

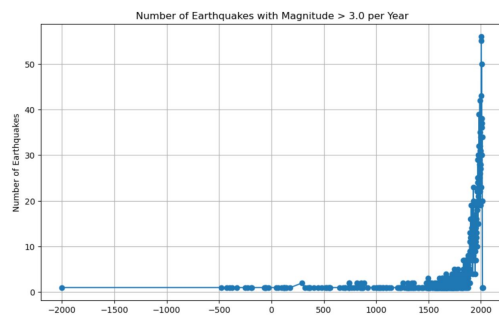
1.1

先有 `pandas` 库对这个文件进行处理，计算出每个国家死亡的总人数，再对其进行排序，找出前 20 个国家，并 `print` 出。

```
Country
CHINA      2075947.0
TURKEY     1188881.0
IRAN       1011453.0
ITALY      498418.0
SYRIA      439224.0
HAITI      323478.0
AZERBAIJAN 317219.0
JAPAN      279607.0
ARMENIA    191890.0
PAKISTAN   145083.0
IRAQ       136200.0
ECUADOR    135496.0
TURKMENISTAN 117412.0
PERU       102169.0
ISRAEL     90388.0
PORTUGAL   83572.0
GREECE     80378.0
CHILE      64277.0
INDIA      63507.0
TAIWAN     57153.0
Name: Deaths, dtype: float64
```

1.2

和 1.1 差不多的思路，筛选出地震震级 `Ms` 大于 3.0 的地震，按年进行分类，并对地震进行计数，最终绘制出相关时间序列图。



1.3

先定义如题目所要求的函数，计算出地震的总数，并找出最大的地震，并且得到最大的地震的地点（用经纬度表示）和时间（年，月，日，时，分，秒）。定义函数之后，剩下的思路 和 1.1 与 1.2 一致，但需要将得到的数据结构化，`Dataframe`，才可以采用降序的功能。

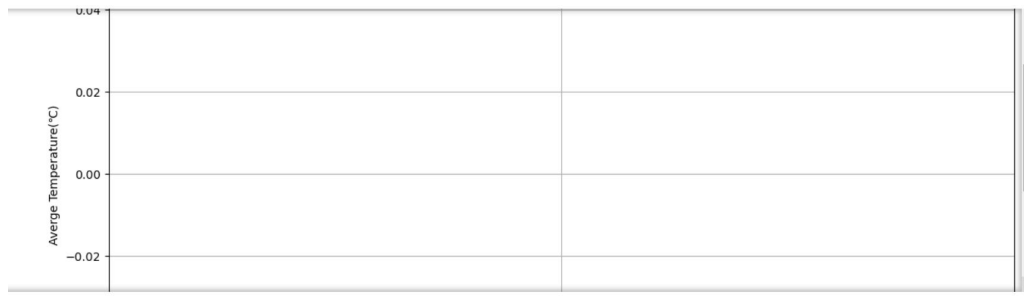
```
Total Earthquakes      Largest Earthquake Time \
15      623  (1920.0, 12.0, 16.0, 12.0, 5.0, 54.7)
34      419  (869.0, 7.0, 13.0, nan, nan, nan)
71      412  (2004.0, 12.0, 26.0, 0.0, 58.0, 53.5)
8       386  (856.0, 12.0, 22.0, nan, nan, nan)
10      337  (1939.0, 12.0, 26.0, 23.0, 57.0, 23.8)
..      ...
120      1  (None, None, None, None, None, None)
126      1  (1914.0, 10.0, 23.0, 6.0, 18.0, 34.0)
128      1  (None, None, None, None, None, None)
135      1  (1963.0, 2.0, 21.0, 17.0, 14.0, 31.0)
0       0  (None, None, None, None, None, None)

Largest Earthquake Location
15      (36.601, 105.317)
34      (38.5, 143.8)
71      (3.295, 95.982)
8       (36.2, 54.3)
10      (39.907, 39.586)
..      ...
120      (None, None)
126      (6.0, 132.5)
128      (None, None)
135      (32.6, 21.0)
0       (None, None)

[158 rows x 3 columns]
```

2.

