In [59]:

```
#由于对时间的把控和对画图工具极不熟悉，本次作业没能完成，做的稀里糊涂的，可能助教看起来也很费劲，能
#第一题
import pandas as pd
import numpy as np
import seaborn as sns
from matplotlib import pyplot as plt
Sig_Eqs=pd.read_csv('earthquakes-2022-10-28_09-18-54_+0800.tsv',sep='\t')
#去掉没有数据的第一行
Sig_Eqs =Sig_Eqs.drop([0])
Sig_Eqs
Sig_Eqs['country']=Sig_Eqs['Location Name'].str.split(':').str.get(0)
Sig_Eqs
Sig_Eqs.groupby('country')['Total Deaths'].sum()
```

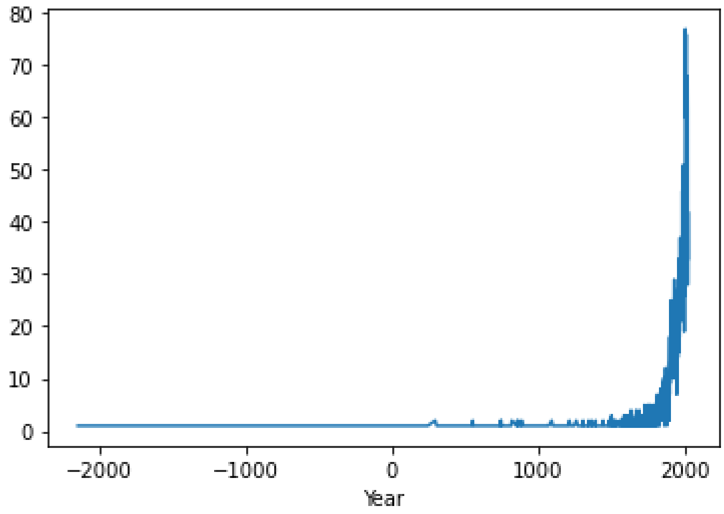
```
INDONESIA; MALAYSIA                                  0.0
INSTANBUL (CONSTANTINOPLE)                           0.0
IRAN                                            742604.0
IRAN-IRAQ                                        16000.0
IRAN; PAKISTAN                                      40.0
IRAQ                                             70200.0
IRAQ-SYRIA                                           0.0
IRELAND                                            100.0
ISRAEL                                          105100.0
ISRAEL; JORDAN                                     268.0
ITALY                                           422678.0
ITALY-BALKANS NW                                     0.0
IWATE, JAPAN                                         0.0
JAMAICA                                           4000.0
JAPAN                                           354138.0
JAPAN TRENCH                                         0.0
JAVA-S. JAVA SEA                                     0.0
JORDAN                                               0.0
KASHIMA, JAPAN                                       0.0
```

In [105]:

```
#1.2
import pandas as pd
Sig_Eqs
Sig_Eqs['Mag']=Sig_Eqs['Mag'].astype(float)
Ms=Sig_Eqs[Sig_Eqs['Mag']>3].groupby('Year')['Mag'].count()
Ms.plot()
#得到的趋势是每年检测到的震级大于三级的地震数目，在公元前前数目基本不增加，在大约300-1500年开始出现
#并在20世纪后的增加速度显著加快，这是因为在最开始地震是由经受地震的人所记载的，这时人类分布的范围小，
#随后因为人口增加，居住范围更广泛，遭受的地震更多，被记载的地震也就更多，到了20实际，地震开始由受灾
#都可以被检测到，所以也就开始出现检测数目激增的情况。
```

Out[105]:

```
<AxesSubplot:xlabel='Year'>
```



In [116]:

```
#1.3(没做出来)
Sig_Eqs['for_count']=1
Sig_Eqs.groupby('country')['for_count'].sum()
Sig_Eqs.groupby('country')['Mag'].max()
```

...

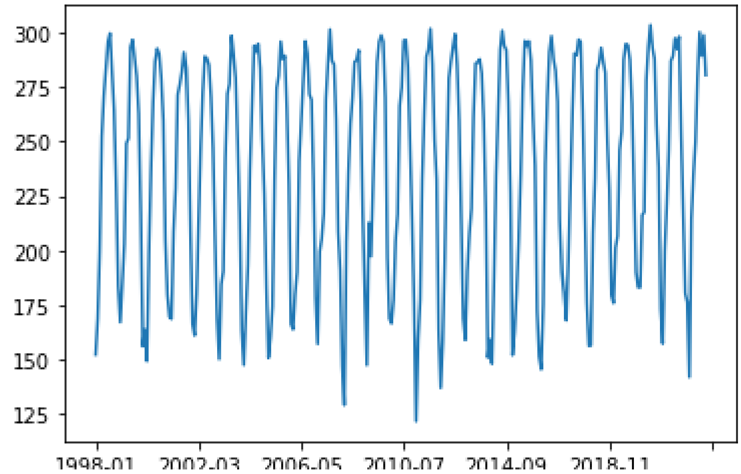
In [191]:

```
#第二题
#TMP的列中有两个数据，逗号前面的是气温数据，逗号后面的是对这个数字可信程度的判断，当逗号前的数据为+9
#为0和1时表明该数据真实可信，逗号后数字为2是表示该数据可能出现问题，逗号后数字为3时表示该数据是错误的
#而是来自NCEI数据库，逗号后数字为6表示该数据来自NCEI数据库且可能不对。在处理时首先应将逗号后数字为9和
met=pd.read_csv('Baoan_Weather_1998_2022.csv')
met.head()
#首先把TMP中的数据拆分成温度列和判断数字列，然后将其转化为int格式，后面用于删除数据
met['temperature']=met['TMP'].str.split(',').str.get(0)
met['define']=met['TMP'].str.split(',').str.get(1)
met['define']=met['define'].astype(int)
met['tem']=met['temperature'].str.split('+').str.get(1)
met['tem']=met['tem'].astype(int)
#对前面说的，判断数字为3和9的参数进行删除
met = met.drop(met[met['define']==9].index)
met = met.drop(met[met['define']==3].index)
#将时间序列中的年和月拆出来重组成为一列具有年和月信息的列
met['year']=met['DATE'].str.split('-').str.get(0)
met['month']=met['DATE'].str.split('-').str.get(1)
met["Date"]= met['year'] + '-' + met['month']
#根据拆出来的年和月求每个月的气温平均值
met2=met.groupby('Date')['tem'].mean()
met2.plot()
#图中每一个峰大致代表一年的数据，以峰脚（1月）和峰顶（7，8月份）来看，在这25年中，宝安的月平均气温，
```

```
es. Specify dtype option on import or set low_memory=False.
  met=pd.read_csv('Baoan_Weather_1998_2022.csv')
```

Out[191]:

```
<AxesSubplot:xlabel='Date'>
```



In [329]:

```
#第三题
df = pd.read_csv('ibtracs.ALL.list.v04r00.csv',
                 usecols=range(17),
                 skiprows=[1, 2],
                 parse_dates=['ISO_TIME'],
                 na_values=['NOT_NAMED', 'NAME'])
df.head()
#3.1没看懂题目啥意思，是要找到10个最大风速，然后把SID和NAME提取出来？还是说根据这两个因素分组后排序？
#再分组要干啥呢？
df2=df.sort_values('WMO_WIND',ascending=False)[0:10]
df3=df2.groupby('NAME')['SID'].count()
df3
```

```
C:\Users\86135\AppData\Local\Temp\ipykernel_39116\3376994072.py:2: DtypeWarning: Col
umns (5) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('ibtracs.ALL.list.v04r00.csv',
```

Out[329]:

```
NAME
BART       1
BOBBY      1
BONITA     1
CARINA     1
FABIO      1
RICK       1
ROXANNE    1
SONGDA     1
Name: SID, dtype: int64
```

