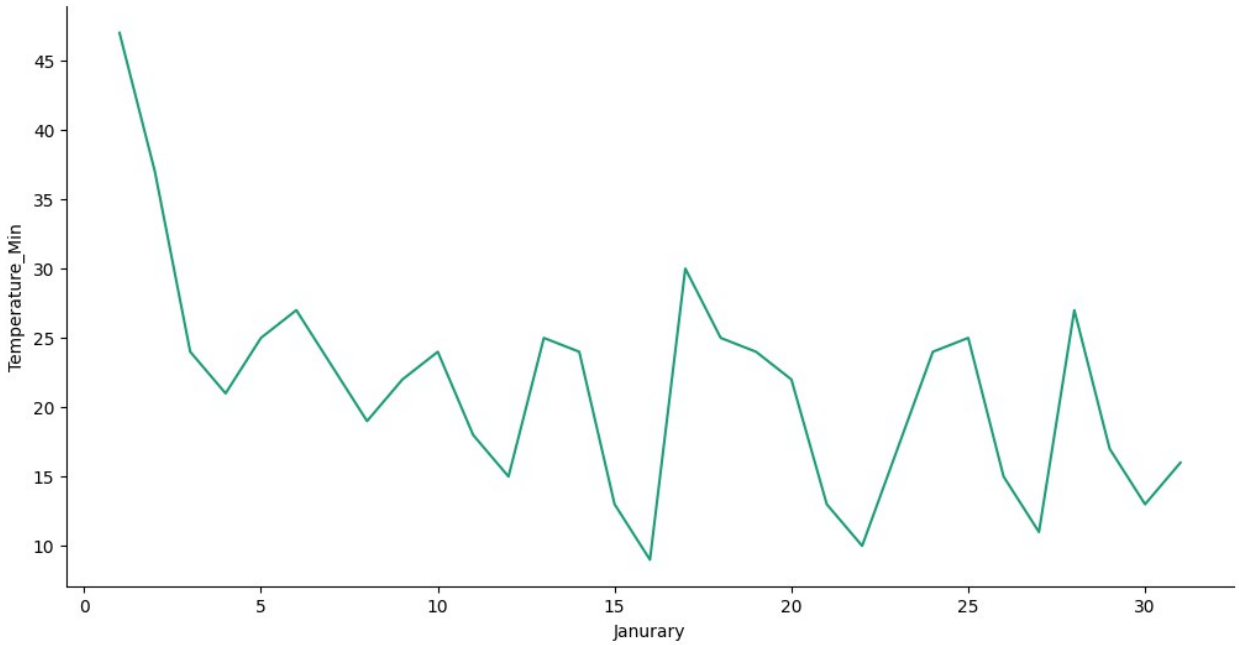
weather_data


```
{"summary":"{\n  \"name\": \"weather_data\",\n  \"rows\": 31,\n
\"fields\": [\n    {\n      \"column\": \"Janurary\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
9,\n        \"min\": 1,\n        \"max\": 31,\n
\"num_unique_values\": 31,\n        \"samples\": [\n          28,\n
16,\n          24\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Temperature_Max(\\u00b0F)\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 8,\n        \"min\": 20,\n
\"max\": 60,\n        \"num_unique_values\": 17,\n        \"samples\":
[\n          54,\n          60,\n          28\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Temperature_Avg\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
7.976829348281991,\n        \"min\": 16.9,\n        \"max\": 50.5,\n
\"num_unique_values\": 30,\n        \"samples\": [\n          21.3,\n
19.8,\n          32.5\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Temperature_Min\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 7,\n        \"min\": 9,\n
\"max\": 47,\n        \"num_unique_values\": 18,\n        \"samples\":
[\n          47,\n          37,\n          22\n        ],\n
```

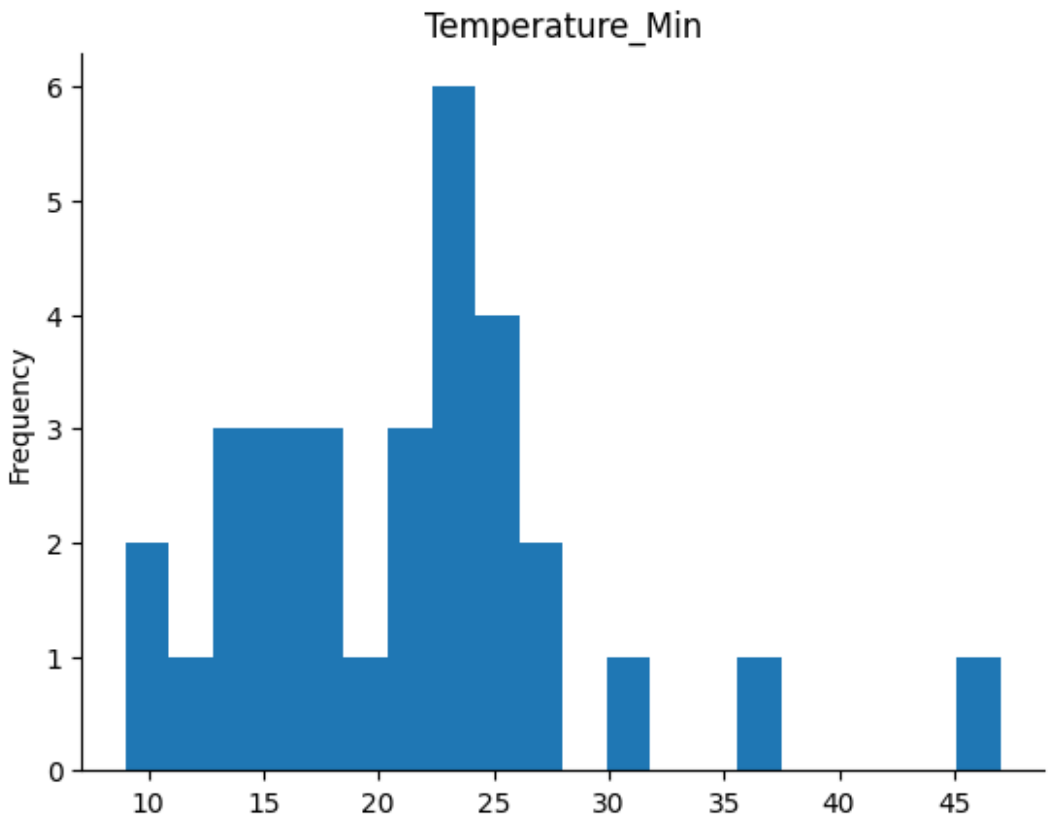
```
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"pm25\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 68,\n        \"min\": 44,\n