读取 CSV 文件，将 TMP 列转换为数值的类型，根据指南中气温的缩放因子来看，应将 CSV 文件中的 TMP 数值除 10,在进行筛选，找到日期列，转为 `datetime` 的类型，计算月平均气温即可，再绘制出相关图。



3.1

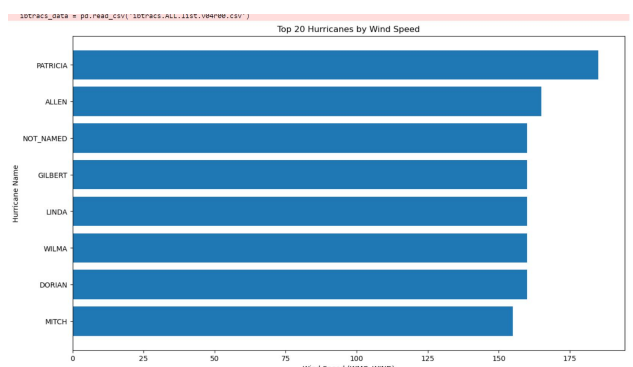
先读取 CSV 文件到。检查必要的列是否存在。将风速列转换为数值类型，并去除 `NaN` 值。根据 `SID` 分组，并计算每个飓风的最大风速。再选出风速最大的前 10 个飓风，合并名称 (`NAME`) 列，因为一个飓风可能有多个名称。根据最大风速降序排序。输出前 10 个飓风的名称和最大风速。

```
10tracs_data = pd.read_csv(
    NAME WMO_WIND
0 PATRICIA 185.0
10 PATRICIA 185.0
1 PATRICIA 185.0
18 PATRICIA 185.0
17 PATRICIA 185.0
...
358 MITCH 155.0
357 MITCH 155.0
356 MITCH 155.0
355 MITCH 155.0
458 RICK 155.0
```

[459 rows x 2 columns]

