Data Preparation

1. Data Preprocessing

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

# Load the provided datasets
file_paths = {
    'CO': 'Data/CO.csv',
    'NO2': 'Data/NO2.csv',
    'NOX': 'Data/NOX.csv',
    'O3': 'Data/O3.csv',
    'PM25': 'Data/PM25.csv',
    'PM10': 'Data/PM10.csv',
    'SO2': 'Data/SO2.csv'
}
```

1.1 Data Cleaning

In [2]:

```
def clean_and_reformat(data_path):
    data = pd.read_csv(data_path)
    data = data.replace('N.A.', pd.NA).apply(pd.to_numeric, errors='ignore')
    data['DATE'] = pd.to_datetime(data['DATE'], dayfirst=False, errors='coerce')
    data['DATETIME'] = data['DATE'] + pd.to_timedelta(data['HOUR'] - 1, unit='h')
    data = data.drop(['DATE', 'HOUR', 'POLLUTANT'], axis=1).set_index('DATETIME')
    return data

# clean and reformat each dataset
datasets = {pollutant: clean_and_reformat(path) for pollutant, path in file_paths.i

# combine all datasets
all_data = pd.concat(datasets.values(), axis=1, keys=datasets.keys())

all_data.head()
```

Out[2]:

	SHATIN	TSUEN WAN	CENTRAL	EASTERN	KWUN TONG	TUEN MUN	TUNG CHUNG	SHAM SHUI PO	SOUTHI
DATETIME									
1990-01-01 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1990-01-01 01:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1990-01-01 02:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1990-01-01 03:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
1990-01-01 04:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

5 rows × 133 columns

In [3]:

```
# generate statistical summary for each pollutant
statistical_summary = all_data.describe()

# show 'CO' as an example
statistical_summary['CO']
```

Out[3]:

	SHATIN	TSUEN WAN	CENTRAL	EASTERN	KWUN TONG	TUEN MUN	TU CHU
count	0.0	220821.000000	212909.000000	0.0	0.0	84554.000000	208964.000
mean	NaN	71.981342	91.662325	NaN	NaN	67.557478	63.483
std	NaN	33.606701	45.396913	NaN	NaN	21.801282	35.420
min	NaN	0.000000	0.000000	NaN	NaN	9.000000	0.000
25%	NaN	49.000000	59.000000	NaN	NaN	53.000000	40.000
50%	NaN	69.000000	85.000000	NaN	NaN	65.000000	57.000
75%	NaN	91.000000	115.000000	NaN	NaN	80.000000	80.000
max	NaN	529.000000	518.000000	NaN	NaN	261.000000	573.000

In [4]:

```
# Check for missing data
missing_data_summary = all_data.isnull().mean().unstack(level=0).mul(100).round(2)
# display the percentage of missing data for each pollutant in each location
missing_data_summary
```

Out[4]:

	CO	NO2	NOX	O3	PM25	PM10	SO2
SHATIN	100.00	16.26	16.26	25.19	66.55	14.45	7.89
TSUEN WAN	25.72	7.47	7.46	18.67	30.51	15.02	6.68
CENTRAL	28.38	28.17	28.17	62.52	30.12	29.74	28.18
EASTERN	100.00	30.69	100.00	30.69	63.87	28.74	30.69
KWUN TONG	100.00	6.70	6.69	27.11	63.68	16.38	5.95
TUEN MUN	71.56	71.78	71.78	71.72	72.08	72.08	71.68
TUNG CHUNG	29.71	29.96	29.97	29.87	29.48	29.17	30.40
SHAM SHUI PO	100.00	6.07	6.07	26.59	64.16	25.91	4.90
SOUTHERN	90.11	90.15	90.15	90.10	90.28	90.27	90.15
YUEN LONG	43.07	20.54	33.83	20.38	45.80	19.94	20.44
CENTRAL/WESTERN	100.00	6.55	6.53	6.11	64.60	14.70	5.32
NORTH	90.10	90.14	90.14	90.09	90.48	90.47	90.12
KWAI CHUNG	96.29	4.52	4.52	4.09	63.05	13.11	3.59
TAP MUN	29.49	29.51	29.26	29.81	30.64	27.66	28.92
TSEUNG KWAN O	77.79	77.80	77.80	77.78	77.79	77.76	77.86
TAI PO	100.00	7.61	50.00	19.61	65.36	27.38	18.55
MONG KOK	8.72	9.16	9.16	63.13	63.22	22.77	8.43
CAUSEWAY BAY	27.30	26.53	26.53	63.05	64.55	26.62	26.40
Average	5.31	0.87	0.85	0.72	26.54	8.88	0.98

CO

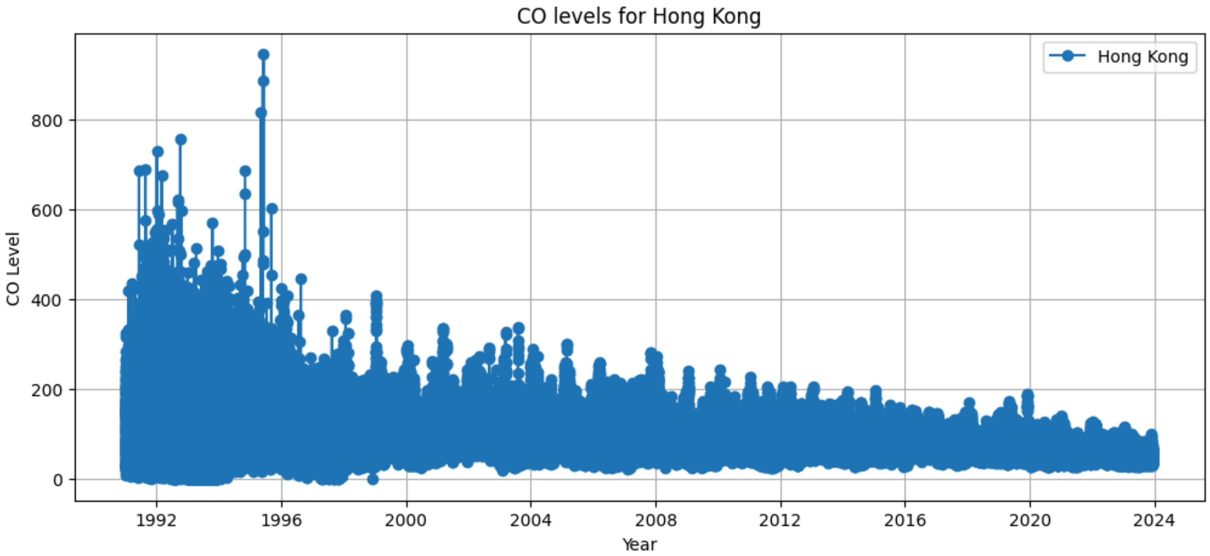
In [5]:

```
co_data = all_data['CO']
# plot the 'CO' data for each location
# def plot_co_continuous(data, location):
#     plt.figure(figsize=(12, 5))
#     plt.plot(data.index, data[location], marker='o', linestyle='-', label=locatio
#     plt.title(f'CO levels for {location}')
#     plt.xlabel('Year')
#     plt.ylabel('CO Level')
#     plt.legend()
#     plt.grid(True)
#     plt.show()
```

```
#
# for location in co_data.columns:
#     plot_co_continuous(co_data, location)

# plot the 'CO' data for the average level of all locations
def plot_co(data):
    plt.figure(figsize=(12, 5))
    plt.plot(data.index, data['Average'], marker='o', linestyle='-', label='Hong Ko
    plt.title('CO levels for Hong Kong')
    plt.xlabel('Year')
    plt.ylabel('CO Level')
    plt.legend()
    plt.grid(True)
    plt.show()


plot_co(co_data)
```



NO_2

In [6]:

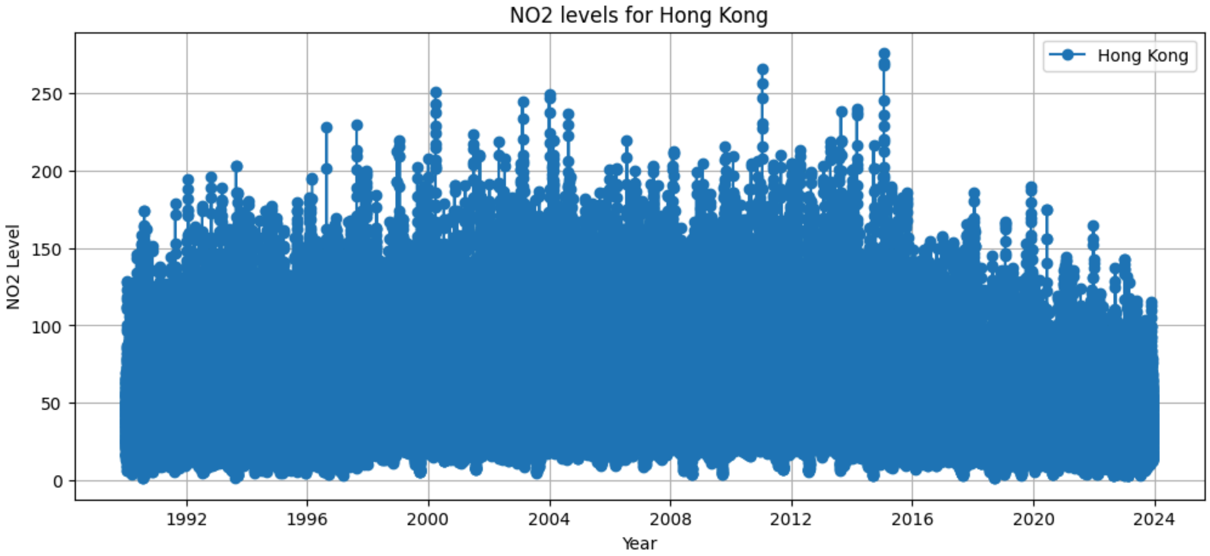
```
no2_data = all_data['NO2']
# plot the 'NO2' data
# def plot_no2_continuous(data, location):
#     plt.figure(figsize=(12, 5))  # Set a larger figure size for better visibility
#     plt.plot(data.index, data[location], marker='o', linestyle='-', label=locatio
#     plt.title(f'NO2 levels for {location}')
#     plt.xlabel('Year')
#     plt.ylabel('NO2 Level')
#     plt.legend()
#     plt.grid(True)
#     plt.show()

#
# for location in no2_data.columns:
#     plot_no2_continuous(no2_data, location)

# plot the 'NO2' data for the average level of all locations
```

```
def plot_no2(data):
    plt.figure(figsize=(12, 5))
    plt.plot(data.index, data['Average'], marker='o', linestyle='-', label='Hong Ko
    plt.title('NO2 levels for Hong Kong')
    plt.xlabel('Year')
    plt.ylabel('NO2 Level')
    plt.legend()
    plt.grid(True)
    plt.show()


