1. 資料前處理

a. 取出 10.11.12 月資料

```
data = pd.read_excel("D://106 年新竹站_20180309.xls")
data = pd.DataFrame(data)
data = data.iloc[4914:6570,]
data = data.drop(['日期','測站', '測項'],axis=1)
```

Index	00	01	02	03	04	05	06
4914	29	29	29	28	28	28	28
4915	2	2	2	2	2	2	2
4916	0.41	0.37	0.3	0.3	0.25	0.26	0.35
4917	0.33	0.24	0.24	0.21	0.22	0.2	0.3
4918	0.7	0	0.2	0.2	-0.1	0	0.7
4919	24	19	16	15	14	14	16
4920	24	19	16	15	14	14	17
4921	15	11	16	15	15	14	14

b. 缺失值以及無效值以前後一小時平均值取代 / c. NR 表示無降雨，以 0 取代

```
def data_preprocess(data):
    data = data.replace('NR',0)
    data.iloc[:,0:] = data.iloc[:,0:].apply(pd.to_numeric,errors='coerce')
    temp_data = pd.DataFrame(data.values.reshape(1,-1))
    temp_f = temp_data.ffill(axis = 1)
    temp_b = temp_data.bfill(axis = 1)
    data = (temp_f + temp_b) / 2
    data = pd.DataFrame(data.values.reshape(1656,24))
    return data
data = data_preprocess(data)
```

Index	0	1	2	3	4	5	6
0	29	29	29	28	28	28	28
1	2	2	2	2	2	2	2
2	0.41	0.37	0.3	0.3	0.25	0.26	0.35
3	0.33	0.24	0.24	0.21	0.22	0.2	0.3
4	0.7	0	0.2	0.2	-0.1	0	0.7
5	24	19	16	15	14	14	16
6	24	19	16	15	14	14	17

d. 將資料切割成訓練集(10.11 月)以及測試集(12 月)

```
def train_test_split(data):
    train = data.iloc[:18*61,:]
    test = data.iloc[18*61:,:]
    return train, test
train, test = train_test_split(data)
```

test	DataFrame	(558, 24)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16 ...
train	DataFrame	(1098, 24)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16 ...

Index	0	1	2	3	4	5	6
0	29	29	29	28	28	28	28
1	2	2	2	2	2	2	2
2	0.41	0.37	0.3	0.3	0.25	0.26	0.35
3	0.33	0.24	0.24	0.21	0.22	0.2	0.3
4	0.7	0	0.2	0.2	-0.1	0	0.7

e. 製作時序資料:

將資料形式轉換為行(row)代表 18 種屬性，欄(column)代表逐時數據資料

```
def reshape_data(train, test):
    train = pd.DataFrame(train.values.reshape(18,-1))
    test = pd.DataFrame(test.values.reshape(18,-1))
    Index = ['AMB_TEMP','CH4','CO','NMHC','NO','NO2','NOx','O3','PM10',
    'PM2.5','RAINFALL','RH','SO2','THC','WD_HR','WIND_DIREC',
    'WIND_SPEED','WS_HR']
    train.index = Index
    test.index = Index
    return train, test
train, test = reshape_data(train, test)
```

test	DataFrame	(18, 744)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 ...
train	DataFrame	(18, 1464)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 ...

Index	0	1	2	3	4	5	6
AMB_TEMP	29	29	29	28	28	28	28
CH4	39	39	44	43	42	41	40
CO	72	71	65	76	67	65	66
NMHC	0.07	0.08	0.04	0.06	0.11	0.08	0.08
NO	3.2	3.2	0.4	1.6	1.6	0.2	0
NO2	2.8	2.5	2.8	3.2	3.3	2.8	2.8

2. 時間序列

a. 取 6 小時為一單位切割 / b. X 請分別取 PM2.5 和所有 18 種屬性

```
def generate_data(data, label):
    for i in range(data.shape[1]-6):
        if label == 'pm':
            if i==0:
                X = data.iloc[:,0:6].values
            else:
                temp = data.iloc[:,i:i+6].values
                X = np.concatenate((X,temp),axis=0)
        if label == 'all':
            if i==0:
                X = data.iloc[:,0:6].values
            else:
                temp = data.iloc[:,i:i+6].values
                X = np.concatenate((X,temp),axis=0)
    Y = data.iloc[:,6:].T.values
    return X, Y
X_train, Y_train = generate_data(X_train, ’pm’)
X_test, Y_test = generate_data(X_test, ‘all’)
```

Pm_X_test	DataFrame	(1, 744)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 ...
Pm_X_train	DataFrame	(1, 1464)	Column names: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16 ...

c. 使用兩種模型 Linear Regression 和 Random Forest Regression 建模 / d. 用測試集資料計算 MAE (會有 4 個結果，2 種模型*2 種 X 資料)

```
def fit(X_train, X_test, label):
    X_train, Y_train = generate_data(X_train, label)
    X_test, Y_test = generate_data(X_test, label)
    lr = LinearRegression().fit(X_train, Y_train.ravel())
    lr_pred = lr.predict(X_test)
    lr_MAE = mean_absolute_error(Y_test.ravel(), lr_pred)
    print(label," LR: ",lr_MAE)

    rf = RandomForestRegressor(n_estimators=30).fit(X_train,
Y_train.ravel())
    rf_pred = rf.predict(X_test)
    rf_MAE = mean_absolute_error(Y_test.ravel(), rf_pred)
    print(label," RF: ",rf_MAE)
    return lr_MAE, rf_MAE

fit(Pm_X_train, Pm_X_test, 'pm')
fit(train, test, 'all')
```

```
pm  LR:  3.5132751101108184
pm  RF:  3.267240284320071
all  LR:  4.870972241167576
all  RF:  4.6170097414517555
```

	PM2.5	18 種屬性
Linear Regression	3. 51327	3. 26724
Random Forest Regression	4. 87097	4. 61701

結論：

不論是只有 PM2.5 或者是 18 種屬性的資料，使用 Linear Regression 的模型得到的誤差都比 Random Forest Regression 的模型低，Linear Regression 表現優於 Random Forest Regression