\"max\": 376,\n        \"num_unique_values\": 25,\n
\"samples\": [\n          159,\n          120,\n          152\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"pm10\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 66,\n
\"min\": 35,\n        \"max\": 352,\n        \"num_unique_values\":
28,\n        \"samples\": [\n          79,\n          78,\n
77\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"o3\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 8,\n        \"min\": 14,\n        \"max\": 46,\n
\"num_unique_values\": 21,\n        \"samples\": [\n          41,\n
44,\n          36\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"no2\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 3,\n        \"min\": 4,\n        \"max\": 18,\n
\"num_unique_values\": 15,\n        \"samples\": [\n          15,\n
6,\n          17\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"so2\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 3,\n        \"min\": 5,\n        \"max\": 22,\n
\"num_unique_values\": 12,\n        \"samples\": [\n          12,\n
14,\n          15\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"
co\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 4,\n        \"min\": 4,\n        \"max\": 23,\n
\"num_unique_values\": 13,\n        \"samples\": [\n          11,\n
12,\n          13\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Humidity_Max(%)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 10,\n        \"min\": 54,\n
\"max\": 94,\n        \"num_unique_values\": 20,\n        \"samples\":
[\n          94,\n          80,\n          59\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Humidity_AVg(%)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
11.233731845775727,\n        \"min\": 42.9,\n        \"max\": 89.4,\n
\"num_unique_values\": 30,\n        \"samples\": [\n          43.5,\n
78.4,\n          64.5\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Humidity_Min(%)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 11,\n        \"min\": 24,\n
\"max\": 86,\n        \"num_unique_values\": 20,\n        \"samples\":
[\n          86,\n          40,\n          27\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"Wind_Speed_Max(mph)\",\n
```

```
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5,\n        \"min\": 6,\n        \"max\": 26,\n
\"num_unique_values\": 13,\n        \"samples\": [\n          24,\n
17,\n          9\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Wind_Speed_Avgmph)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 4.519084977322075,\n        \"min\":
0.8,\n        \"max\": 17.0,\n        \"num_unique_values\": 30,\n
\"samples\": [\n          16.2,\n          14.7,\n          6.4\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Wind_Speed_Min(mph)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