plot_no2(no2_data)
```


NO2 levels for Hong Kong

NOX

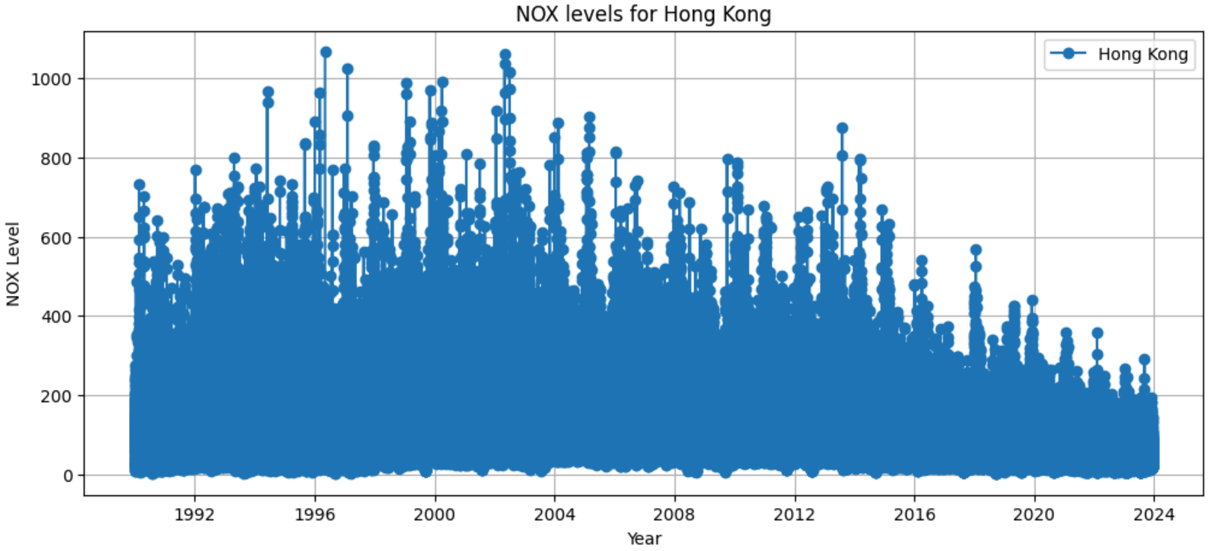
In [7]:

```
nox_data = all_data['NOX']
# # plot the 'NOX' data
# def plot_nox_continuous(data, location):
#     plt.figure(figsize=(12, 5))
#     plt.plot(data.index, data[location], marker='o', linestyle='-', label=locatio
#     plt.title(f'NOX levels for {location}')
#     plt.xlabel('Year')
#     plt.ylabel('NOX Level')
#     plt.legend()
#     plt.grid(True)
#     plt.show()

#
# for location in nox_data.columns:
#     plot_nox_continuous(nox_data, location)

def plot_nox(data):
    plt.figure(figsize=(12, 5))
    plt.plot(data.index, data['Average'], marker='o', linestyle='-', label='Hong Ko
    plt.title('NOX levels for Hong Kong')
    plt.xlabel('Year')
    plt.ylabel('NOX Level')
    plt.legend()
    plt.grid(True)
    plt.show()


plot_nox(nox_data)
```



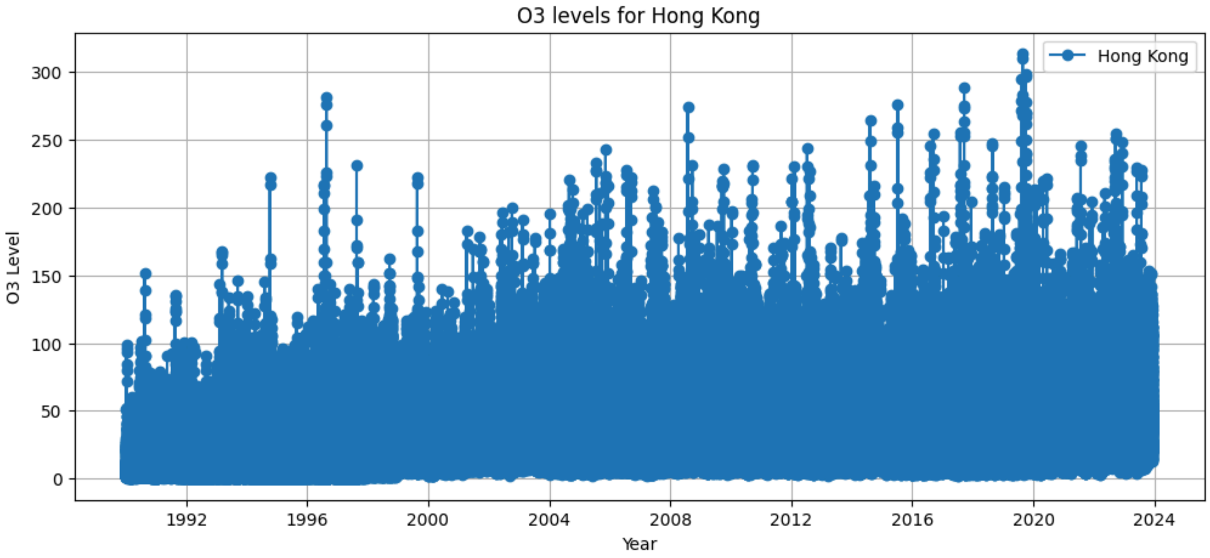
O_3

```
In [8]: o3_data = all_data['O3']
# plot the 'O3' data
# def plot_o3_continuous(data, location):
#     plt.figure(figsize=(12, 5))
#     plt.plot(data.index, data[location], marker='o', linestyle='-', label=locatio
#     plt.title(f'O3 levels for {location}')
#     plt.xlabel('Year')
#     plt.ylabel('O3 Level')
#     plt.legend()
#     plt.grid(True)
#     plt.show()

#
# for location in o3_data.columns:
#     plot_o3_continuous(o3_data, location)

def plot_o3(data):
    plt.figure(figsize=(12, 5))
    plt.plot(data.index, data['Average'], marker='o', linestyle='-', label='Hong Ko
    plt.title('O3 levels for Hong Kong')
    plt.xlabel('Year')
    plt.ylabel('O3 Level')
    plt.legend()
    plt.grid(True)
    plt.show()


plot_o3(o3_data)
```



$PM25$

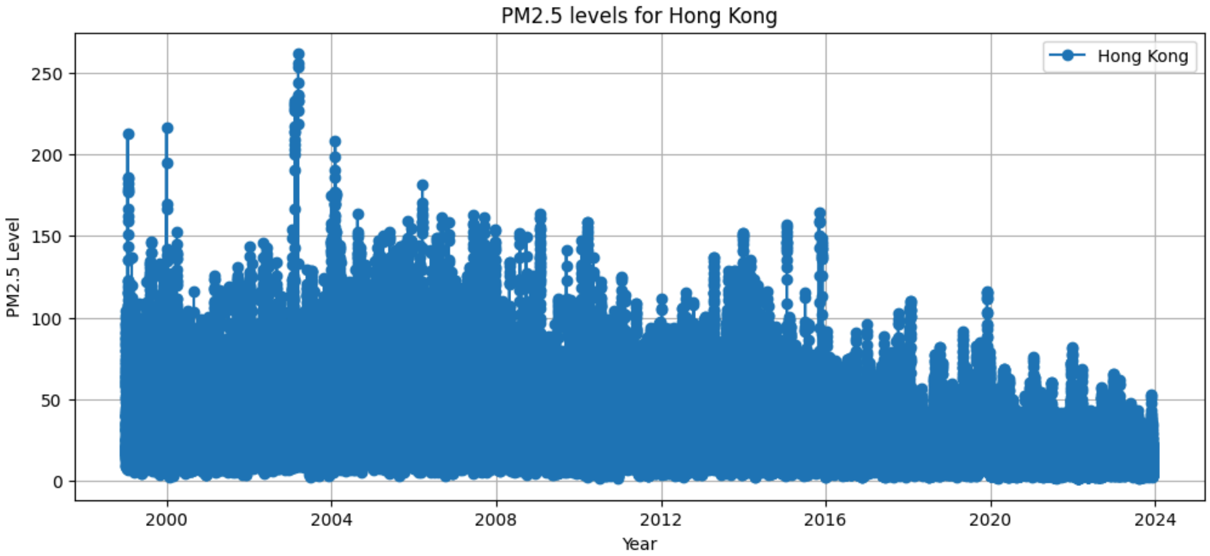
In [9]:

```
pm25_data = all_data['PM25']
# plot the 'PM2.5' data
# def plot_pm25_continuous(data, location):
#     plt.figure(figsize=(12, 5))
#     plt.plot(data.index, data[location], marker='o', linestyle='-', label=locatio
#     plt.title(f'PM2.5 levels for {location}')
#     plt.xlabel('Year')
#     plt.ylabel('PM2.5 Level')
#     plt.legend()
#     plt.grid(True)
#     plt.show()

#
# for location in pm25_data.columns:
#     plot_pm25_continuous(pm25_data, location)

def plot_pm25(data):
    plt.figure(figsize=(12, 5))
    plt.plot(data.index, data['Average'], marker='o', linestyle='-', label='Hong Ko
    plt.title('PM2.5 levels for Hong Kong')
    plt.xlabel('Year')
    plt.ylabel('PM2.5 Level')
    plt.legend()
    plt.grid(True)
    plt.show()


plot_pm25(pm25_data)
```



$PM10$

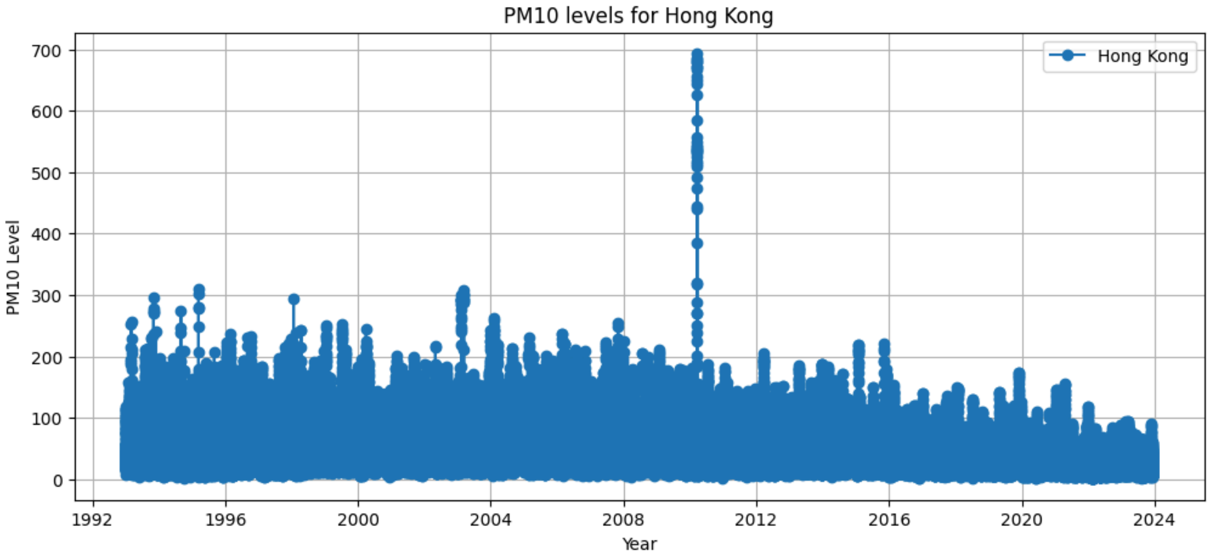
In [10]:

```python
pm10_data = all_data['PM10']
# plot the 'PM10' data for each location
# def plot_pm10_continuous(data, location):
#     plt.figure(figsize=(12, 5))
#     plt.plot(data.index, data[location], marker='o', linestyle='-', label=locatio
#     plt.title(f'PM10 levels for {location}')
#     plt.xlabel('Year')
#     plt.ylabel('PM10 Level')
#     plt.legend()
#     plt.grid(True)
#     plt.show()

