

1. Significant earthquakes since 2150 B.C.

1.1 在 Significant Earthquake Database 中下载 earthquakes-2025-10-29_21-06-00_+0800.tsv, 使用 `pandas.read_csv` 读取文件, 然后根据题目要求, 按国家分组求和并排序, 最后打印死亡人数前十名的国家及总数 (先做了一个 Nan 的填充, 防止出错, 将 Nan 值填充为 0)。

Result:

```
deaths_by_country = Sig_Eqs.fillna({'Total Deaths': 0})
top_10_deaths = deaths_by_country.groupby('Country')['Total Deaths'].sum().sort_values(ascending=False).head(10)

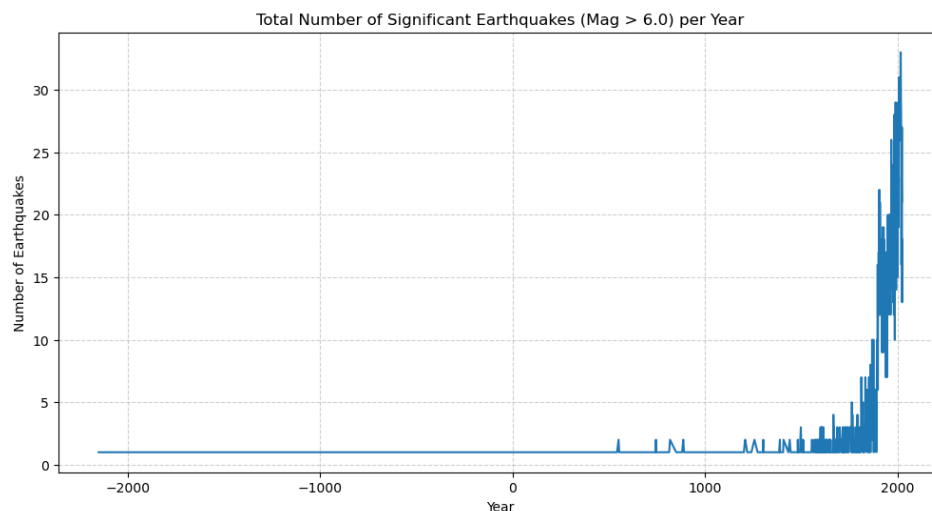
print(top_10_deaths)
```

| Country | Total Deaths |
|------------|--------------|
| CHINA | 2106524.0 |
| TURKEY | 1008863.0 |
| IRAN | 761654.0 |
| SYRIA | 487726.0 |
| ITALY | 423280.0 |
| HAITI | 323782.0 |
| JAPAN | 319443.0 |
| AZERBAIJAN | 319251.0 |
| INDONESIA | 282838.0 |
| ARMENIA | 189000.0 |

Name: Total Deaths, dtype: float64

1.2 检查 'Mag' (震级) 和 'Year' (年份) 两列, 删除有 Nan 的数据。筛选大于 6 级地震的数据, 存储在 `mag_gt_6` 变量中, 再进行分组并计数。

Result:



由图可得: 记录到的地震数量呈明显上升趋势, 尤其是在近一百年内。

但这并不代表地震发生的实际频率增加了, 可能存在以下情况:

监测技术: 全球地震监测台站的建立和技术的进步, 使得我们能检测到更多

地震。

记录保存：历史记录不完整，越久远的地震（尤其是发生在中等强度或偏远地区的）越不容易被记录下来。

1.3 通过布尔索引 `Sig_Eqs[Sig_Eqs['Country'] == country]`筛选出指定国家的所有地震记录，使用 `len()` 计算地震总数。最大震级数据查找：代码采用 `dropna(subset=['Mag'])`过滤掉缺失震级的记录，然后通过 `idxmax()`方法找到最大震级的索引，再用 `loc` 定位到具体的地震记录行。

Result:

| 国家 | 地震总数 | 最大地震日期 | 最大震级 |
|-------------|-------|------------|-------|
| ----- | ----- | ----- | ----- |
| CHINA | 623 | 1668-07-25 | 8.5 |
| JAPAN | 424 | 2011-03-11 | 9.1 |
| INDONESIA | 421 | 2004-12-26 | 9.1 |
| IRAN | 388 | 856-12-22 | 7.9 |
| TURKEY | 358 | 1939-12-26 | 7.8 |
| ITALY | 333 | 1915-01-13 | 7.5 |
| GREECE | 289 | 365-07-21 | 8.0 |
| USA | 280 | 1964-03-28 | 9.2 |
| PHILIPPINES | 230 | 1897-09-21 | 8.7 |
| MEXICO | 214 | 1787-03-28 | 8.6 |

2.Wind speed in Shenzhen from 2010 to 2020

第一步是将 WND 列拆分为独立的字段：风向、风向质量、风类型、风速和风速质量。

随后，应用了以下过滤步骤：

1. 质量码检查： 代码同时检查“风向质量码”（WND 列的第 2 个值）和“风速质量码”（第 5 个值）。一条记录只有在这两个质量码同时为可接受的代码（即 '0', '1', '4', '5', 或 '9' 之一）时才会被保留。任何一个字段的质量码如果为“可疑” ('2', '6') 或 “错误” ('3', '7')，该整条记录都将被丢弃。

2. 风类型码检查： 代码包含了一个 OR (或) 条件，用于处理指南中提到的特殊静风情况。如果一条记录的“风类型码”为 '9' 且其“风速”为 0 (来自 "0000")，那么即使它的质量码不符合标准（例如，风向质量码可能是 '9'），这条记录也会

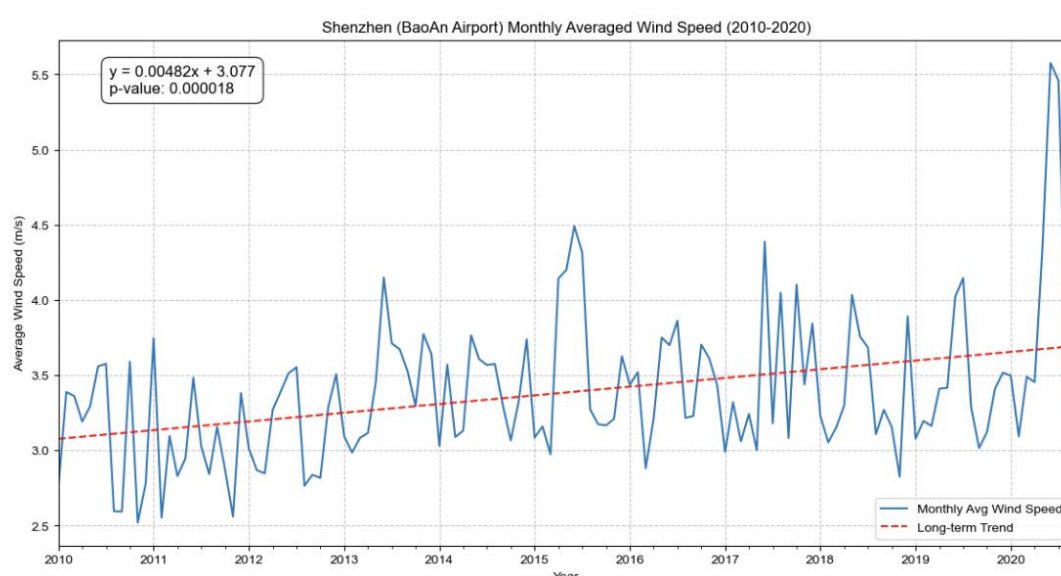
被视为有效的静风记录并保留下来。

3. 风向合理性检查： 此步骤已包含在步骤 1 的质量码检查中。根据指南，如果风类型为 'V' (Variable)，风向为 999 是合理的，此时其风向质量码应为 '1' 或 '5' 等有效代码，该记录会被保留。如果风类型不是 'V' 但风向为 999，这将被视为一个错误，其质量码（很可能是 '3' 或 '7'）会导致该行在步骤 1 中被丢弃。

4. 风速缺失值检查： 代码会检查原始风速值。所有风速为 9999（指南中定义的缺失标识）的记录，都会在转换为数字时被视为空值 (NaN)，并从分析中排除。

5. 数据转换和范围验证： 最后，原始风速文本（如 "0020"）被转换为数字。此转换过程（以及对 9999 的排除）确保了数据落在 0000-0900 的有效范围内。根据指南的缩放因子要求，所有有效的风速值都会除以 10，以得到米/秒 (m/s) 为单位的最终风速，用于后续的绘图和趋势计算。

Result:



尽管风速有强烈的季节性波动（每年冬高夏低），但在 2010 年至 2020 年期间，深圳宝安机场的月平均风速显示出一个显著的长期上升趋势。

3.Explore a data set

3.1 本题使用的数据是 Cooley 等基于 ICESat-2 激光测高任务得到的全球湖

泊水位数据（2018.10-2022.07），使用 read_csv 和 shape 读取并展示该 CSV 数据的行列数和各字段名称。

行数 (Rows): 227386
列数 (Columns): 27

| | id | Latitude | Longitude | Area | Type | Oct2018 | Nov2018 | \ | |
|---|------------|-----------|-------------|----------|-----------|------------|------------|-----------|---|
| 0 | 1 | 77.912613 | -23.634773 | 0.666206 | Natural | NaN | NaN | | |
| 1 | 2 | 77.892250 | -23.822008 | 0.108693 | Natural | NaN | 928.545349 | | |
| 2 | 3 | 77.868866 | -20.779461 | 0.315888 | Natural | NaN | NaN | | |
| 3 | 4 | 77.843330 | -110.950310 | 0.598154 | Natural | NaN | -3.412189 | | |
| 4 | 5 | 77.831947 | -20.891270 | 0.355333 | Natural | NaN | NaN | | |
| | | | | | | | | | |
| | | Dec2018 | Jan2019 | Feb2019 | Mar2019 | Apr2019 | May2019 | \ | |
| 0 | 775.671814 | NaN | NaN | NaN | NaN | NaN | NaN | | |
| 1 | | NaN | 928.693787 | NaN | NaN | NaN | 928.926147 | | |
| 2 | | NaN | NaN | NaN | NaN | 90.351400 | NaN | | |
| 3 | -3.328942 | NaN | -3.199217 | NaN | NaN | NaN | -3.154937 | | |
| 4 | | NaN | NaN | NaN | NaN | 171.377846 | 171.282623 | | |
| | | | | | | | | | |
| | | Jun2019 | Jul2019 | Aug2019 | Sep2019 | Oct2019 | Nov2019 | Dec2019 | \ |
| 0 | | NaN | NaN | NaN | NaN | NaN | 772.795502 | NaN | |
| 1 | | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| 2 | | NaN | 89.984301 | NaN | NaN | 89.826809 | NaN | NaN | |
| 3 | -3.167234 | NaN | NaN | NaN | -3.637697 | NaN | NaN | -3.448276 | |
| 4 | | NaN | 170.826790 | NaN | NaN | 171.002904 | 170.972336 | NaN | |
| | | | | | | | | | |
| | | Jan2020 | Feb2020 | Mar2020 | Apr2020 | May2020 | Jun2020 | Jul2020 | \ |
| 0 | | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |
| 1 | | NaN | 927.205017 | NaN | NaN | NaN | NaN | NaN | |
| 2 | | NaN | NaN | NaN | 90.550261 | NaN | NaN | NaN | |
| 3 | NaN | -3.323550 | -3.277608 | NaN | NaN | NaN | -3.353647 | NaN | |
| 4 | | NaN | NaN | NaN | NaN | NaN | NaN | NaN | |

由于原数据包含全球数十万个湖泊这里考虑展示超大湖泊，筛选出面积字段大于 10000 的数据（共筛选出 15 个湖泊）。

```
# 提出面积大于10000的数据
df_lakes = df[df['Area'] > 10000]
print(len(df_lakes))
print(df_lakes)
```

转为数字格式使用 isnull 读取缺失值，为保证后面序列图较为完整，考虑采用简单的插值进行填充（使用简单的线性插值，边缘的直接使用最近月份填充）

Result:

```
转换为数字格式后，所有月份列中总共有 25 个 NaN 值。
线性插值完成。
边缘填充完成。
插值和边缘填充后，所有月份列中剩余 0 个 NaN 值。

--- 3.1 (宽格式) 插值完成 ---
插值后数据预览 (df_cleaned):
      id  Latitude  Longitude      Area      Type  Oct2018  \
95540  95541  66.170349 -123.098793  22622.509073  Natural  142.182180
142792  142793  61.623337 -114.189430  26039.404469  Natural  126.583547
149420  149421  60.752140  31.949997  17116.098144  Reservoir  20.877350
191046  191047  53.086105 -98.453865  24827.936811  Reservoir  184.391208
193605  193606  52.548405  106.370277  31951.020835  Reservoir  421.951141

      Nov2018  Dec2018  Jan2019  Feb2019  Mar2019  \
95540  143.606663  142.518844  143.242034  143.835225  143.567022
142792  127.078623  127.541004  127.211344  127.163485  126.985538
149420  19.895657  19.942883  20.431718  21.033176  19.634016
191046  184.288714  184.278557  185.391982  187.286412  186.097685
193605  418.897403  419.534525  419.740347  417.782043  417.859833

      Apr2019  May2019  Jun2019  Jul2019  Aug2019  \
95540  143.540918  143.718724  142.948381  143.356049  143.763718
142792  126.825382  126.617694  127.043706  126.247117  127.421238
149420  21.487126  20.627504  20.313654  20.277207  20.247953
191046  184.908958  189.195905  185.168013  186.473242  187.778471
193605  420.148443  421.285627  417.424483  420.437762  423.451042

      Sep2019  Oct2019  Nov2019  Dec2019  Jan2020  \
95540  143.518135  142.839623  142.367170  143.590946  143.734313
142792  127.344134  127.108449  127.883572  128.722741  127.283182
149420  23.034639  20.076752  20.009856  19.534496  21.897986
191046  183.776223  184.775945  184.567415  185.127782  185.498128
193605  419.351915  420.813239  419.347108  418.628769  420.761242

      Feb2020  Mar2020  Apr2020  May2020  Jun2020  Jul2020
95540  143.383112  143.370858  144.823421  143.574838  143.143247  143.478931
142792  127.117013  127.262662  127.772008  128.684613  127.195734  127.195734
149420  21.590687  20.121696  22.742567  21.146817  22.509172  20.937564
191046  185.436180  185.711409  186.486258  187.740797  184.656055  186.586935
193605  417.469236  417.501212  418.777261  418.502194  418.323616  421.978538
```

3.2 该数据与多维数据不同，直接将各月水位数据作为列，所以除去前五列信息外后面为时间及其对应的水位。读取后面的列名转化为时间格式，挑选了前五个湖泊绘制它们的水位时间序列图。

Result:

