HW2

谢婷 12532753

1. Significant earthquakes since 2150 B.C.

（1）Import the work library first.

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

（2）Read the file and view its basic information.

Sig_Eqs.head()

	Search Parameters	Id	Year	Mo	Dy	Hr	Mn	Sec	Tsu	Vol	...	Total Missing	Total Missing Description
0	[]	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
1	NaN	1.0	-2150.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
2	NaN	2.0	-2000.0	NaN	NaN	NaN	NaN	NaN	1.0	NaN	...	NaN	NaN
3	NaN	3.0	-2000.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN
4	NaN	5877.0	-1610.0	NaN	NaN	NaN	NaN	NaN	3.0	1351.0	...	NaN	NaN

5 rows × 49 columns

Sig_Eqs.info()

```
Data columns (total 49 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Search Parameters                   1 non-null      object
 1   Id                                  6615 non-null   float64
 2   Year                                6615 non-null   float64
 3   Mo                                  6202 non-null   float64
 4   Dy                                  6045 non-null   float64
 5   Hr                                  4502 non-null   float64
 6   Mn                                  4289 non-null   float64
 7   Sec                                 3831 non-null   float64
 8   Tsu                                 2083 non-null   float64
 9   Vol                                 81 non-null     float64
 10  Country                             6615 non-null   object
 11  Area                                332 non-null    object
 12  Region                              6615 non-null   float64
 13  Location Name                       6615 non-null   object
 14  Latitude                            6554 non-null   float64
 15  Longitude                           6554 non-null   float64
 16  Focal Depth (km)                    3574 non-null   float64
 17  Mag                                 4823 non-null   float64
 18  Mw                                  1845 non-null   float64
 19  Ms                                  3087 non-null   float64
 20  Mb                                  1880 non-null   float64
 21  Ml                                  216 non-null    float64
 22  Mfa                                 14 non-null     float64
 23  Unk                                 824 non-null    float64
 24  MMI Int                             3343 non-null   float64
 25  Deaths                              2244 non-null   float64
 26  Death Description                   2748 non-null   float64
 27  Missing                             24 non-null     float64
 28  Missing Description                 26 non-null     float64
 29  Injuries                            1451 non-null   float64
 30  Injuries Description                1699 non-null   float64
 31  Damage ($Mil)                       631 non-null    float64
 32  Damage Description                  4833 non-null   float64
 33  Houses Destroyed                    923 non-null    float64
 34  Houses Destroyed Description        1990 non-null   float64
 35  Houses Damaged                      610 non-null    float64
 36  Houses Damaged Description          1187 non-null   float64
 37  Total Deaths                        2112 non-null   float64
 38  Total Death Description             2526 non-null   float64
 39  Total Missing                       27 non-null     float64
 40  Total Missing Description           32 non-null     float64
 41  Total Injuries                      1476 non-null   float64
 42  Total Injuries Description          1730 non-null   float64
 43  Total Damage ($Mil)                 620 non-null    float64
 44  Total Damage Description            4050 non-null   float64
 45  Total Houses Destroyed              957 non-null    float64
 46  Total Houses Destroyed Description  2075 non-null   float64
 47  Total Houses Damaged                558 non-null    float64
 48  Total Houses Damaged Description    1094 non-null   float64
dtypes: float64(45), object(4)
memory usage: 2.5+ MB
```

1.1 [5 points] Compute the total number of deaths caused by earthquakes since 2150 B.C. in each country, and then print the top ten countries along with the total number of deaths.

