

```
In [5]: from urllib import request
from shutil import unpack_archive
import pandas as pd
import numpy as np
from datetime import datetime

def RetriveData(baseurl, files):
    for i in range(1,31):
        if i < 10:
            url = str(baseurl) + "0" + str(i) + ".zip"
            request.urlretrieve(url, files[i-1])
            unpack_archive(filename = files[i-1], extract_dir = "D:\\data" )
        else:
            url = str(baseurl) + str(i) + ".zip"
            request.urlretrieve(url, files[i-1])
            unpack_archive(filename = files[i-1], extract_dir = "D:\\data" )

baseurl = "https://ci.taiwan.gov.tw/dsp/history/iis_airbox/202110/iis_airbox_202110"
files = []
for i in range(1,31):
    if i < 10:
        files.append("D:\\data\\iis_airbox_2021100" + str(i) + ".zip")
    else:
        files.append("D:\\data\\iis_airbox_202110" + str(i) + ".zip")

#RetriveData(baseurl, files)
```

In [6]:

```
def ExtractTaichungData(filename):
    df = pd.read_csv(filename)
    df["timestamp"] = pd.to_datetime(df["timestamp"], format="%Y-%m-%d %H:%M:%S")
    df = df[ df["SiteName"].str.contains("台中市") | df["SiteName"].str.contains("臺中市") ]
    df["month"] = df["timestamp"].dt.month
    df["day"] = df["timestamp"].dt.day
    df["hour"] = df["timestamp"].dt.hour
    tm = df.groupby(["month", "day", "hour"])
    tm = tm.agg(np.nanmean)
    tm = tm.rename(columns={"PM25": "台中市PM25"})
    return tm

def ExtractTaipeiData(filename):
    df = pd.read_csv(filename)
    df["timestamp"] = pd.to_datetime(df["timestamp"], format="%Y-%m-%d %H:%M:%S")
    df = df[ df["SiteName"].str.startswith("市立") ]
    df["month"] = df["timestamp"].dt.month
    df["day"] = df["timestamp"].dt.day
    df["hour"] = df["timestamp"].dt.hour
    tm = df.groupby(["month", "day", "hour"])
    tm = tm.agg(np.nanmean)
    tm = tm.rename(columns={"PM25": "台北市PM25"})
    return tm

def mergedata(a,b):
    con = pd.concat([a,b],axis=1)
    return con

a = ExtractTaichungData("D:\data\iis_airbox_20211001.zip")
b = ExtractTaipeiData("D:\data\iis_airbox_20211001.zip")
con = mergedata(a,b)
```

```

a = ExtractTaichungData("D:\data\iis_airbox_20211001.zip")
b = ExtractTaipeiData("D:\data\iis_airbox_20211001.zip")
con = mergedata(a,b)

for i in range(2,31):
    if i < 10:
        fn = "D:\data\iis_airbox_2021100" +str(i)+ ".csv"
    else:
        fn = "D:\data\iis_airbox_202110" +str(i)+ ".csv"

    a = ExtractTaichungData(fn)
    b = ExtractTaipeiData(fn)
    X = mergedata(a,b)
    con = pd.concat([con,X])