In [333]:

```
3.2#麻烦助教至少让我知道这东西bug在哪
df = pd.read_csv('ibtracs.ALL.list.v04r00.csv',
                 usecols=range(17),
                 skiprows=[1, 2],
                 parse_dates=['ISO_TIME'],
                 na_values=['NOT_NAMED', 'NAME'])
df2=df.sort_values('WMO_WIND',ascending=False)[0:20]
df3 = df2.set_index('NAME')
df3['WMO_WIND'].plot()
```

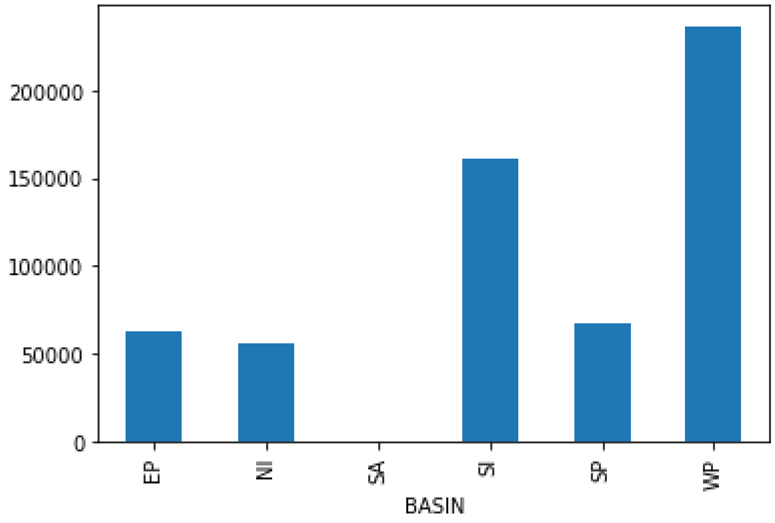
```
C:\Users\86135\AppData\Local\Temp\ipykernel_39116\2109543342.py:2: DtypeWarning: Col
umns (5) have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('ibtracs.ALL.list.v04r00.csv',

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call last)
Input In [333], in <cell line: 9>()
      7 df2=df.sort_values('WMO_WIND',ascending=False)[0:20]
      8 df3 = df2.set_index('NAME')
----> 9 df3['WMO_WIND'].plot()

File ~\anaconda3\lib\site-packages\pandas\plotting\_core.py:972, in PlotAcce
ssor.__call__(self, *args, **kwargs)
    969             label_name = label_kw or data.columns
    970             data.columns = label_name
--> 972 return plot_backend.plot(data, kind=kind, **kwargs)

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\__init__.p
y:71, in plot(data, kind, **kwargs)
     69         kwargs["ax"] = getattr(ax, "left_ax", ax)
     70 plot_obj = PLOT_CLASSES[kind](data, **kwargs)
---> 71 plot_obj.generate()
     72 plot_obj.draw()
     73 return plot_obj.result

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:327
, in MPLPlot.generate(self)
    325 def generate(self):
    326     self._args_adjust()
--> 327     self._compute_plot_data()
    328     self._setup_subplots()
    329     self._make_plot()

File ~\anaconda3\lib\site-packages\pandas\plotting\_matplotlib\core.py:506
, in MPLPlot._compute_plot_data(self)
    504 # no non-numeric frames or series allowed
    505 if is_empty:
--> 506     raise TypeError("no numeric data to plot")
    508 self.data = numeric_data.apply(self._convert_to_ndarray)

TypeError: no numeric data to plot
```

In [274]:

```
#3.3
df['number']=1
df4=df.groupby('BASIN')['number'].sum()
df4.plot(kind="bar")
```