Classify the data by country, then calculate the total number of deaths from earthquakes since 2150 BC. Sort the results in descending order and print the top 10 countries.

```
Sig_Eqs.groupby('Country')['Deaths'].sum().sort_values(ascending=False)[0:10]
```

The results are as follows:

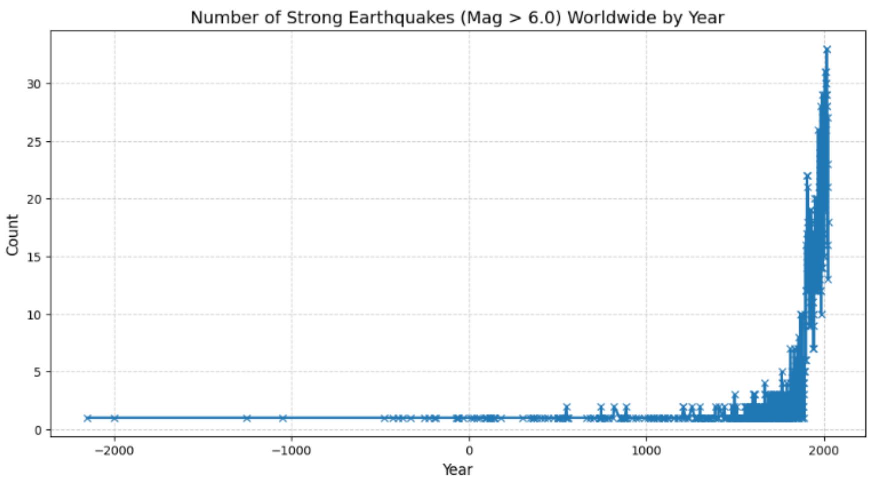
```
Country
CHINA         2139210.0
TURKEY        1199742.0
IRAN          1014453.0
ITALY          498219.0
SYRIA          419226.0
HAITI          323484.0
AZERBAIJAN     319251.0
JAPAN          242445.0
ARMENIA        191890.0
PAKISTAN       145083.0
Name: Deaths, dtype: float64
```

1.2 [10 points] Compute the total number of earthquakes with magnitude larger than 6.0 (use column Mag as the magnitude) worldwide each year, and then plot the time series. Do you observe any trend? Explain why or why not?

Group data with a magnitude greater than 6 by year, then calculate the total number of earthquakes.

```
Sig_Eqs[Sig_Eqs['Mag'] > 6.0].groupby('Year').size()
```

Plot the time series chart



In terms of trends, before 1800, the total number of earthquakes with a magnitude greater than 6 on a global scale was at a relatively low level. However, after 1800, the number of such earthquakes soared sharply, even exceeding 30 in some periods.

The reasons are as follows. Before 1800, seismic monitoring technology was backward and instruments were scarce, making it difficult to accurately record many earthquakes. Meanwhile, human activities had a small scope and low intensity, having limited impacts on the internal stress of the Earth. After 1800, technological advancements improved monitoring capabilities, and large - scale urban construction and resource exploitation activities by humans increased, altering the Earth's stress state and thus leading to a rise in the number of earthquakes.

1.3 [10 points]Write a function CountEq_LargestEq that returns both (1) the total number of earthquakes since 2150 B.C. in a given country AND (2) the date of the largest earthquake ever happened in this country. Apply CountEq_LargestEq to every country in the file, report your results in a descending order.

Data preprocessing: First, process the date data by removing missing date values, converting the date data type, and merging the year, month, and day into a single field. Next, convert missing magnitude values to negative values to avoid affecting the determination of the maximum magnitude. Finally, remove missing values from the country column.

```
# Handling Date  Missing Values
Sig_Eqs=Sig_Eqs.dropna(subset=['Year','Mo','Dy']).copy()
#change date type
Sig_Eqs['Mo'] = Sig_Eqs['Mo'].astype(int)
Sig_Eqs['Dy'] = Sig_Eqs['Dy'].astype(int)
Sig_Eqs['Year'] = Sig_Eqs['Year'].astype(int)
#merge date
Sig_Eqs['Date']=pd.to_datetime({'year':Sig_Eqs['Year'],'month':Sig_Eqs['Mo'],'day':Sig_
#delete missing data
Sig_Eqs=Sig_Eqs.dropna(subset=['Date'])

# Processing massing mag
Sig_Eqs['Mag']=Sig_Eqs['Mag'].fillna(-np.inf)

#Processing country
Sig_Eqs=Sig_Eqs.dropna(subset=['Country'])
```