#
# for location in pm10_data.columns:
#     plot_pm10_continuous(pm10_data, location)

def plot_pm10(data):
    plt.figure(figsize=(12, 5))
    plt.plot(data.index, data['Average'], marker='o', linestyle='-', label='Hong Ko
    plt.title('PM10 levels for Hong Kong')
    plt.xlabel('Year')
    plt.ylabel('PM10 Level')
    plt.legend()
    plt.grid(True)
    plt.show()


plot_pm10(pm10_data)
```



SO_2

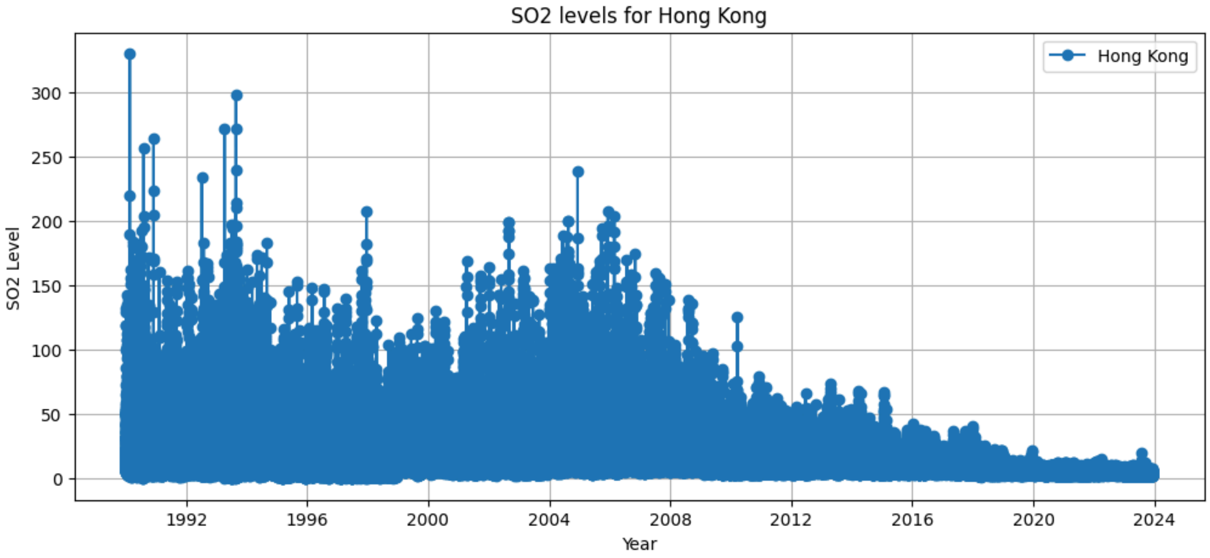
In [11]:

```
# Continuing from your existing code
so2_data = all_data['SO2']
# Now let's plot the 'SO2' data for each location with continuous values
# def plot_so2_continuous(data, location):
#     plt.figure(figsize=(12, 5))  # Set a larger figure size for better visibility
#     plt.plot(data.index, data[location], marker='o', linestyle='-', label=locatio
#     plt.title(f'SO2 levels for {location}')
#     plt.xlabel('Year')
#     plt.ylabel('SO2 Level')
#     plt.legend()
#     plt.grid(True)
#     plt.show()

#
# for location in so2_data.columns:
#     plot_so2_continuous(so2_data, location)

def plot_so2(data):
    plt.figure(figsize=(12, 5))  # Set a larger figure size for better visibility
    plt.plot(data.index, data['Average'], marker='o', linestyle='-', label='Hong Ko
    plt.title('SO2 levels for Hong Kong')
    plt.xlabel('Year')
    plt.ylabel('SO2 Level')
    plt.legend()
    plt.grid(True)
    plt.show()


plot_so2(so2_data)
```



2. Exploratory Data Analysis