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

file_path = "/content/cleaned_data.csv"
df = pd.read_csv(file_path)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9357 entries, 0 to 9356
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Date           9357 non-null   object
 1   Time           9357 non-null   object
 2   CO(GT)         9357 non-null   float64
 3   PT08.S1(CO)    9357 non-null   float64
 4   C6H6(GT)       9357 non-null   float64
 5   PT08.S2(NMHC)  9357 non-null   float64
 6   NOx(GT)        9357 non-null   float64
 7   PT08.S3(NOx)   9357 non-null   float64
 8   NO2(GT)        9357 non-null   float64
 9   PT08.S4(NO2)   9357 non-null   float64
 10  PT08.S5(O3)    9357 non-null   float64
 11  T              9357 non-null   float64
 12  RH             9357 non-null   float64
 13  AH             9357 non-null   float64
 14  DateandTime    9357 non-null   object
dtypes: float64(12), object(3)
memory usage: 1.1+ MB
```

```
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 9357,\n  \"fields\":
[\n    {\n      \"column\": \"Date\",\n      \"properties\": {\n
\"dtype\": \"object\",\n        \"num_unique_values\": 391,\n
\"samples\": [\n          \"2004-03-19\",\n          \"2004-04-21\",\n
\"2004-04-12\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Time\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 24,\n        \"samples\": [\n
\"02:00:00\",\n          \"10:00:00\",\n          \"18:00:00\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"CO(GT)\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
1.3160683129140403,\n        \"min\": 0.1,\n        \"max\": 11.9,\n
\"num_unique_values\": 97,\n        \"samples\": [\n          0.4,\n
5.9,\n          9.3\n        ],\n        \"semantic_type\": \"\",\n
```