Define function：return total_eqs,largest_date

```
#1.3Define function
def CountEq_LargestEq(country):
    country_eqs=Sig_Eqs[(Sig_Eqs['Country']==country)]
    total_eqs=len(country_eqs)
    if total_eqs>0:
        largest_eq=country_eqs.sort_values('Mag',ascending=False).iloc[0]
        largest_date=largest_eq['Date']
    else:
        largest_date=None
    return(total_eqs, largest_date)
```

Apply function：

```python
#1.3.3 apply function
countries=Sig_Eqs['Country'].unique()
results={}
for country in countries:
    count_and_date = CountEq_LargestEq(country)
    results[country]=count_and_date

result_df=pd.DataFrame.from_dict(results,orient='index',columns=['Total Earthquakes','I

result_df=result_df.sort_values('Total Earthquakes',ascending=False).reset_index()
result_df=result_df.rename(columns={'index':'Country'})

#result
print(result_df)
```

```
         Country  Total Earthquakes Largest Earthquake Date
0          CHINA                476              1906-12-22
1      INDONESIA                406              2004-12-26
2          JAPAN                348              2011-03-11
3           IRAN                280              2013-04-16
4            USA                275              1964-03-28
..           ...                ...                     ...
149   TASMAN SEA                  1              1892-01-26
150         TOGO                  1              1933-05-19
151   MONTSERRAT                  1              1897-04-25
152     KIRIBATI                  1              1905-06-30
153      COMOROS                  1              2018-05-15

[154 rows x 3 columns]
```

2. Wind speed in Shenzhen from 2010 to 2020

Import the work library first， and read the file and view its basic information.

```python
#读取文件
WDSZ=pd.read_csv('2281305.csv')
WDSZ.info()
```

```python
WDSZ.head()
```

	STATION	DATE	SOURCE	REPORT_TYPE	CALL_SIGN	QUALITY_CONTROL	AA
0	59493099999	2010-01-02T00:00:00	4	SY-MT	ZGSZ	V020	06,0000,2,
1	59493099999	2010-01-02T01:00:00	4	FM-15	ZGSZ	V020	Na
2	59493099999	2010-01-02T02:00:00	4	FM-15	ZGSZ	V020	Na
3	59493099999	2010-01-02T03:00:00	4	SY-MT	ZGSZ	V020	Na
4	59493099999	2010-01-02T04:00:00	4	FM-15	ZGSZ	V020	Na

5 rows × 43 columns

Extract the two required columns, split the [WND] column into five separate columns, and delete the original column to facilitate subsequent data filtering.

	DATE	WIND-OBSERVATION direction angle	WIND-OBSERVATION direction quality code	WIND-OBSERVATION type code	WIND-OBSERVATION speed rate	WIND-OBSERVATION speed quality code
0	2010-01-02T00:00:00	040	1	N	0020	1
1	2010-01-02T01:00:00	999	9	V	0010	1
2	2010-01-02T02:00:00	999	9	C	0000	1
3	2010-01-02T03:00:00	140	1	N	0010	1
4	2010-01-02T04:00:00	300	1	N	0040	1
...	...	...	...	...	...	...
111979	2020-09-11T17:00:00	170	1	N	0030	1
111980	2020-09-11T18:00:00	180	1	N	0040	1
111981	2020-09-11T19:00:00	220	1	V	0030	1
111982	2020-09-11T20:00:00	260	1	N	0030	1
111983	2020-09-11T21:00:00	310	1	V	0020	1

111984 rows × 6 columns

filter the data: delete 'WIND-OBSERVATION direction quality code'、'WIND-OBSERVATION speed quality code' the data points with poor quality: 2, 3, 6, and 7.

Missing values (‘999’) in ‘WIND-OBSERVATION speed rate’ should be deleted.

```
WDSZ2['WIND-OBSERVATION direction quality code'] =WDSZ2[
    'WIND-OBSERVATION direction quality code'].replace([
    '2','3','6','7'],pd.NA).dropna().astype(int)


WDSZ2['WIND-OBSERVATION speed rate'] = WDSZ2[
    'WIND-OBSERVATION speed rate'].replace('999',pd.NA).dropna().astype(int)


WDSZ['WIND-OBSERVATION speed quality code'] = WDSZ2[
    'WIND-OBSERVATION speed quality code'].replace([
    '2','3','6','7'],pd.NA).dropna().astype(int)
```

2.1[10 points] Plot monthly averaged wind speed as a function of the observation time. Is there a trend in monthly averaged wind speed from 2010 to 2020?

Create a new column for month and year. Group the wind speed column by month and year, calculate the average, and plot the data.

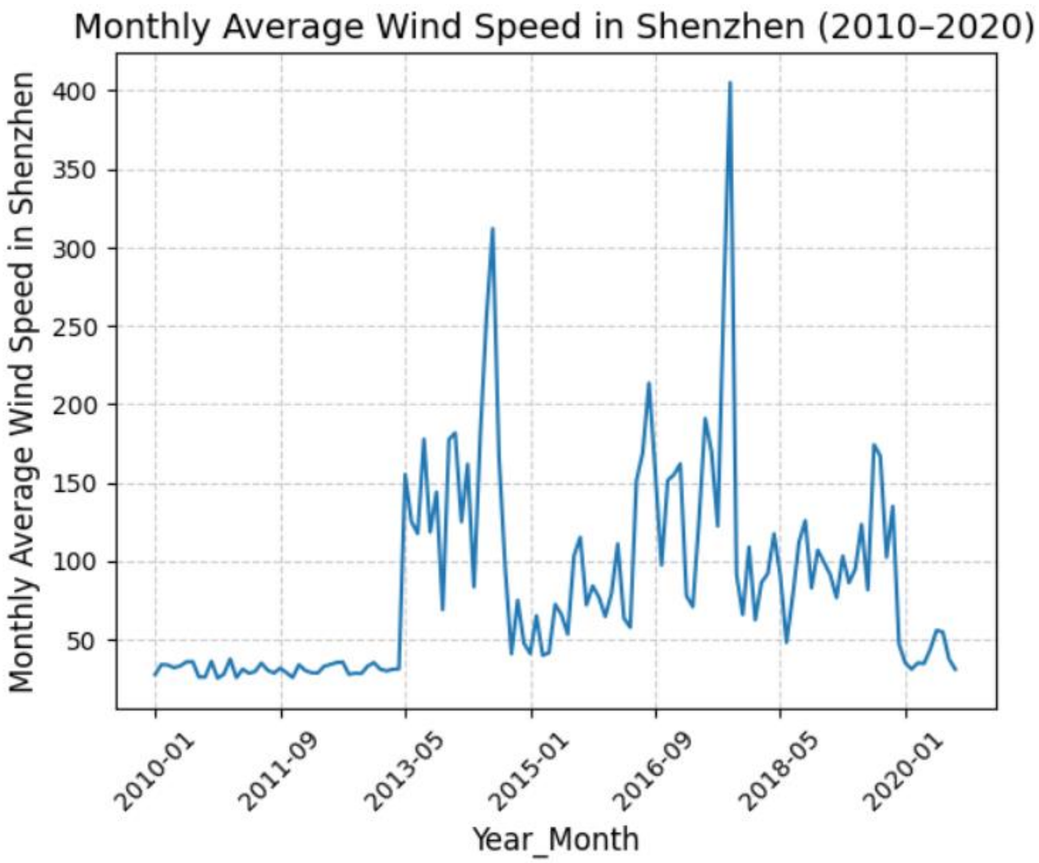
```
WDSZ2['DATE']=pd.to_datetime(WDSZ2['DATE'])

WDSZ2['year_month'] = WDSZ2['DATE'].dt.strftime('%Y-%m')

month=WDSZ2['WIND-OBSERVATION speed rate'].groupby(WDSZ2['year_month']).mean().plot()

plt.xticks(rotation=45)
plt.title('Monthly Average Wind Speed in Shenzhen (2010–2020)', fontsize=14)
plt.xlabel('Year_Month', fontsize=12)
plt.ylabel('Monthly Average Wind Speed in Shenzhen', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.6)
```

Results as follow:



Monthly average wind speeds in Shenzhen fluctuated significantly between 2010 and 2020, with multiple peaks occurring during this period (such as in 2015 and 2016). However, no clear long-term upward or downward trend was observed overall. Wind speeds in both 2010 and 2020 remained at relatively low levels, primarily exhibiting interannual variability.

3. Explore a data set

Data Source: https://data.casearth.cn/dataset/5feae826819aec33049b7cc7

3.1 [5 points] Load the csv, XLS, or XLSX file, and clean possible data points with missing values or bad quality.

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
!pip install openpyxl
```

```
data=pd.read_excel("2018 Lanzhou University Cold and Arid Regions Scientific Observati
```

```
data.head()
```

	Date/Time	Wdir	Wnd	Std_uy	Tv	H2O	CO2	Ustar	L
0	-6999	°	m/s	m/s	°C	g/m3	mg/m3	[m+1s-1]	m
1	2018-09-23 00:30:00	8.819528	305.542536	0.772491	15.684483	8.819528	305.542536	0.379347	97.322818
2	2018-09-23 01:00:00	8.202629	302.645823	0.694023	15.270833	8.202629	302.645823	0.351198	72.222056
3	2018-09-23 01:30:00	7.617878	300.314145	0.627909	15.081841	7.617878	300.314145	0.307679	56.211913
4	2018-09-23 02:00:00	7.609132	300.266224	0.631314	15.026563	7.609132	300.266224	0.298211	46.919563

Remove all missing values (-6999) from the data. The number 9 in this dataset represents data gaps, so eliminate all points with data gaps.

```
data=data.replace(-6999,pd.NA).dropna()

data['QA_Hs']=data['QA_Hs'].replace(9,pd.NA).dropna()
data['QA_LE']=data['QA_LE'].replace(9,pd.NA).dropna()
data['QA_Fc']=data['QA_Fc'].replace(9,pd.NA).dropna()

data
```

3.2 [5 points] Plot the time series of a certain variable.

Split the time using the method from the previous question to facilitate creating a time series chart.

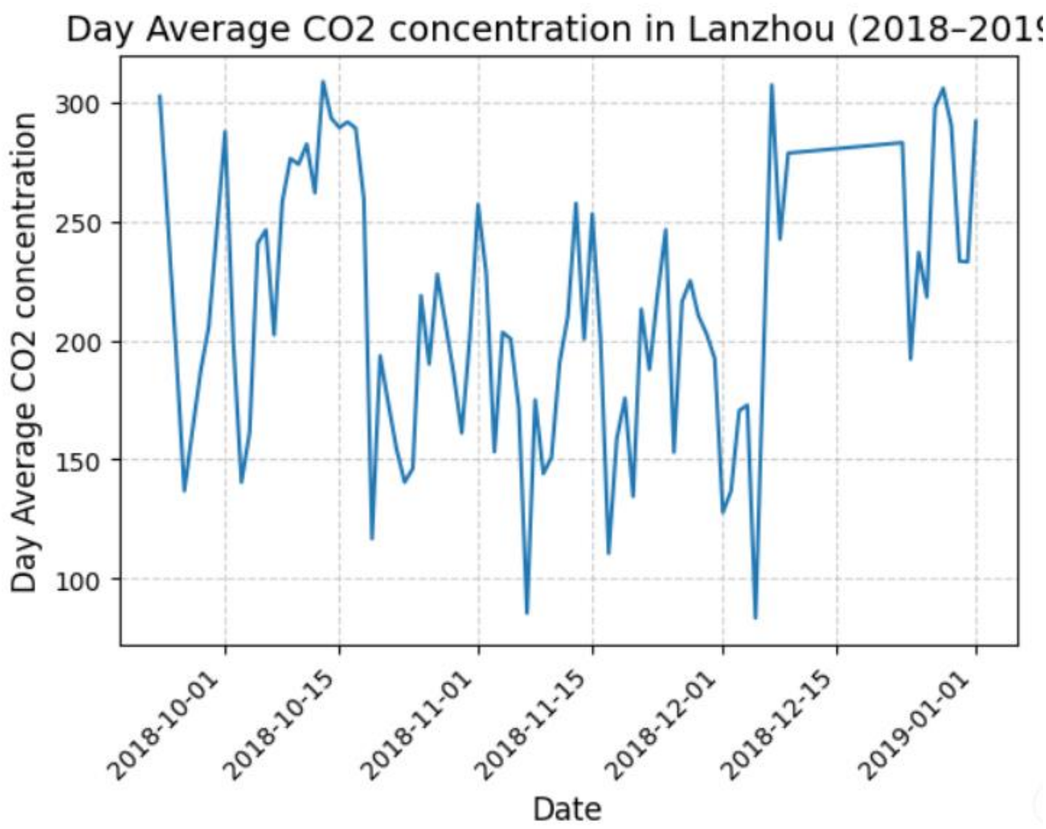
```
#3.2
data['Date/Time']=data['Date/Time'].astype(str)
data[['Date', 'Time']] = data['Date/Time'].str.split(' ', expand=True)

data
```

```
data['Date']=pd.to_datetime(data['Date'])
```

```
data_=data['CO2'].groupby(data['Date']).mean().plot()
data_
plt.xticks(rotation=45)
plt.title('Day Average CO2 concentration in Lanzhou (2018 - 2019)', fontsize=14)
plt.xlabel('Date', fontsize=12)
plt.ylabel('Day Average CO2 concentration', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.6)
```

Plot a time series graph of CO2 concentration.



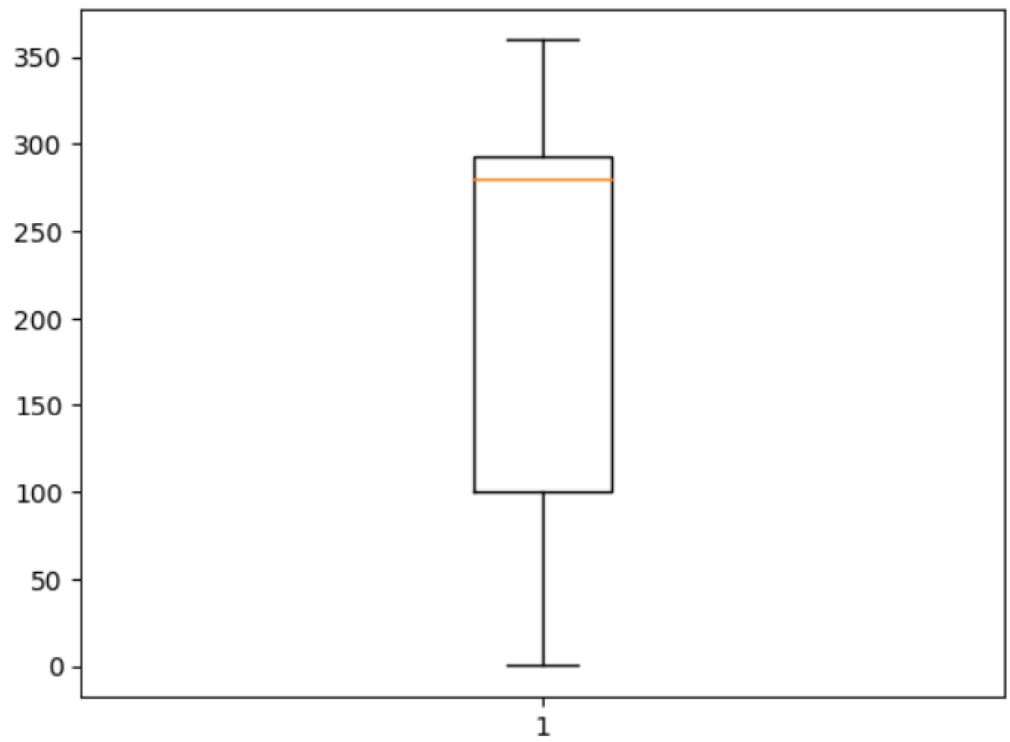
3.3 [5 points] Conduct at least 5 simple statistical checks with the variable, and report your findings.