2,\n        \"min\": 0,\n        \"max\": 8,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          5,\n
6,\n          8\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Precipitation (in)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.24515871532320596,\n        \"min\":
0.0,\n        \"max\": 1.1,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          0.77,\n          1.1,\n          0.01\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    }\n  ]\n}","type":"dataframe","variable_name":"weather_data"}
```

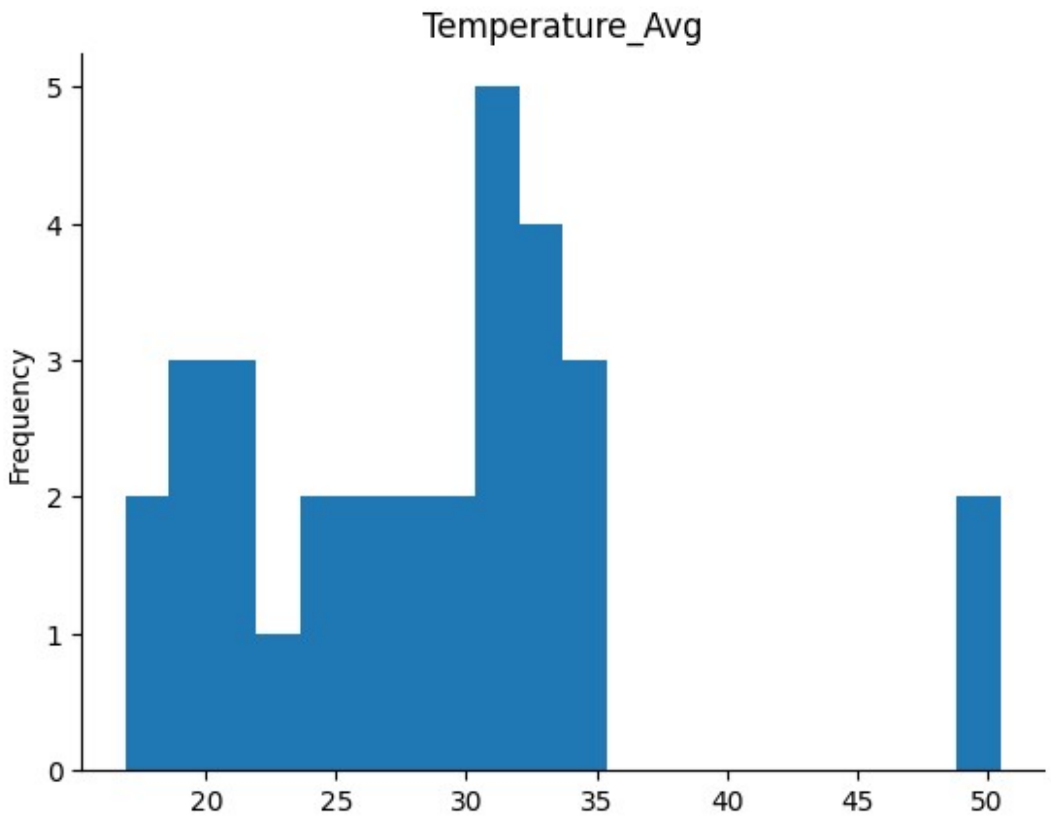

```
# @title Janurary vs Temperature_Min

from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['Janurary']
  ys = series['Temperature_Min']

  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = weather_data.sort_values('Janurary', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Janurary')
_ = plt.ylabel('Temperature_Min')
```



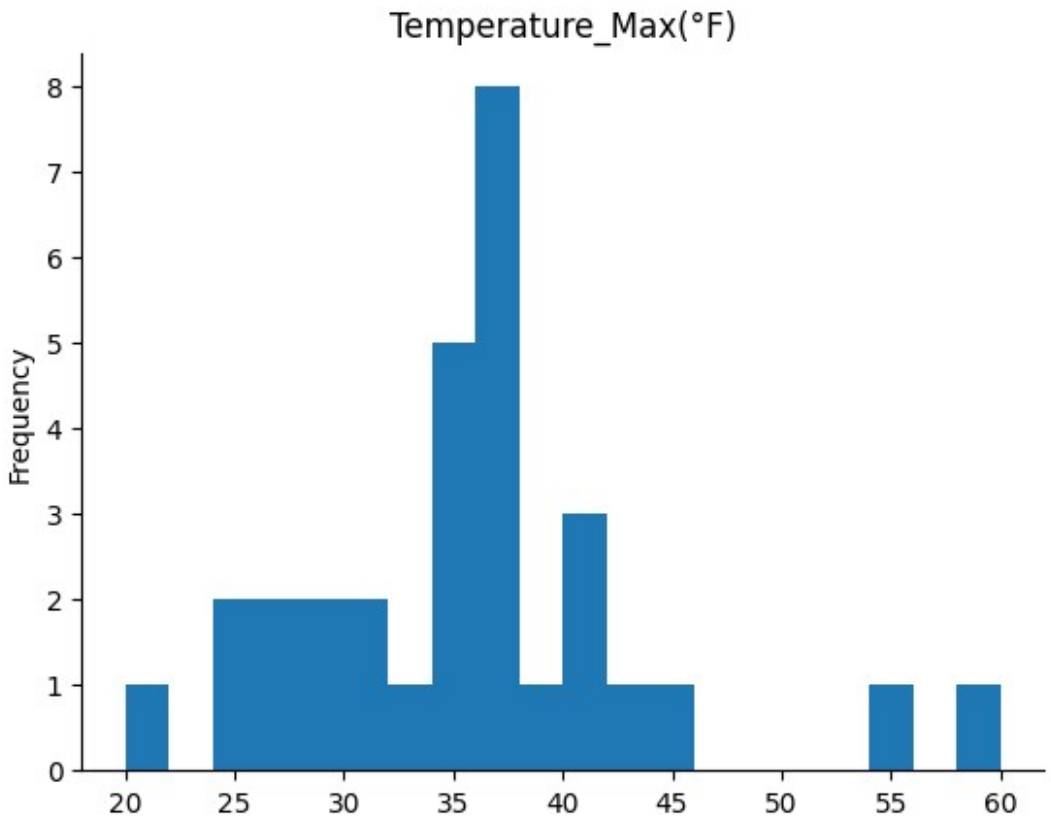

```
# @title Temperature_Min

from matplotlib import pyplot as plt
weather_data['Temperature_Min'].plot(kind='hist', bins=20,
title='Temperature_Min')
plt.gca().spines[['top', 'right',]].set_visible(False)
```



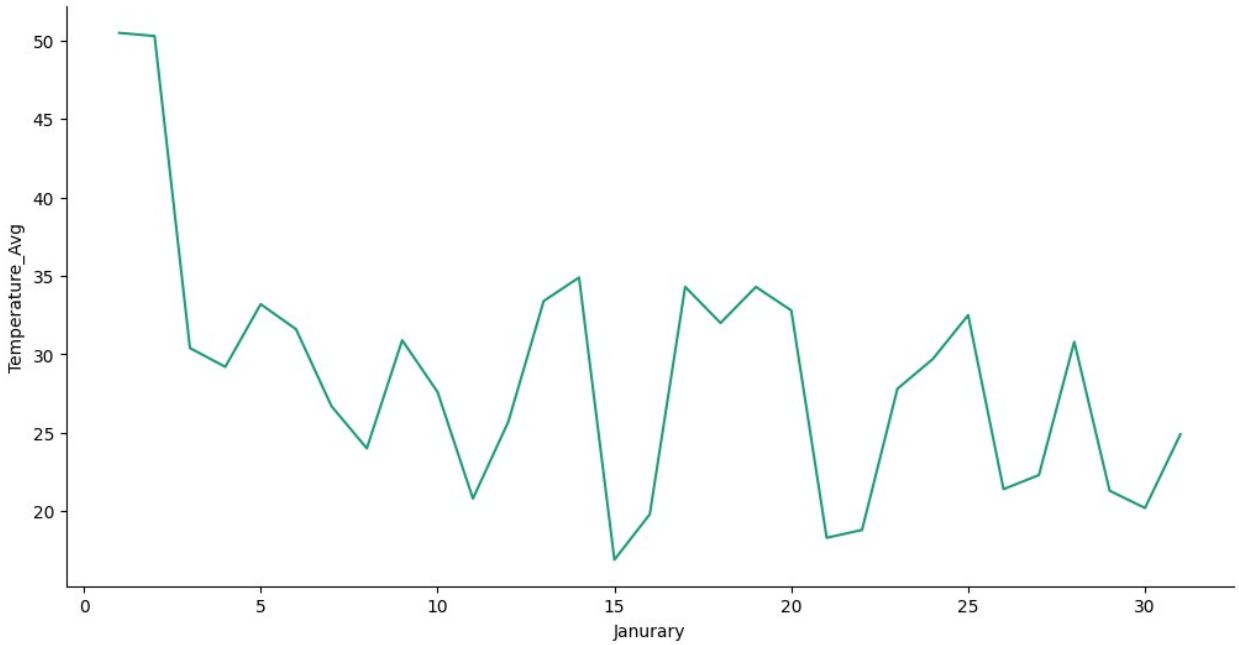

```
# @title Temperature_Avg

from matplotlib import pyplot as plt
weather_data['Temperature_Avg'].plot(kind='hist', bins=20,
title='Temperature_Avg')
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
# @title Temperature_Max(°F)

from matplotlib import pyplot as plt
weather_data['Temperature_Max(°F)'].plot(kind='hist', bins=20,
title='Temperature_Max(°F)')
plt.gca().spines[['top', 'right',]].set_visible(False)
```




```
# @title Janurary vs Temperature_Avg

from matplotlib import pyplot as plt
import seaborn as sns
def _plot_series(series, series_name, series_index=0):
  palette = list(sns.palettes.mpl_palette('Dark2'))
  xs = series['Janurary']
  ys = series['Temperature_Avg']

  plt.plot(xs, ys, label=series_name, color=palette[series_index %
len(palette)])

fig, ax = plt.subplots(figsize=(10, 5.2), layout='constrained')
df_sorted = weather_data.sort_values('Janurary', ascending=True)
_plot_series(df_sorted, '')
sns.despine(fig=fig, ax=ax)
plt.xlabel('Janurary')
_ = plt.ylabel('Temperature_Avg')
```




```
# prompt: Using dataframe weather_data:

# Select the columns of interest
weather_data_subset = weather_data[['Temperature_Max(\u00b0F)',
'Temperature_Avg', 'Temperature_Min']]

# Calculate the mean of each column
weather_data_means = weather_data_subset.mean(axis=0)

# Print the results
print(weather_data_means)
```

```
Temperature_Max(°F)    35.161290
Temperature_Avg        28.622581
Temperature_Min        21.354839
dtype: float64
```