```
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PT08.S1(CO)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 218.2015608286704,\n        \"min\":
647.25,\n        \"max\": 2039.75,\n        \"num_unique_values\":
3541,\n        \"samples\": [\n          865.0,\n          1006.25,\n
1022.25\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"C6H6(GT)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 7.503295395520342,\n        \"min\":
0.1490477388337664,\n        \"max\": 63.74147644829163,\n
\"num_unique_values\": 4138,\n        \"samples\": [\n
4.882728460742918,\n          3.461428597119324,\n
23.53492475843005\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PT08.S2(NMHC)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 267.86485421169107,\n        \"min\":
383.25,\n        \"max\": 2214.0,\n        \"num_unique_values\":
4134,\n        \"samples\": [\n          1187.0,\n          754.5,\n
1029.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"NOx(GT)\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 193.41941704776588,\n        \"min\": 2.0,\n
\"max\": 1479.0,\n        \"num_unique_values\": 2467,\n
\"samples\": [\n          757.6,\n          209.4,\n          253.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"PT08.S3(NOx)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
255.708806840193,\n        \"min\": 322.0,\n        \"max\": 2682.75,\
n        \"num_unique_values\": 3868,\n        \"samples\": [\n
443.0,\n          1137.25,\n          506.3026315789474\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"NO2(GT)\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
43.91109515811381,\n        \"min\": 2.0,\n        \"max\": 339.7,\n
\"num_unique_values\": 1420,\n        \"samples\": [\n
125.0,\n          168.8,\n          171.6\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"PT08.S4(NO2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
343.2013608680068,\n        \"min\": 551.0,\n        \"max\": 2775.0,\
n        \"num_unique_values\": 4762,\n        \"samples\": [\n
1462.0,\n          1833.25,\n          1960.5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"PT08.S5(O3)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
404.4426107819769,\n        \"min\": 221.0,\n        \"max\":
2522.75,\n        \"num_unique_values\": 5041,\n        \"samples\":
[\n          1469.5,\n          841.75,\n          405.5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

```
n    },\n    {\n      \"column\": \"T\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 8.782367715415278,\n
\"min\": -1.8999999761581,\n        \"max\": 44.60000038147,\n
\"num_unique_values\": 3733,\n        \"samples\": [\n
32.39999961853,\n          7.0750000476837,\n          33.02499961853\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"RH\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 17.194080942558262,\n        \"min\": 9.1750001907349,\n
\"max\": 88.72500038147,\n        \"num_unique_values\": 5265,\n
\"samples\": [\n          19.200000286102,\n
76.075000762939,\n          66.250000953674\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"AH\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.4022030923673327,\n
\"min\": 0.1846790209991702,\n        \"max\": 2.2310357155831864,\n
\"num_unique_values\": 9353,\n        \"samples\": [\n
0.4866926276720519,\n          0.7236289583157048,\n
1.2282241423597364\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"DateandTime\",\n      \"properties\": {\n        \"dtype\":
\"object\",\n        \"num_unique_values\": 9357,\n
\"samples\": [\n          \"2004-07-12 15:00:00\",\n          \"2004-
07-30 06:00:00\",\n          \"2004-09-08 22:00:00\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"df"}
```


```
df['DateandTime'] = pd.to_datetime(df['DateandTime'])
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9357 entries, 0 to 9356
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Date           9357 non-null   object
 1   Time           9357 non-null   object
 2   CO(GT)         9357 non-null   float64
 3   PT08.S1(CO)    9357 non-null   float64
 4   C6H6(GT)       9357 non-null   float64
 5   PT08.S2(NMHC)  9357 non-null   float64
 6   NOx(GT)        9357 non-null   float64
 7   PT08.S3(NOx)   9357 non-null   float64
 8   NO2(GT)        9357 non-null   float64
 9   PT08.S4(NO2)   9357 non-null   float64
 10  PT08.S5(O3)    9357 non-null   float64
 11  T              9357 non-null   float64
 12  RH             9357 non-null   float64
 13  AH             9357 non-null   float64
```

```
 14  DateandTime    9357 non-null   datetime64[ns]
dtypes: datetime64[ns](1), float64(12), object(2)
memory usage: 1.1+ MB
```

```
df.set_index('DateandTime', inplace=True)
```

```
df.head()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 9357,\n  \"fields\":
[\n    {\n      \"column\": \"DateandTime\",\n      \"properties\": {\
n        \"dtype\": \"date\",\n        \"min\": \"2004-03-10
18:00:00\",\n        \"max\": \"2005-04-04 14:00:00\",\n
\"num_unique_values\": 9357,\n        \"samples\": [\n
\"2004-07-12 15:00:00\",\n          \"2004-07-30 06:00:00\",\n
\"2004-09-08 22:00:00\"\n        ],\n        \"semantic_type\": \"\",\
n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"Date\",\n      \"properties\": {\n        \"dtype\":
\"object\",\n        \"num_unique_values\": 391,\n        \"samples\":
[\n          \"2004-03-19\",\n          \"2004-04-21\",\n
\"2004-04-12\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Time\",\n      \"properties\": {\n        \"dtype\": \"object\",\n
\"num_unique_values\": 24,\n        \"samples\": [\n
\"02:00:00\",\n          \"10:00:00\",\n          \"18:00:00\"\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"CO(GT)\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
1.3160683129140403,\n        \"min\": 0.1,\n        \"max\": 11.9,\n
\"num_unique_values\": 97,\n        \"samples\": [\n          0.4,\n
5.9,\n          9.3\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PT08.S1(CO)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 218.2015608286704,\n        \"min\":
647.25,\n        \"max\": 2039.75,\n        \"num_unique_values\":
3541,\n        \"samples\": [\n          865.0,\n          1006.25,\n
1022.25\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"C6H6(GT)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 7.503295395520342,\n        \"min\":
0.1490477388337664,\n        \"max\": 63.74147644829163,\n
\"num_unique_values\": 4138,\n        \"samples\": [\n
4.882728460742918,\n          3.461428597119324,\n
23.53492475843005\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"PT08.S2(NMHC)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 267.86485421169107,\n        \"min\":
383.25,\n        \"max\": 2214.0,\n        \"num_unique_values\":
4134,\n        \"samples\": [\n          1187.0,\n          754.5,\n
1029.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
```

```
\"NOx(GT)\",\n      \"properties\": {\n        \"dtype\": \"number\",\
n        \"std\": 193.41941704776588,\n        \"min\": 2.0,\n
\"max\": 1479.0,\n        \"num_unique_values\": 2467,\n
\"samples\": [\n          757.6,\n          209.4,\n          253.3\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"PT08.S3(NOx)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
255.708806840193,\n        \"min\": 322.0,\n        \"max\": 2682.75,\
n        \"num_unique_values\": 3868,\n        \"samples\": [\n
443.0,\n          1137.25,\n          506.3026315789474\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"NO2(GT)\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\":
43.91109515811381,\n        \"min\": 2.0,\n        \"max\": 339.7,\n
\"num_unique_values\": 1420,\n        \"samples\": [\n
125.0,\n          168.8,\n          171.6\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"PT08.S4(NO2)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
343.2013608680068,\n        \"min\": 551.0,\n        \"max\": 2775.0,\
n        \"num_unique_values\": 4762,\n        \"samples\": [\n
1462.0,\n          1833.25,\n          1960.5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"PT08.S5(O3)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
404.4426107819769,\n        \"min\": 221.0,\n        \"max\":
2522.75,\n        \"num_unique_values\": 5041,\n        \"samples\":
[\n          1469.5,\n          841.75,\n          405.5\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"T\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 8.782367715415278,\n
\"min\": -1.8999999761581,\n        \"max\": 44.60000038147,\n
\"num_unique_values\": 3733,\n        \"samples\": [\n
32.39999961853,\n          7.0750000476837,\n          33.02499961853\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"RH\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 17.194080942558262,\n        \"min\": 9.1750001907349,\n
\"max\": 88.72500038147,\n        \"num_unique_values\": 5265,\n
\"samples\": [\n          19.200000286102,\n
76.075000762939,\n          66.250000953674\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"AH\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.4022030923673327,\n
\"min\": 0.1846790209991702,\n        \"max\": 2.2310357155831864,\n
\"num_unique_values\": 9353,\n        \"samples\": [\n
0.4866926276720519,\n          0.7236289583157048,\n
1.2282241423597364\n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n   }\n   }\n]\
n}","type":"dataframe","variable_name":"df"}

```
pip install ydata_profiling
```

```
Collecting ydata_profiling
  Downloading ydata_profiling-4.16.1-py2.py3-none-any.whl.metadata (22
kB)
Requirement already satisfied: scipy<1.16,>=1.4.1 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(1.14.1)
Requirement already satisfied: pandas!=1.4.0,<3.0,>1.1 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling) (2.2.2)
Requirement already satisfied: matplotlib<=3.10,>=3.5 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(3.10.0)
Requirement already satisfied: pydantic>=2 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(2.11.1)
Requirement already satisfied: PyYAML<6.1,>=5.0.0 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling) (6.0.2)
Requirement already satisfied: jinja2<3.2,>=2.11.1 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling) (3.1.6)
Collecting visions<0.8.2,>=0.7.5 (from
visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling)
  Downloading visions-0.8.1-py3-none-any.whl.metadata (11 kB)
Requirement already satisfied: numpy<2.2,>=1.16.0 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling) (2.0.2)
Collecting htmlmin==0.1.12 (from ydata_profiling)
  Downloading htmlmin-0.1.12.tar.gz (19 kB)
  Preparing metadata (setup.py) ...  ydata_profiling)
  Downloading phik-0.12.4-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (5.6 kB)
Requirement already satisfied: requests<3,>=2.24.0 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(2.32.3)
Requirement already satisfied: tqdm<5,>=4.48.2 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(4.67.1)
Requirement already satisfied: seaborn<0.14,>=0.10.1 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(0.13.2)
Collecting multimethod<2,>=1.4 (from ydata_profiling)
  Downloading multimethod-1.12-py3-none-any.whl.metadata (9.6 kB)
Requirement already satisfied: statsmodels<1,>=0.13.2 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(0.14.4)
Requirement already satisfied: typeguard<5,>=3 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling) (4.4.2)
Collecting imagehash==4.3.1 (from ydata_profiling)
```

```
  Downloading ImageHash-4.3.1-py2.py3-none-any.whl.metadata (8.0 kB)
Requirement already satisfied: wordcloud>=1.9.3 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling) (1.9.4)
Collecting dacite>=1.8 (from ydata_profiling)
  Downloading dacite-1.9.2-py3-none-any.whl.metadata (17 kB)
Requirement already satisfied: numba<=0.61,>=0.56.0 in
/usr/local/lib/python3.11/dist-packages (from ydata_profiling)
(0.60.0)
Collecting PyWavelets (from imagehash==4.3.1->ydata_profiling)
  Downloading pywavelets-1.8.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.0 kB)
Requirement already satisfied: pillow in
/usr/local/lib/python3.11/dist-packages (from imagehash==4.3.1-
>ydata_profiling) (11.1.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/dist-packages (from jinja2<3.2,>=2.11.1-
>ydata_profiling) (3.0.2)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5-
>ydata_profiling) (1.3.1)
Requirement already satisfied: cycler>=0.10 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5-
>ydata_profiling) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5-
>ydata_profiling) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5-
>ydata_profiling) (1.4.8)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5-
>ydata_profiling) (24.2)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5-
>ydata_profiling) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in
/usr/local/lib/python3.11/dist-packages (from matplotlib<=3.10,>=3.5-
>ydata_profiling) (2.8.2)
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
/usr/local/lib/python3.11/dist-packages (from numba<=0.61,>=0.56.0-
>ydata_profiling) (0.43.0)
Requirement already satisfied: pytz>=2020.1 in
/usr/local/lib/python3.11/dist-packages (from pandas!=1.4.0,<3.0,>1.1-
>ydata_profiling) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in
/usr/local/lib/python3.11/dist-packages (from pandas!=1.4.0,<3.0,>1.1-
>ydata_profiling) (2025.2)
Requirement already satisfied: joblib>=0.14.1 in
/usr/local/lib/python3.11/dist-packages (from phik<0.13,>=0.11.1-
```

```
>ydata_profiling) (1.4.2)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic>=2-
>ydata_profiling) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic>=2-
>ydata_profiling) (2.33.0)
Requirement already satisfied: typing-extensions>=4.12.2 in
/usr/local/lib/python3.11/dist-packages (from pydantic>=2-
>ydata_profiling) (4.13.0)
Requirement already satisfied: typing-inspection>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic>=2-
>ydata_profiling) (0.4.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0-
>ydata_profiling) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0-
>ydata_profiling) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0-
>ydata_profiling) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests<3,>=2.24.0-
>ydata_profiling) (2025.1.31)
Requirement already satisfied: patsy>=0.5.6 in
/usr/local/lib/python3.11/dist-packages (from statsmodels<1,>=0.13.2-
>ydata_profiling) (1.0.1)
Requirement already satisfied: attrs>=19.3.0 in
/usr/local/lib/python3.11/dist-packages (from visions<0.8.2,>=0.7.5-
>visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling) (25.3.0)
Requirement already satisfied: networkx>=2.4 in
/usr/local/lib/python3.11/dist-packages (from visions<0.8.2,>=0.7.5-
>visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling) (3.4.2)
Collecting puremagic (from visions<0.8.2,>=0.7.5-
>visions[type_image_path]<0.8.2,>=0.7.5->ydata_profiling)
  Downloading puremagic-1.28-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.7-
>matplotlib<=3.10,>=3.5->ydata_profiling) (1.17.0)
Downloading ydata_profiling-4.16.1-py2.py3-none-any.whl (400 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 400.1/400.1 kB 14.2 MB/s eta
0:00:00
ageHash-4.3.1-py2.py3-none-any.whl (296 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 296.5/296.5 kB 21.7 MB/s eta
0:00:00
ultimethod-1.12-py3-none-any.whl (10 kB)
Downloading phik-0.12.4-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (687 kB)
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 687.8/687.8 kB 35.8 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 105.4/105.4 kB 8.6 MB/s eta
0:00:00
agic-1.28-py3-none-any.whl (43 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 43.2/43.2 kB 3.6 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.5/4.5 MB 88.3 MB/s eta
0:00:00
lmin
  Building wheel for htmlmin (setup.py) ... lmin: filename=htmlmin-
0.1.12-py3-none-any.whl size=27081
sha256=27c5ab352ef0cdb2646b3cb32b7748eeba635832c8e9f5c62502e52d86b682a
6
  Stored in directory:
/root/.cache/pip/wheels/8d/55/1a/19cd535375ed1ede0c996405ebffe34b196d7
8e2d9545723a2
Successfully built htmlmin
Installing collected packages: puremagic, htmlmin, PyWavelets,
multimethod, dacite, imagehash, visions, phik, ydata_profiling
Successfully installed PyWavelets-1.8.0 dacite-1.9.2 htmlmin-0.1.12
imagehash-4.3.1 multimethod-1.12 phik-0.12.4 puremagic-1.28 visions-
0.8.1 ydata_profiling-4.16.1
```

```
from ydata_profiling import ProfileReport
profile = ProfileReport(df, title="Pandas Profiling Report")
profile.to_file("AirQuality_report.html")
```

```
<IPython.core.display.HTML object>
```


```
{"model_id":"4ab7db8891c44facb65ee7d29a24c669","version_major":2,"vers
ion_minor":0}
```

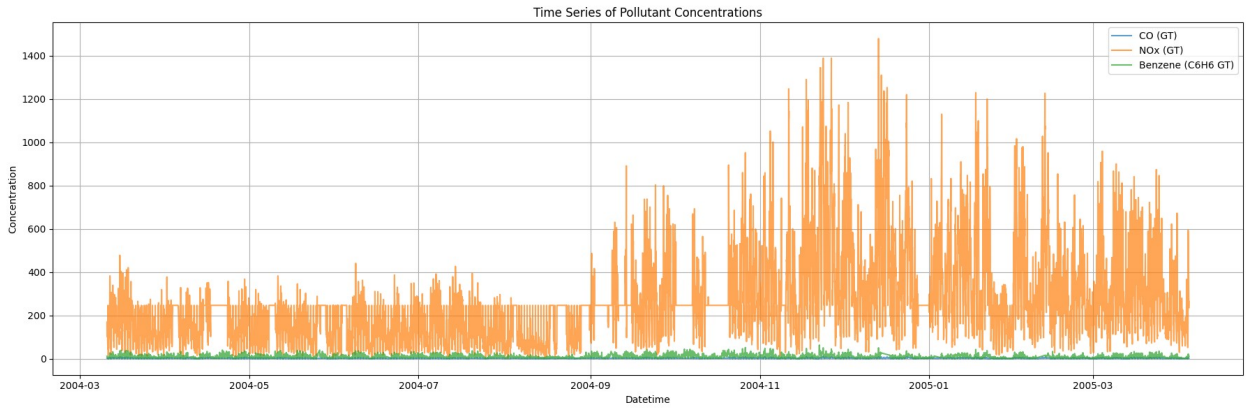

```
  0%|          | 0/14 [00:00<?, ?it/s]
```


```
{"model_id":"d142a2917cdc40718db82f8545adc51c","version_major":2,"vers
ion_minor":0}

{"model_id":"46f7a75b70c54fd79eef4322037e79d6","version_major":2,"vers
ion_minor":0}

{"model_id":"81dae606ad1b4811b4054076f414733f","version_major":2,"vers
ion_minor":0}
```


```
data.columns
```

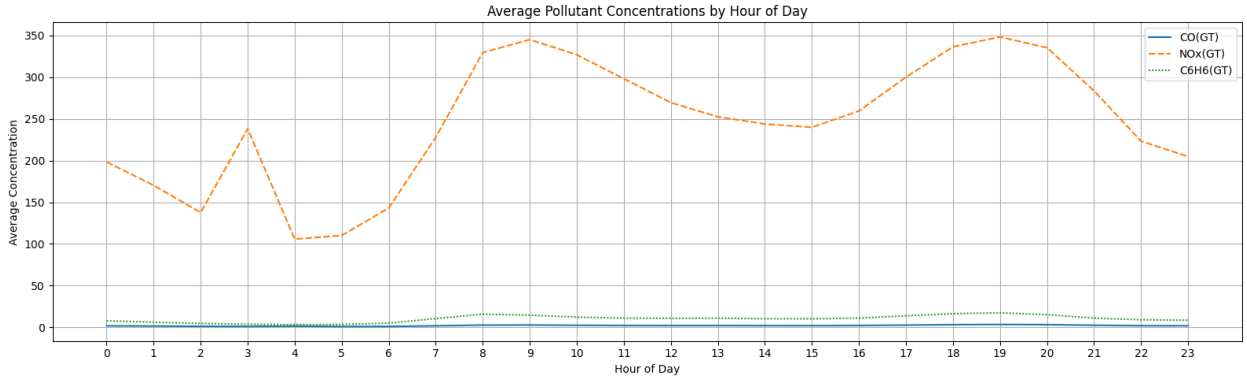
```
Index(['Date', 'Time', 'CO(GT)', 'PT08.S1(CO)', 'C6H6(GT)',
'PT08.S2(NMHC)',
       'NOx(GT)', 'PT08.S3(NOx)', 'NO2(GT)', 'PT08.S4(NO2)',
```

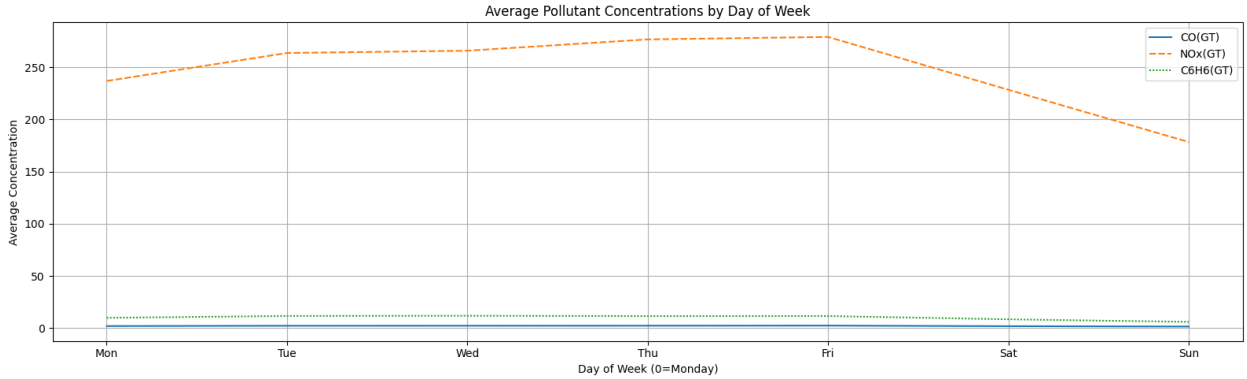
```
'PT08.S5(O3)',
       'T', 'RH', 'AH'],
      dtype='object')

#Time-series plots for CO, NOx, and Benzene
plt.figure(figsize=(18, 6))
plt.plot(df.index, df['CO(GT)'], label='CO (GT)', alpha=0.7)
plt.plot(df.index, df['NOx(GT)'], label='NOx (GT)', alpha=0.7)
plt.plot(df.index, df['C6H6(GT)'], label='Benzene (C6H6 GT)',
alpha=0.7)
plt.title('Time Series of Pollutant Concentrations')
plt.xlabel('Datetime')
plt.ylabel('Concentration')
plt.legend()
plt.tight_layout()
plt.grid(True)
plt.show()
```




```
#Adding Hours and DayOfWeek columns
df['hour'] = df.index.hour
df['day_of_week'] = df.index.dayofweek  # Monday=0, Sunday=6

#Daily patterns: average by hour
hourly_avg = df.groupby('hour')[['CO(GT)', 'NOx(GT)',
'C6H6(GT)']].mean()

#Weekly patterns: average by day of week
weekly_avg = df.groupby('day_of_week')[['CO(GT)', 'NOx(GT)',
'C6H6(GT)']].mean()