```
#3.3
print('max:',data['CO2'].max())
print('min:',data['CO2'].min())
print('mean:',data['CO2'].mean())
print(data['CO2'].describe())
print('标准差：',data['CO2'].std())
```

```
max: 359.714003166053
min: 0.053299356419432
mean: 208.91937077373453
count      3653.000000
unique     3653.000000
top         305.542536
freq          1.000000
Name: CO2, dtype: float64
标准差： 104.5320950222388
```

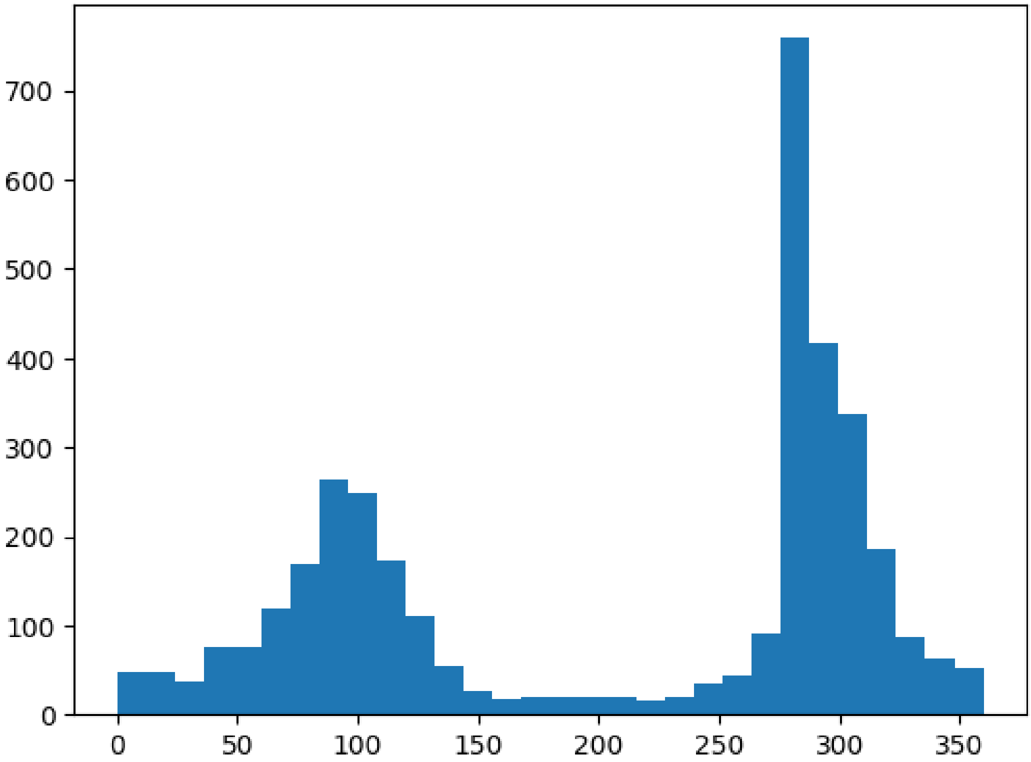
箱型图如下：

```
plt.boxplot(data['CO2'])
plt.show()
print('图形分析：箱型图中可以得出没有异常值')
```



图形分析：箱型图中可以得出没有异常值

直方图如下：



CO_2的浓度在 0.05—359.71 mg/m^{3}之间，经过滤之后有 3653 个数据，众数在 305.54 mg/m^3，平均数为 208.92 mg/m^3，标准差为 104.53。从箱型图中可看出并无异常值，从直方图中可看出在 300 mg/m^{3}左右的分布较多。