```

```

In [56]: con = con.drop(con[con["台中市PM25"] < 0].index)
con = con.drop(con[con["台北市PM25"] < 0].index)
con[con["台中市PM25"]<0]
con[con["台北市PM25"]<0]
### 因為太多極值，造成圖片視覺不佳

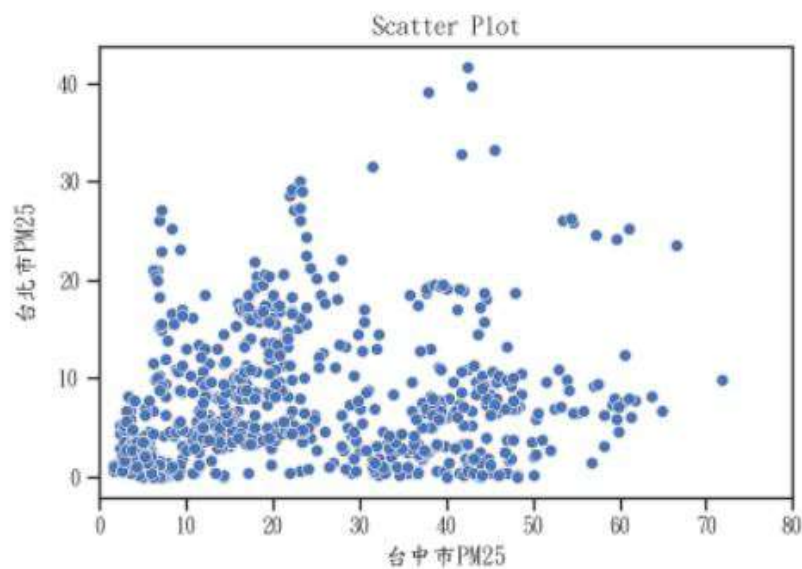
```

```

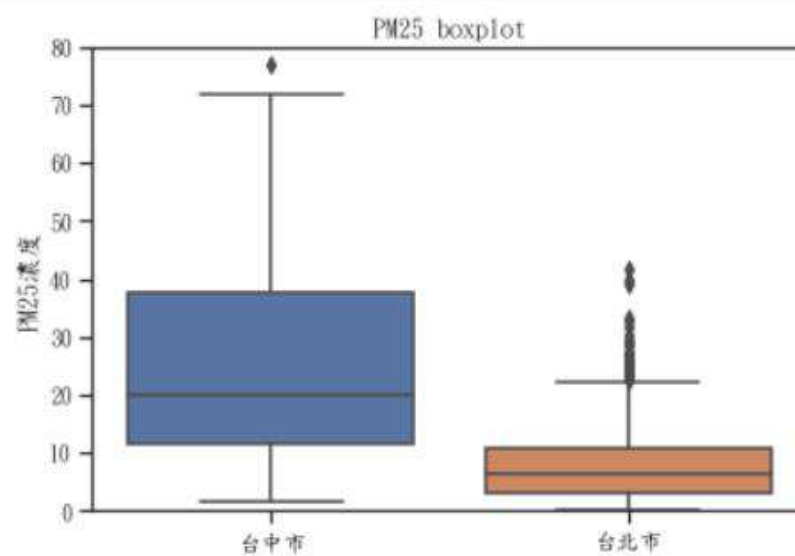
In [57]: import seaborn as sns
import matplotlib.pyplot as plt
plt.style.use("bmh")
%matplotlib inline
from matplotlib.font_manager import FontProperties

myfont = FontProperties(fname = "C:\Windows\Fonts\kaiu.ttf",size=14)
sns.set(font=myfont.get_name())
sns.set_theme(font=myfont.get_name(),style="ticks")
sns.scatterplot(x = "台中市PM25",y = "台北市PM25",data = con)
plt.title("Scatter Plot")
plt.xlim(0,80)
plt.show()

```



```
In [58]: dta = pd.DataFrame({"台中市":con["台中市PM25"],"台北市":con["台北市PM25"]})
sns.boxplot(data = dta)
plt.title("PM25 boxplot")
plt.ylabel("PM25濃度")
plt.ylim(0,80)
plt.show()
```



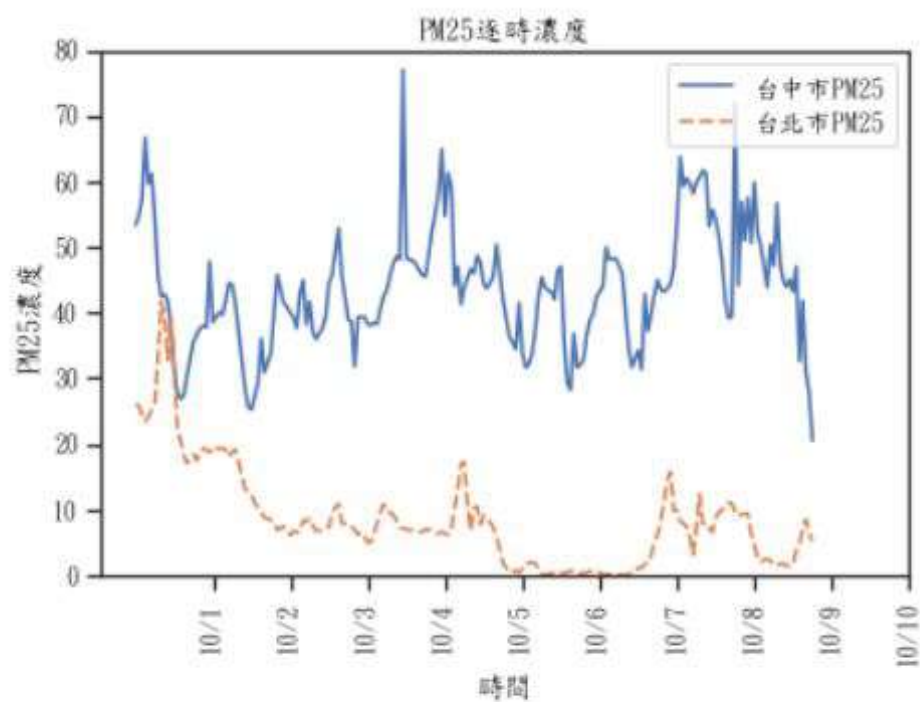
```
In [60]: #con.reset_index(inplace=True)
a1 = con[con.day < 11]
x = con [con.day > 10 ]
a2 = x[x.day < 21]
a3 = con[con.day > 20]

x1=[]
x2=[]
d1 = a1[["台中市PM25","台北市PM25"]]

for i in range(1,31):
    x1.append(i*24)
    x2.append("10/"+str(i))

sns.lineplot(data=d1)
plt.title("PM25逐時濃度")
plt.xlabel("時間")
plt.ylabel("PM25濃度")
plt.xticks(x1[0:10],x2[0:10],rotation="vertical")
plt.ylim(0,80)
plt.show()

#### 分成三張圖 防止距離太近，看不清楚，分別是 (10/1~10/10),(10/11~10/20),(10/21~10/30)
```

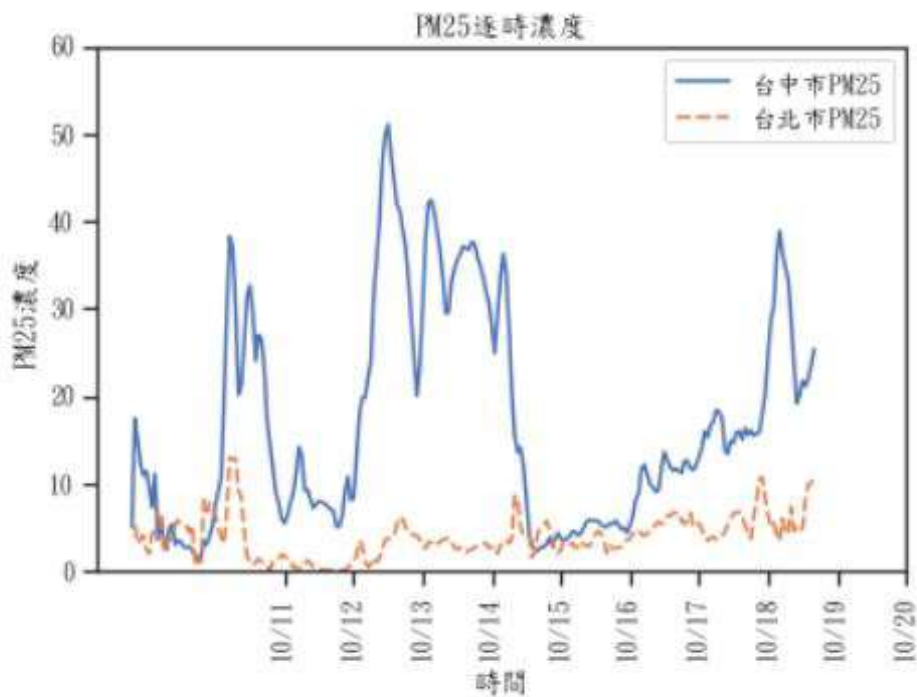


```
In [61]: d2 = a2[["台中市PM25", "台北市PM25"]]

sns.lineplot(data=d2)
plt.title("PM25逐時濃度")
plt.xlabel("時間")
plt.ylabel("PM25濃度")
plt.xticks(x1[10:20], x2[10:20], rotation="vertical")
plt.ylim(0, 60)
plt.show()
```

```
In [61]: d2 = a2[["台中市PM25", "台北市PM25"]]

sns.lineplot(data=d2)
plt.title("PM25逐時濃度")
plt.xlabel("時間")
plt.ylabel("PM25濃度")
plt.xticks(x1[10:20], x2[10:20], rotation="vertical")
plt.ylim(0, 60)
plt.show()
```





```

In [62]: d3 = a3[["台中市PM25", "台北市PM25"]]
d2 = a2[["台中市PM25", "台北市PM25"]]

sns.lineplot(data=d3)
plt.title("PM25逐時濃度")
plt.xlabel("時間")
plt.ylabel("PM25濃度")
plt.xticks(x1[20:31], x2[20:31], rotation="vertical")
plt.ylim(0, 40)
plt.show()

```