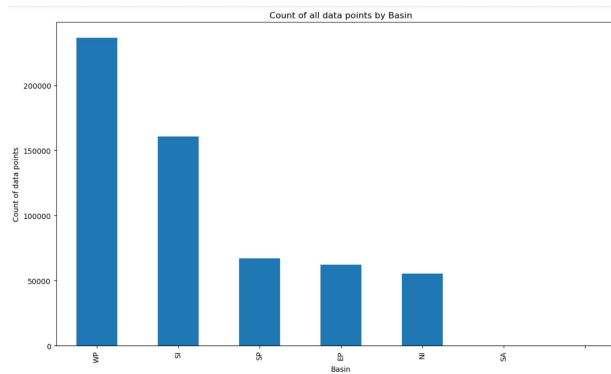
3.2

基本思想和 3.1 一致，根据风速进行降序排列，选出前 20 个，绘制条形图，展示 20 个飓风的风速和名称。



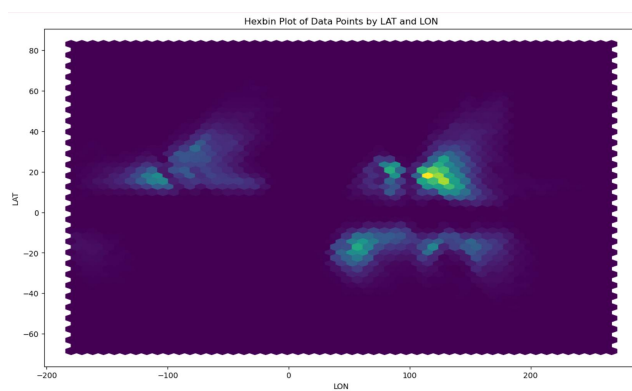
3.3

前面读取文件的步骤与 3.1 类似，用 `value_counts()` 的计数方法计算每个流域的数据点的数量，绘制条形图，设置好 x 轴，y 轴以及标题即可。



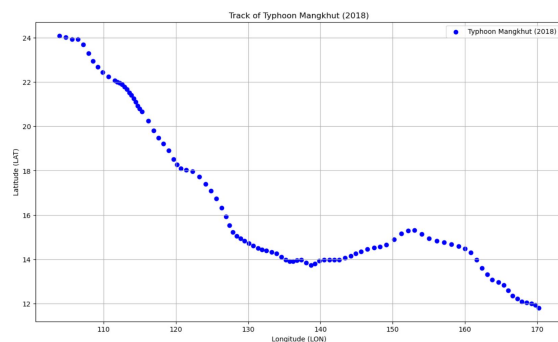
3.4

读取文件的步骤与 3.1 类似，将经纬度转为数值类型，在去除掉 NaN 值，利用 hexbin 函数画六边形分箱图，`gridsize` 用来控制六边形的大小，`cmap` 用来控制颜色的映射，设置好图表的标题，x 轴和 y 轴，显示图表即可。



3.5

读取文件，在文件中选出山竹的相关数据，其余操作与上面类似。



3.6

读取文件，将 `season` 转化为数值型，并去除 NaN 值，筛选出 1970 年以后以及 `wp` 和 `ep` 部分的数据，生成 `dataframe` 即可。

```

350395 1970050H07151 1970.0 22 WP MM NANCY
350396 1970050H07151 1970.0 22 WP MM NANCY
350397 1970050H07151 1970.0 22 WP MM NANCY
350398 1970050H07151 1970.0 22 WP MM NANCY
350399 1970050H07151 1970.0 22 WP MM NANCY
...
707086 2022275N10316 2022.0 76 EP MM JULIA
707087 2022275N10316 2022.0 76 EP MM JULIA
707175 2022286N15151 2022.0 80 WP MM NOT_NAMED
707176 2022286N15151 2022.0 80 WP MM NOT_NAMED
707177 2022286N15151 2022.0 80 WP MM NOT_NAMED

ISO_TIME NATURE LAT LON ... BOH_GUST_PER \
350395 1970-02-19 00:00:00 TS 7.00000 151.400 ...
350396 1970-02-19 03:00:00 TS 7.24752 151.285 ...
350397 1970-02-19 06:00:00 TS 7.50000 151.000 ...
350398 1970-02-19 09:00:00 TS 7.75747 150.772 ...
350399 1970-02-19 12:00:00 TS 8.00000 150.500 ...
...
707086 2022-10-10 15:00:00 TS 13.9957 -90.2940 ...
707087 2022-10-10 18:00:00 NR 14.5000 -91.0000 ...
707175 2022-10-12 12:00:00 NR 15.2000 151.300 ...
707176 2022-10-12 15:00:00 NR 15.0500 151.325 ...
707177 2022-10-12 18:00:00 NR 14.9000 151.350 ...

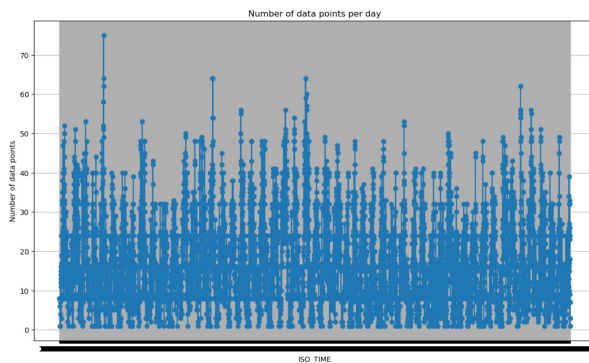
REUNION_GUST REUNION_GUST_PER USA_SEAHGT USA_SEARAD_NE USA_SEARAD_SE \
350395
350396
350397
350398
350399
...
707086
707087
707175
707176
707177

USA_SEARAD_SW USA_SEARAD_NW STORM_SPEED STORM_DIR
350395 6 322
350396 6 321
350397 7 320
350398 7 315
350399 7 308
...
707086 16 306
707087 17 306
707175 3 0

```

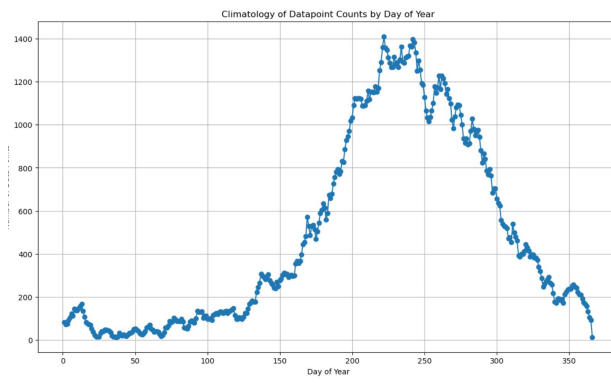
3.7

读取文件，找出文件中的 `ISO_TIME`，将其转换为 `datetime` 类型，并按照 `year`，`month`，`day` 进行划分，在计算每天的数量点的数量，设置图片 `x` 轴，`y` 轴，标题以及相关性质，最终出图即可。



3.8

思路与 3.7 类似，只是编程提取每年每天的数据点数量，其余的导入文件以及绘图步骤并没有明显变化。



3.9

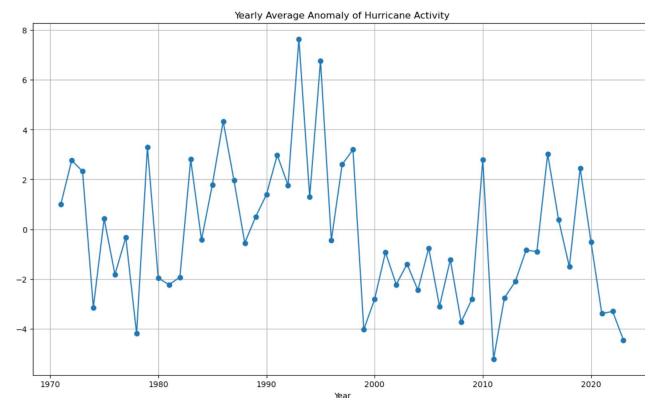
接着 3.8 的内容计算平均值和异常值即可。

```
      DayOfYear  Anomaly
0           50   1.500000
1           51   0.000000
2           52   0.833333
3           53   1.400000
4           54   2.600000
...          ...      ...
10812        277  -8.980769
10813        278 -11.693878
10814        282 -17.301887
10815        283 -12.769231
10816        285 -15.269231
```

[10817 rows x 2 columns]

3.10

导入文件，转换数据类型后，计算异常值的均值。设置年份为索引，并使用'YE'频率进行年度重采样（其实这个地方我没懂），再绘制每年的平均异常值图表。图表设置与之前相似



4.1

选择了一个 ccl4 相关文件。导入了。并清除掉了 9999 值

```
      time DD  MM  YYYY  hh  mm  mole  fraction  repeability  flag
12  1993.60693  10  8  1993  13  2  100.980  0.899          B
13  1993.60718  10  8  1993  14  22  101.540  0.892          B
14  1993.60730  10  8  1993  15  42  101.183  0.859          B
15  1993.60742  10  8  1993  17  2  100.250  0.824          B
16  1993.60754  10  8  1993  18  22  101.779  0.826          B
...          ...  ...  ...  ...  ...  ...          ...
329109 2023.49548  30  6  2023  20  51  72.939  0.176          B
329110 2023.49561  30  6  2023  21  31  72.983  0.176          B
329111 2023.49573  30  6  2023  22  11  72.847  0.175          B
329112 2023.49573  30  6  2023  22  51  72.703  0.170          B
329113 2023.49585  30  6  2023  23  31  72.748  0.171          B
```

[295528 rows x 9 columns]

4.2

和 3 中的操作一致，读取文件，确认列的存在，画图就好，但是我不知道为啥说 time 不存在，明明存在诶。。

4.3

计算 clean_data 中 mole 的平均数，中位数，标准差，最大值，最小值。

87.36715431363528 88.311 8.657385566976934 72.237 105.348