1. Data preprocessing_EXTRACT

Step 1: extract 10~12 month data from the whole year

```
In [2]: raw_data = pd.read_csv("/Users/lnl/Documents/DM_data_air_2019.csv")
        raw_data.head()
```

Out[2]:

	date	variable	0	1	2	3	4	5	6	7	...	14	15	16	17
0	2019/1/1 00:00	AMB_TEMP	17.5	17.4	17.3	17.1	16.8	16.9	16.8	17.1	...	20	19.5	18.7	18.1
1	2019/1/1 00:00	CH4	1.79	1.78	1.8	1.8	1.8	1.8	1.8	1.8	...	1.81	1.81	1.81	1.8
2	2019/1/1 00:00	CO	0.19	0.21	0.22	0.22	0.21	0.21	0.23	0.25	...	0.29	0.3	0.31	0.31
3	2019/1/1 00:00	NMHC	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.05	...	0.06	0.06	0.07	0.07
4	2019/1/1 00:00	NO	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	...	1.1	1.2	0.7	0.6

5 rows x 26 columns

Step 2: slice the month by the date

```
In [4]: raw_data['month'] = pd.DatetimeIndex(raw_data['date']).month
        raw_data
```

Step 3: extract only 10.11.12 these three months

```
In [6]: octdecmonth = ['10', '11','12']
        data = raw_data[raw_data['month'].isin(octdecmonth)]
        data
```

2. TRANSFORM

Step 4: withdraw NaN

```
In [9]: data.isna().sum()
```

Step 5: detect the data feature

```
In [7]: del data['month']
        data.shape
Out[7]: (1656, 26)
```

```
In [10]: len(data['variable'].value_counts())
Out[10]: 18
```

```
In [11]: data_023 = data.drop(columns=['date', 'variable'])
         data_023.head()
```

Out[11]:

	0	1	2	3	4	5	6	7	8	9	...	14	15	16	17	18
4824	24.7	25.1	25.4	25.5	25.3	25.1	24.6	24.6	25.1	26.8	...	32.4	31.9	30.4	29.1	28.7
4825	1.66	1.66	1.7	1.71	1.72	1.71	1.75	1.76	1.73	1.73	...	1.69	1.72	1.74	1.74	1.78
4826	0.05	0.13	0.15	0.17	0.16	0.16	0.22	0.42	0.36	0.29	...	0.27	0.32	0.36	0.39	0.49
4827	0	0	0	0.02	0.02	0.01	0.03	0.08	0.08	0.08	...	0.07	0.1	0.08	0.08	0.13
4828	0	0.3	0.3	0.3	0.3	0.3	1.2	5.3	5.6	4.8	...	1.6	1.1	0.8	0.6	0.6

5 rows x 24 columns

Print them3

```
In [13]: for (columnName, columnData) in data_023.iteritems():
             print('Colunm Name : ', columnName)
             col_columnName = data_023.loc[columnData.str.contains('#|\*|x|A')]
             print(col_columnName[columnName])
             col_columnName_index = col_columnName[columnName].index
             print(col_columnName_index)
             print("!!!!!!!!!!","number of missing/invalid value in row_",columnNa
         me," : ", len(col_columnName[columnName]),"!!!!!!!!!")
             data_023.loc[col_columnName[columnName].index,columnName] = np.nan
             print(data_023.loc[col_columnName[columnName].index,columnName])
             print("------------------------------------------------------------
         ----")
             print("------------------------------------------------------------
         ----")
```

```
Colunm Name :  0
5168    0.23#
5170     0.3#
5171     5.4#
5172     5.5#
5178     1.4#
Name: 0, dtype: object
Int64Index([5168, 5170, 5171, 5172, 5178], dtype='int64')
!!!!!!!!! number of missing/invalid value in row_ 0  :  5 !!!!!!!!!
5168    NaN
5170    NaN
5171    NaN
5172    NaN
5178    NaN
Name: 0, dtype: object
------------------------------------------------------------------
------------------------------------------------------------------
Colunm Name :  1
Series([], Name: 1, dtype: object)
Int64Index([], dtype='int64')
!!!!!!!!! number of missing/invalid value in row_ 1  :  0 !!!!!!!!!
Series([], Name: 1, dtype: object)
------------------------------------------------------------------
------------------------------------------------------------------
```

```
In [14]: data_023.isna().sum()
```

```
Out[14]: 0       5
         1       0
         2       0
         3       0
         4       6
         5       6
         6       3
         7       5
         8       8
         9      16
         10     22
         11     54
         12     30
         13     26
         14     25
         15     30
         16     24
         17     15
         18      3
         19      3
         20      1
         21      4
         22      7
         23      0
         dtype: int64
```

Step6: Transpose

```
In [118]: data_023_reindex = data_023.reset_index(drop = True)
          data_023_reindex
```

```
In [123]: df = pd.DataFrame()
          for i in range(data_023.shape[0]//18+1):
              data_i = data_023_reindex[18*(i-1):18*i][data_023_reindex.columns]
              data_i_reindex = data_i.reset_index(drop = True)
              data_i_tr = data_i_reindex.transpose()
              df = df.append(data_i_tr)
              df = df.reset_index(drop = True)
              i = i + 1
          df
```

```
In [124]: df_new = df.drop(df.index[:24])
          df_new = df_new.reset_index(drop = True)
          df_new
```

Out[124]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	24.7	1.66	0.05	0	0	1.3	1.2	16.7	18	2	0	89	1.1	1.65	292	283	2.1	0.9
1	25.1	1.66	0.13	0	0.3	1.1	1.2	17.3	18	4	0	87	1.3	1.65	286	289	1.4	1.1
2	25.4	1.7	0.15	0	0.3	1.3	1.7	21.9	29	6	0	85	1.4	1.7	291	276	2	1.8
3	25.5	1.71	0.17	0.02	0.3	1.8	2.1	21.5	29	9	0	84	1.5	1.73	286	289	2.5	2.2
4	25.3	1.72	0.16	0.02	0.3	2.1	2.4	18.8	27	10	0	85	1.5	1.74	268	250	1.8	1.8
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2203	15.1	1.75	0.36	0.09	0.8	10.8	11.5	33.3	22	12	0	72	1.4	1.84	51	47	4.4	3.7
2204	15	1.74	0.31	0.07	0.3	8.8	9.2	34.1	24	13	0	73	1.4	1.81	57	64	4.1	3.5
2205	15	1.74	0.3	0.07	0.8	8.6	9.3	34.1	27	10	0	72	1.7	1.81	50	40	4.4	3.1
2206	15	1.74	0.29	0.07	0.6	7.8	8.4	34.1	28	12	0	72	1.6	1.81	49	55	3.7	3.2
2207	15.2	1.74	0.29	0.06	0.3	6.9	7.3	33.7	30	11	0	73	1.4	1.8	53	52	4.6	3.6

2208 rows x 18 columns

Step7: NaN transform by the former or the latter

```
In [126]: df_new_ffill = df_new.fillna(method='ffill')
          df_new_bfill = df_new.fillna(method='bfill')
          df_new_ffill = df_new_ffill.astype(float)
          df_new_bfill = df_new_bfill.astype(float)

          df_fill = (df_new_ffill + df_new_bfill)/2
```

```
In [127]: df_fill.isna().sum()

Out[127]: 0     0
          1     0
          2     0
          3     0
          4     0
          5     0
          6     0
          7     0
          8     0
          9     0
          10    0
          11    0
          12    0
          13    0
          14    0
          15    0
          16    0
          17    0
          dtype: int64
```

3. LOAD THE DATA

Step 8: Slice the data to become training and testing

In [158]:
```
month_10 = pd.DataFrame({'month' : np.tile(10, 744)})
month_11 = pd.DataFrame({'month' : np.tile(11, 720)})
month_12 = pd.DataFrame({'month' : np.tile(12, 744)})
month_df = pd.concat([month_10, month_11, month_12], ignore_index=True)
month_df
```

In [161]:
```
df_month = pd.concat([month_df, df_fill], axis=1)
df_month
```

In [166]:
```
octdecmonth = ['10', '11']
train = df_month[df_month['month'].isin(octdecmonth)]
train.drop('month',axis = 1, inplace = True)
train
```

In [167]:
```
test = df_month[df_month['month'].isin(['12'])]
test.drop('month',axis = 1, inplace = True)
test
```