#Plotting hourly averages
plt.figure(figsize=(16, 5))
sns.lineplot(data=hourly_avg)
plt.title('Average Pollutant Concentrations by Hour of Day')
plt.xlabel('Hour of Day')
plt.ylabel('Average Concentration')
```

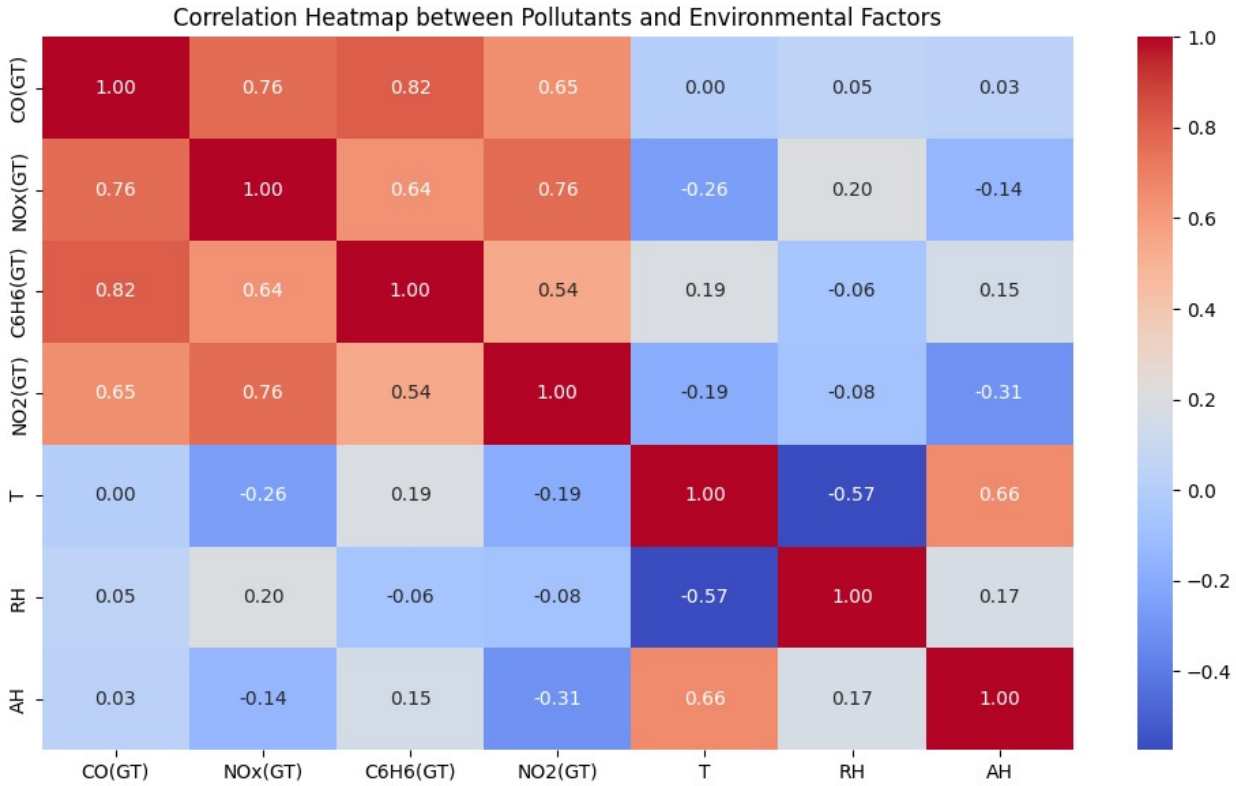
```
plt.grid(True)
plt.xticks(range(0, 24))
plt.tight_layout()
plt.show()

#Plotting weekly averages
plt.figure(figsize=(16, 5))
sns.lineplot(data=weekly_avg)
plt.title('Average Pollutant Concentrations by Day of Week')
plt.xlabel('Day of Week (0=Monday)')
plt.ylabel('Average Concentration')
plt.grid(True)
plt.xticks(range(0, 7), ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat',
'Sun'])
plt.tight_layout()
plt.show()
```



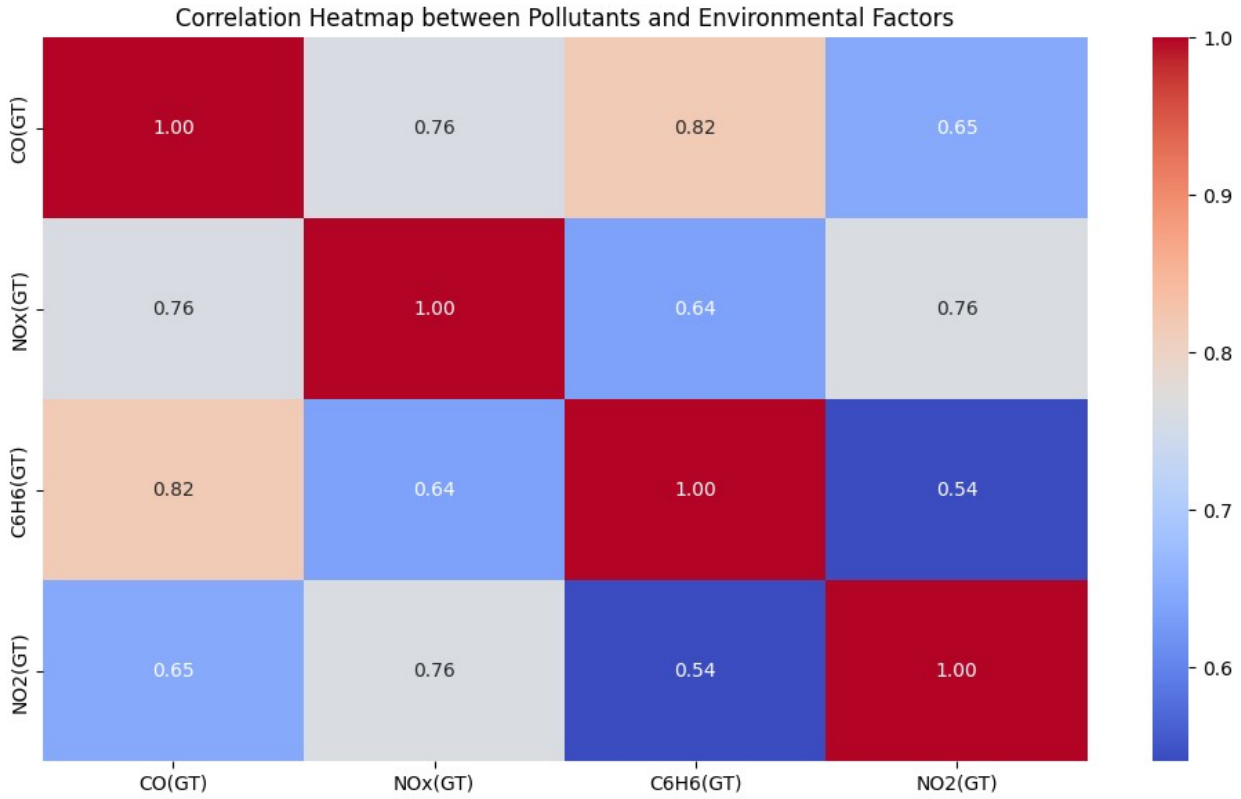



```
#Compute correlation matrix for selected pollutants
pollutant_cols = ['CO(GT)', 'NOx(GT)', 'C6H6(GT)', 'NO2(GT)', 'T',
'RH', 'AH']
corr_matrix = df[pollutant_cols].corr()

# Plot heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
```

```
plt.title('Correlation Heatmap between Pollutants and Environmental
Factors')
plt.tight_layout()
plt.show()
```

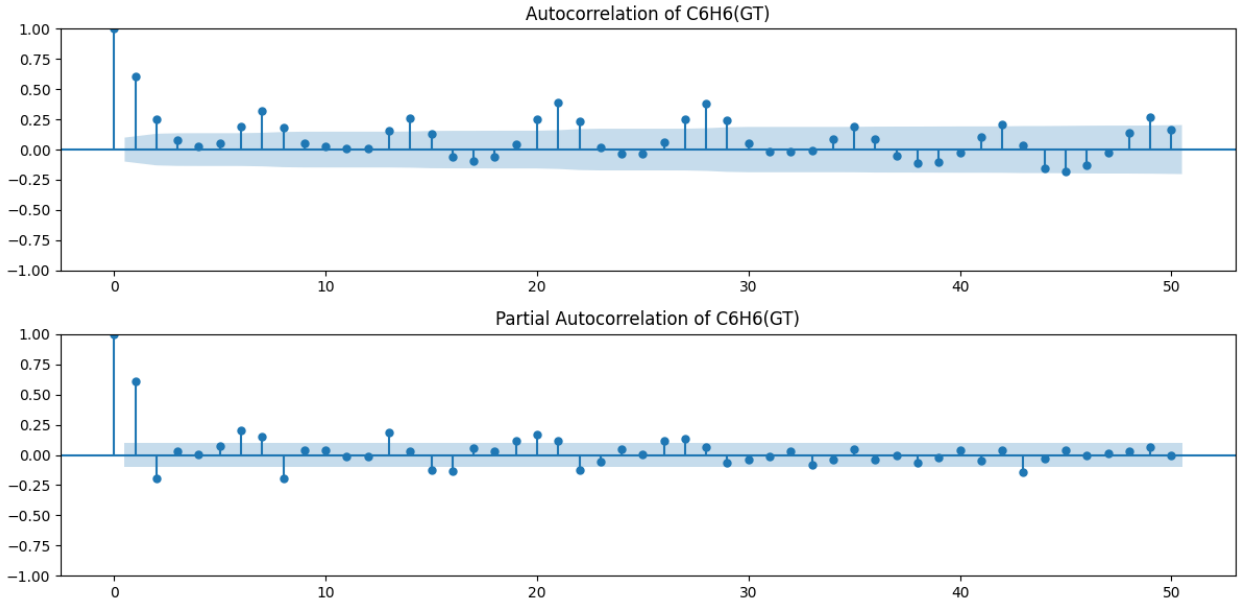



```
#Compute correlation matrix for selected pollutants
pollutant_cols = ['CO(GT)', 'NOx(GT)', 'C6H6(GT)', 'NO2(GT)']
corr_matrix = df[pollutant_cols].corr()

# Plot heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap between Pollutants and Environmental
Factors')
plt.tight_layout()
plt.show()
```

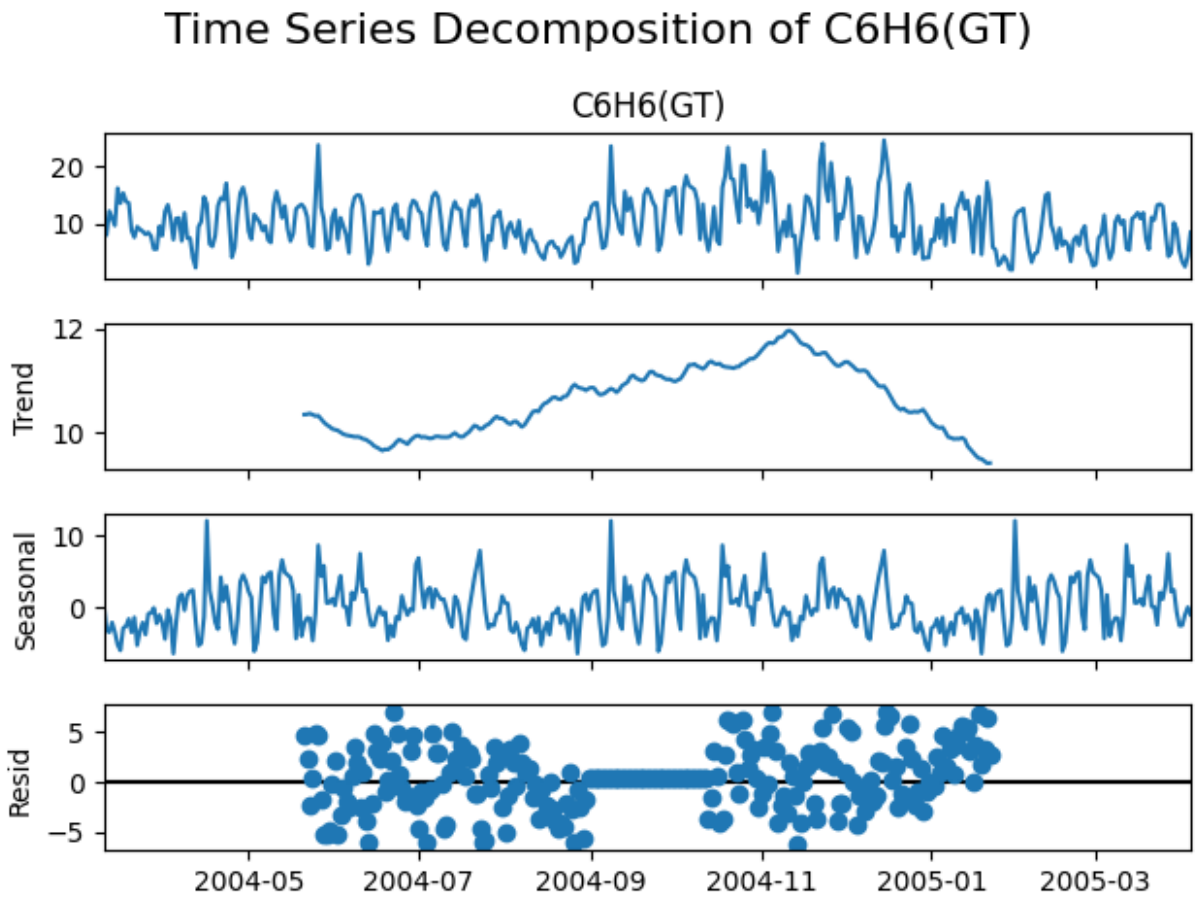


Advanced Analysis

```
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.seasonal import seasonal_decompose

data = pd.read_csv('/content/cleaned_data.csv',
parse_dates=['DateandTime'], index_col='DateandTime')

#Pollutant C6H6(GT) for Analysis
pollutant = 'C6H6(GT)'
data[pollutant] = pd.to_numeric(data[pollutant], errors='coerce')
data = data.dropna(subset=[pollutant])

daily_data = data[pollutant].resample('D').mean()

#Plot Autocorrelation and Partial Autocorrelation
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
plot_acf(daily_data, lags=50, ax=plt.gca(), title='Autocorrelation of
C6H6(GT)')
plt.subplot(2, 1, 2)
plot_pacf(daily_data, lags=50, ax=plt.gca(), title='Partial
Autocorrelation of C6H6(GT)')
plt.tight_layout()
plt.show()
```




```
#Time Series Decomposition
decomposition = seasonal_decompose(daily_data, model='additive',
period=145)  # Assuming yearly seasonality

#Plot
decomposition.plot()
plt.suptitle('Time Series Decomposition of C6H6(GT)', fontsize=16)
plt.tight_layout()
plt.show()
```


Time Series Decomposition of C6H6(GT)
C6H6(GT)
20
10
Trend
12
10
Seasonal
10
0
Resid
5
0
−5
2004-05
2004-07
2004-09
2004-11
2005-01
2005-03