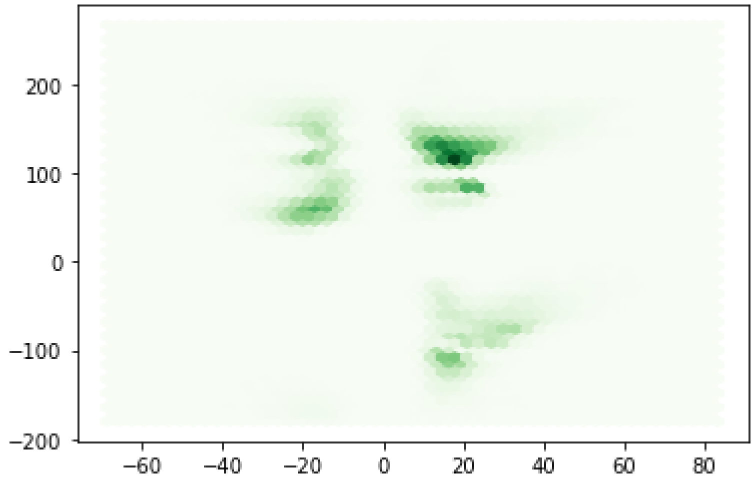
Out[274]:

```
<AxesSubplot:xlabel='BASIN'>
```



In [280]:

```
#3.4
x = df['LAT']
y = df['LON']
plt.hexbin(x, y, gridsize = 50, cmap ='Greens')
plt.show()
```

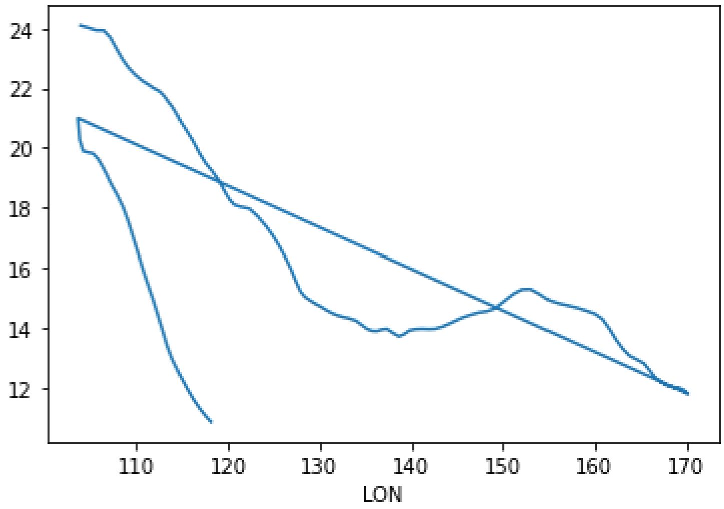


In [344]:

```
#3.5
df.head()
df2=df.loc[df['NAME']=='MANGKHUT']
df3 = df2.set_index('ISO_TIME')
df3=df2.set_index('LON')
df3['LAT'].plot()
```

Out[344]:

```
<AxesSubplot:xlabel='LON'>
```



In [315]:

```
#第四题:数据来自 Advanced Global Atmospheric Gases Experiment (AGAGE)网站
#4.1
my_data=pd.read_excel(’em-cfc——11.xls’)
my_data
my_data = my_data.drop(columns=[’Unnamed: 3’])
my_data = my_data.drop(columns=[’Unnamed: 7’])
my_data = my_data.drop(columns=[’Unnamed: 11’])
my_data = my_data.drop(columns=[’Unnamed: 15’])
my_data = my_data.drop(columns=[’Unnamed: 19’])
my_data
```

Out[315]:

	Table 1:	Production and Atmospheric Release	Unnamed: 2	Unnamed: 4	Unnamed: 5	Unnamed: 6	Expanded Data	Unnamed: 9	Unnar
0	CFC-11	(thousand metric tonnes)	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	NaN	Cumulative	NaN	NaN	NaN	NaN	
2	NaN	Annual	NaN	Total	NaN	NaN	Refrigeration hermetic	NaN	
3	NaN	Production	Released	Production	Released	Unreleased	Sales	Released	Unrele
4	1931	0	0	0	0	0	0	0	

In [ ]: