ESE5023 Assignment 01 Report
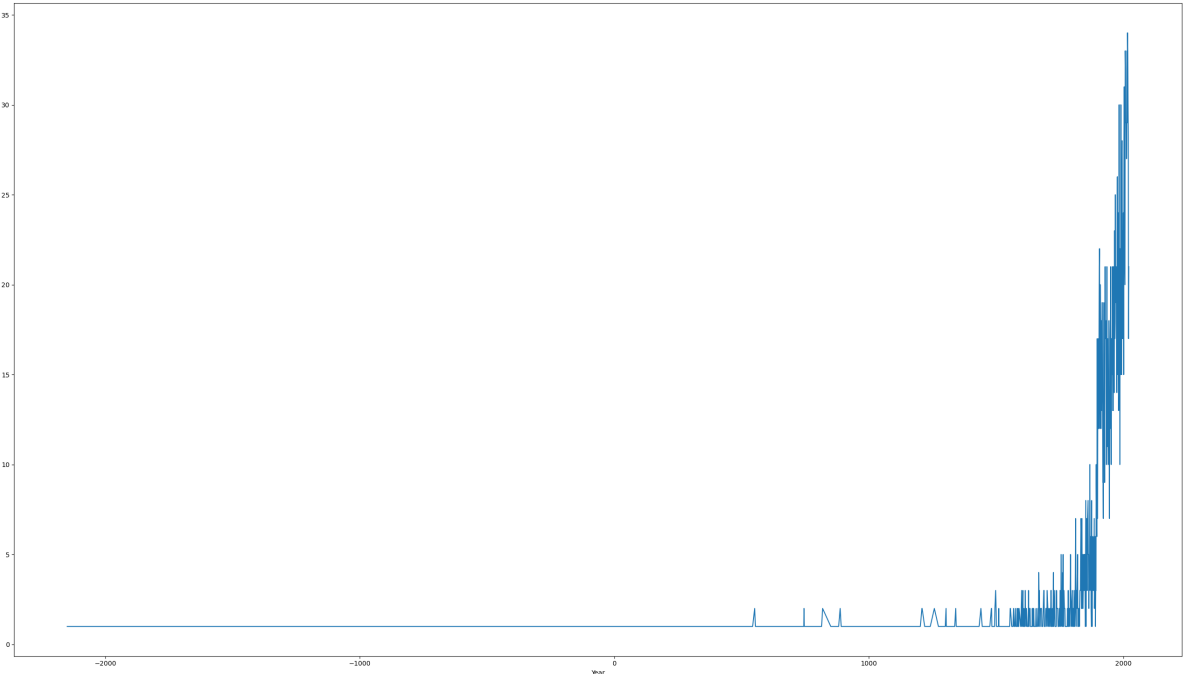
李骏垚 12132451

1. Significant earthquakes since 2150 B.C.
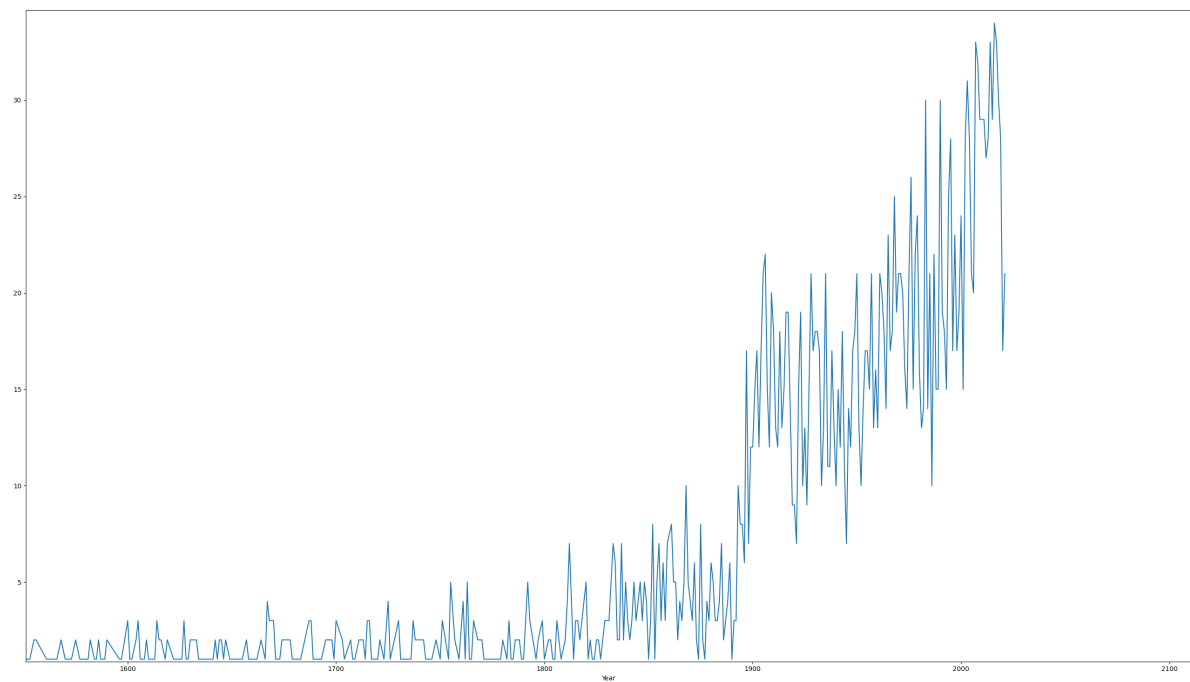
因地震而死亡的人数最高的十个国家如下：

Country	Deaths
CHINA	2074900.0
TURKEY	1074769.0
IRAN	1011437.0
SYRIA	439224.0
ITALY	434863.0
HAITI	323472.0
AZERBAIJAN	317219.0
JAPAN	278138.0
ARMENIA	191890.0
PAKISTAN	148764.0

每年地震震级≥6级的时间序列统计图如下：

放大观察1600-2020的数据：

可以发现有记录的地震次数在1900年附近发生了明显的增加，其原因应该是近代以来，人类观测与记录地震的手段和方法不断完善，因此地震记录次数明显地增加了。

按照题目要求对每个国家使用函数 CountEq_LargestEq() 后，将记录保存在文件 **mag_df.csv** 中

按降序记录每个国家地震发生总数以及最大震级地震发生的日期，绘制成如下表格：

Country	total_eqs	maxMag_date
14	CHINA	610
32	JAPAN	409
68	INDONESIA	399
7	IRAN	380
9	TURKEY	330
5	ITALY	326
51	USA	271
3	GREECE	269
65	PHILIPPINES	221
57	MEXICO	204
55	CHILE	198
48	PERU	185
15	RUSSIA	150
8	INDIA	99
72	TAIWAN	98
85	PAPUA NEW GUINEA	98
62	COLOMBIA	79
98	NEW ZEALAND	71
50	VENEZUELA	66
59	ECUADOR	64
115	SOLOMON ISLANDS	61
22	AFGHANISTAN	59
44	ALGERIA	57
16	ALBANIA	56
105	VANUATU	54
20	PAKISTAN	53
41	CROATIA	49
27	FRANCE	43
66	USA TERRITORY	40
71	NICARAGUA	39

Country	total_eqs	maxMag_date
63	EL SALVADOR	38
61	GUATEMALA	38
64	COSTA RICA	35
80	MYANMAR (BURMA)	33
1	SYRIA	33
37	SWITZERLAND	31
56	AZORES (PORTUGAL)	27
11	SPAIN	27
13	PORTUGAL	26
119	TAJIKISTAN	26
31	IRAQ	24
104	AUSTRALIA	24
4	ISRAEL	23
67	PANAMA	23
102	TONGA	22
30	SLOVENIA	22
107	NEW CALEDONIA	21
77	ARGENTINA	21
40	MOROCCO	20
73	CANADA	20
21	SOUTH KOREA	20
74	JAMAICA	19
109	FIJI	19
17	BULGARIA	18
52	DOMINICAN REPUBLIC	18
117	KERMADEC ISLANDS (NEW ZEALAND)	17
83	BANGLADESH	17
78	HAITI	17
36	ICELAND	17
39	NEPAL	16

Country	total_eqs	maxMag_date
25	AZERBAIJAN	16
18	GEORGIA	15
47	SERBIA	15
12	EGYPT	15
54	ROMANIA	15
6	LEBANON	14
10	KYRGYZSTAN	14
92	SOUTH AFRICA	14
35	UZBEKISTAN	14
75	CUBA	14
58	HONDURAS	13
34	UK	13
33	ARMENIA	13
23	MACEDONIA	12
2	TURKMENISTAN	11
79	MARTINIQUE	10
28	KAZAKHSTAN	10
38	YEMEN	10
45	BOSNIA-HERZEGOVINA	10
60	MONTENEGRO	9
53	GERMANY	9
97	GUADELOUPE	9
24	TUNISIA	9
69	ETHIOPIA	9
125	SAMOA	8
42	UKRAINE	8
121	TANZANIA	8
90	TRINIDAD AND TOBAGO	8
43	AUSTRIA	7
130	SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	7

Country	total_eqs	maxMag_date
94	CONGO	7
116	MONGOLIA	6
114	BOLIVIA	6
19	CYPRUS	6
29	NORTH KOREA	6
150	POLAND	6
132	BRAZIL	6
82	ATLANTIC OCEAN	6
81	ERITREA	6
127	VIETNAM	5
147	BHUTAN	5
134	ANTARCTICA	5
146	RWANDA	5
0	JORDAN	5
84	HUNGARY	5
70	GHANA	5
122	MICRONESIA FED. STATES OF	4
26	THAILAND	4
139	MALAWI	4
123	UGANDA	4
151	MOZAMBIQUE	3
149	SAUDI ARABIA	3
46	SLOVAKIA	3
142	NETHERLANDS	3
136	MALAYSIA	3
76	ANTIGUA AND BARBUDA	3
129	INDIAN OCEAN	3
128	KENYA	3
103	SOUTH SUDAN	3
88	TOGO	2

Country	total_eqs	maxMag_date
91	CANARY ISLANDS	2
145	LAOS	2
87	SAINT LUCIA	2
120	CAMEROON	2
86	FRENCH GUIANA	2
112	SOLOMON SEA	2
95	UK TERRITORY	2
131	PACIFIC OCEAN	2
108	COTE D'IVOIRE	2
89	SIERRA LEONE	1
99	BARBADOS	1
144	SUDAN	1
49	IRELAND	1
100	SAINT VINCENT AND THE GRENADINES	1
148	BURUNDI	1
110	SRI LANKA	1
106	BRITISH VIRGIN ISLANDS	1
152	CZECH REPUBLIC	1
153	MADAGASCAR	1
154	ZAMBIA	1
143	WALLIS AND FUTUNA (FRENCH TERRITORY)	1
137	BELGIUM	1
141	BERING SEA	1
140	DJIBOUTI	1
111	URUGUAY	1
138	GUINEA	1
135	GABON	1
113	MONTSERRAT	1
133	LIBYA	1
96	GRENADA	1

Country	total_eqs	maxMag_date
101	FRENCH POLYNESIA	1
93	NORWAY	1
126	CENTRAL AFRICAN REPUBLIC	1
124	PALAU	1
118	KIRIBATI	1
155	COMOROS	1

2. Wind speed in Shenzhen during the past 10 years

如果直接将csv文件读入会发生如下报错

```
sys:1: DtypeWarning: Columns (4,8,9,12,15,21,22,24,26,31,33,34) have mixed
types.Specify dtype option on import or set low_memory=False.
```

所以应该设置参数 low_memory=False，具体解释可见：[Pandas read_csv low_memory and dtype options](#)

```
Shenzhen_windspeed = pd.read_csv("./2281305.csv", low_memory=False)
```

通过阅读数据集使用说明，可以知道 csv 文件的最后一列 "WND" 即是我们需要的数据，其中 "WND" 又分为 5 列数据，使用 split() 函数可以将使用 ',' 分隔的数据进行分割，我们只需要使用最后两列数据即可，分别表示的意义如下：

POS: 66-69
WIND-OBSERVATION speed rate
The rate of horizontal travel of air past a fixed point.
MIN: 0000   MAX: 0900   UNITS: meters per second
SCALING FACTOR: 10
DOM: A general domain comprised of the numeric characters (0-9).
          9999 = Missing.

POS: 70-70
WIND-OBSERVATION speed quality code
The code that denotes a quality status of a reported WIND-OBSERVATION speed rate.
DOM: A specific domain comprised of the characters in the ASCII character set.
          0 = Passed gross limits check
          1 = Passed all quality control checks
          2 = Suspect
          3 = Erroneous
          4 = Passed gross limits check, data originate from an NCEI data source
          5 = Passed all quality control checks, data originate from an NCEI data source
          6 = Suspect, data originate from an NCEI data source
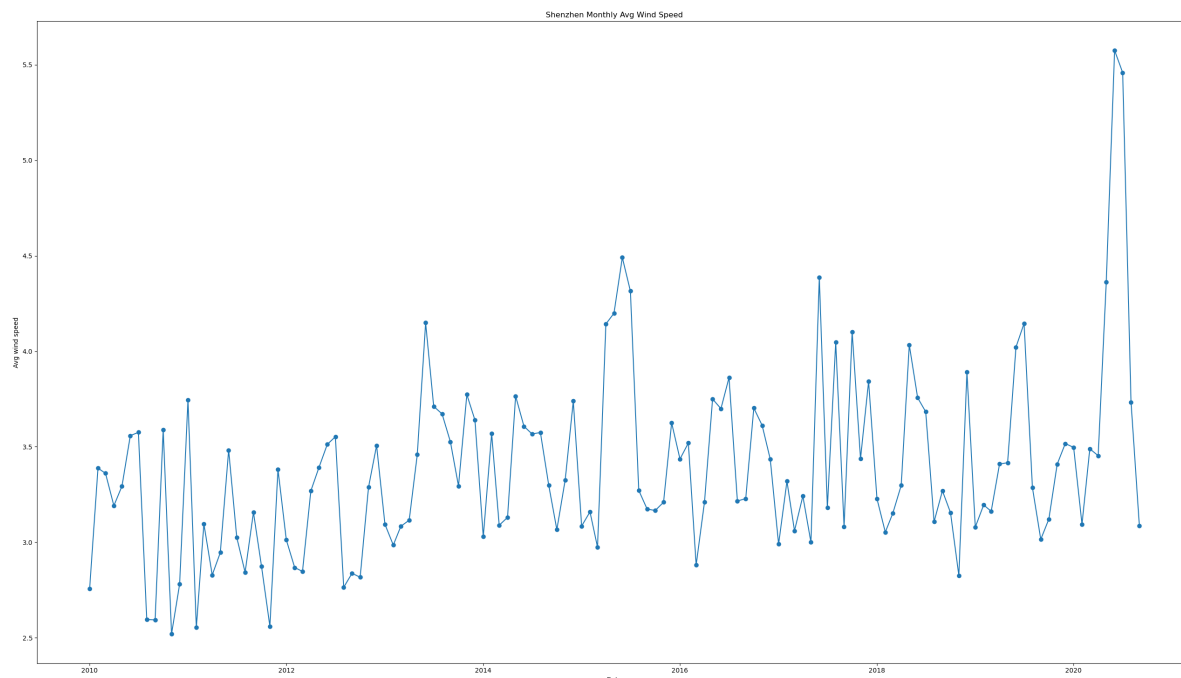   7 = Erroneous, data originate from an NCEI data source
          9 = Passed gross limits check if element is present

注意到 WIND-OBSERVATION speed rate 的 scale 是 10 ，所以使用的时候需要除以 10 以得到以 米/秒 为单位的数值。另外，对 quality code 做一个统计如下：

```
windspeed.groupby(['SQC']).size()

SQC
1    111345
5         1
9       638
dtype: int64
```

因此，需要对数据进行过滤，将 quality code 是 9 的数据丢弃即可，最后绘制出的图形如下：

可以观察到一个规律：每年年初以及年中（大概5-9月）的每月平均风速会较大一些

---

3. Explore a data set

这里使用 NOAA 提供的气象数据 Climate Data Online

选择 Climate Data Online 下的 Global Summary of the Year 数据集，该数据集由大量文件（约80000个）构成，所以首先需要进行数据的拼接工作。考虑到数据集特征，使用 outer 连接方法

在拼接过程中发现，如果简单地将两个 csv 文件读入的 DataFrame 进行 merge 操作，随着操作的进行，执行速度会越来越慢，对此现象的我的解释是，每次进行 merge 操作，计算机都需要将原来庞大的 DataFrame 销毁，再生成新的更加庞大的 DataFrame，这无疑是非常耗时的，所以我使用分治的算法思想，简单的将 merge 操作分为若干区块进行，大大加快了文件合并的速度

文件合并的代码在 PS2_3_preprocess.py 中，如果不想运行该代码（约耗时1h30min），直接使用合并后的文件 gsoy.csv 即可

gsoy.csv 文件体积较大，我放在了个人云盘中，连接Sustech校园网后点击链接就可以下载了
gsoy.csv 下载

由于该数据集十分庞大，首先需要筛选出少量的数据列用于简单的分析，提取少量数据列的代码如下：

```python
def extractDPData():
    df = pd.read_csv("./gsoy.csv", low_memory=False)

    dp_df = df[['STATION', 'DATE', 'LATITUDE', 'LONGITUDE', 'ELEVATION', 'NAME',
'DP01', 'DP01_ATTRIBUTES', 'DP10', 'DP10_ATTRIBUTES', 'DP1X', 'DP1X_ATTRIBUTES',
'EMXP', 'EMXP_ATTRIBUTES', 'PRCP', 'PRCP_ATTRIBUTES']]

    dp_df.to_csv('./dp_gsoy.csv')
```

提取完毕后，为了减少后续分析的运算量，便将结果直接保存在文件 **dp_gsoy.csv** 中

阅读数据集使用说明文档[Dataset Description Document Global Summary of the Month/Year Dataset](#)

我在本次 assignment 使用的数据是 DP01，DP10，DP1X 和 PRCP，其具体定义如下：

13. PRCP

Total Monthly (Annual) precipitation. Precipitation totals are based on daily or multi-day (if daily is missing) precipitation report, in millimeters to tenths.

The value is set to missing if more than 5 daily values are missing or flagged and there is an additional stipulation that there can be no more than 5 consecutive days of accumulation in a month (accumulations that cross a month are ignored, i.e., accumulated values are set to missing). This is to ensure consistency with the newest GHCN-Monthly data set (version 3).

15. DP01

Number of days with >= 0.01 inch/0.254 millimeter in the month (year). (Non-Accumulation) Note: values originally recorded in inches as 0.01" are stored as 0.3

millimeters in GHCN-Daily; technically this test is for values greater than or equal to 0.3 mm.

16. DP10

Number of days with >= 0.1 inch/2.54 millimeter in the month (year). (Non-Accumulation) Note: values originally recorded in inches as 0.10" are stored as 2.5 millimeters in GHCN-Daily; technically this test is for values greater than or equal to 2.5 mm.

17. DP1X

Number of days with >= 1.0 inch (25.4mm) precipitation in the month (year). (Non-Accumulation)
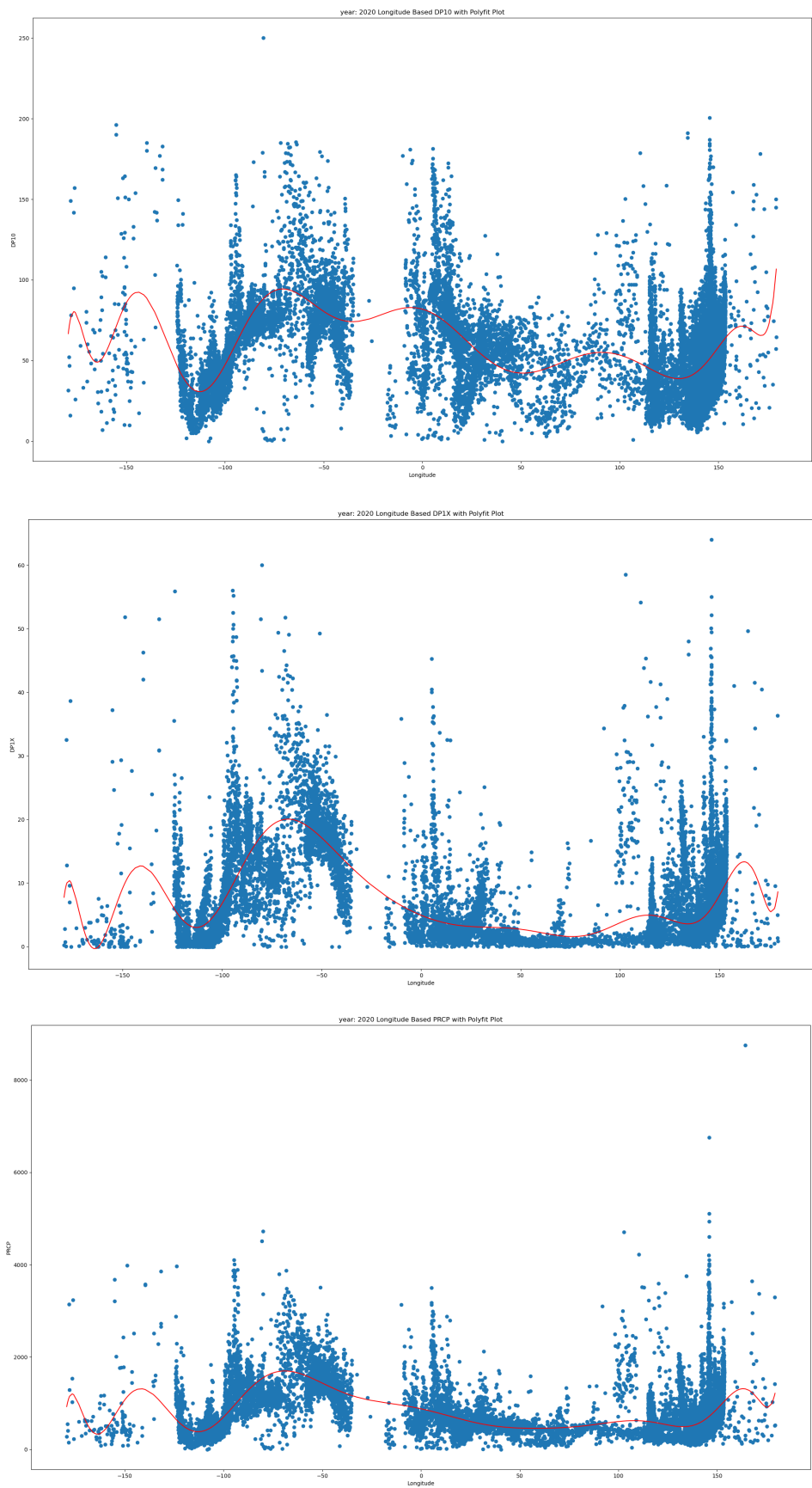
首先绘制站点 **ASN00077034** 的年均总降雨量折线图，这是关于时间序列的：
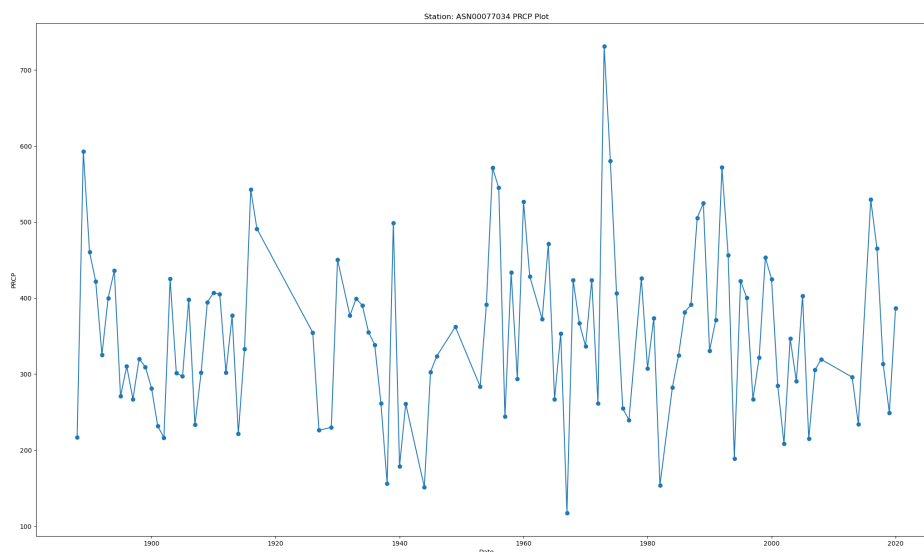
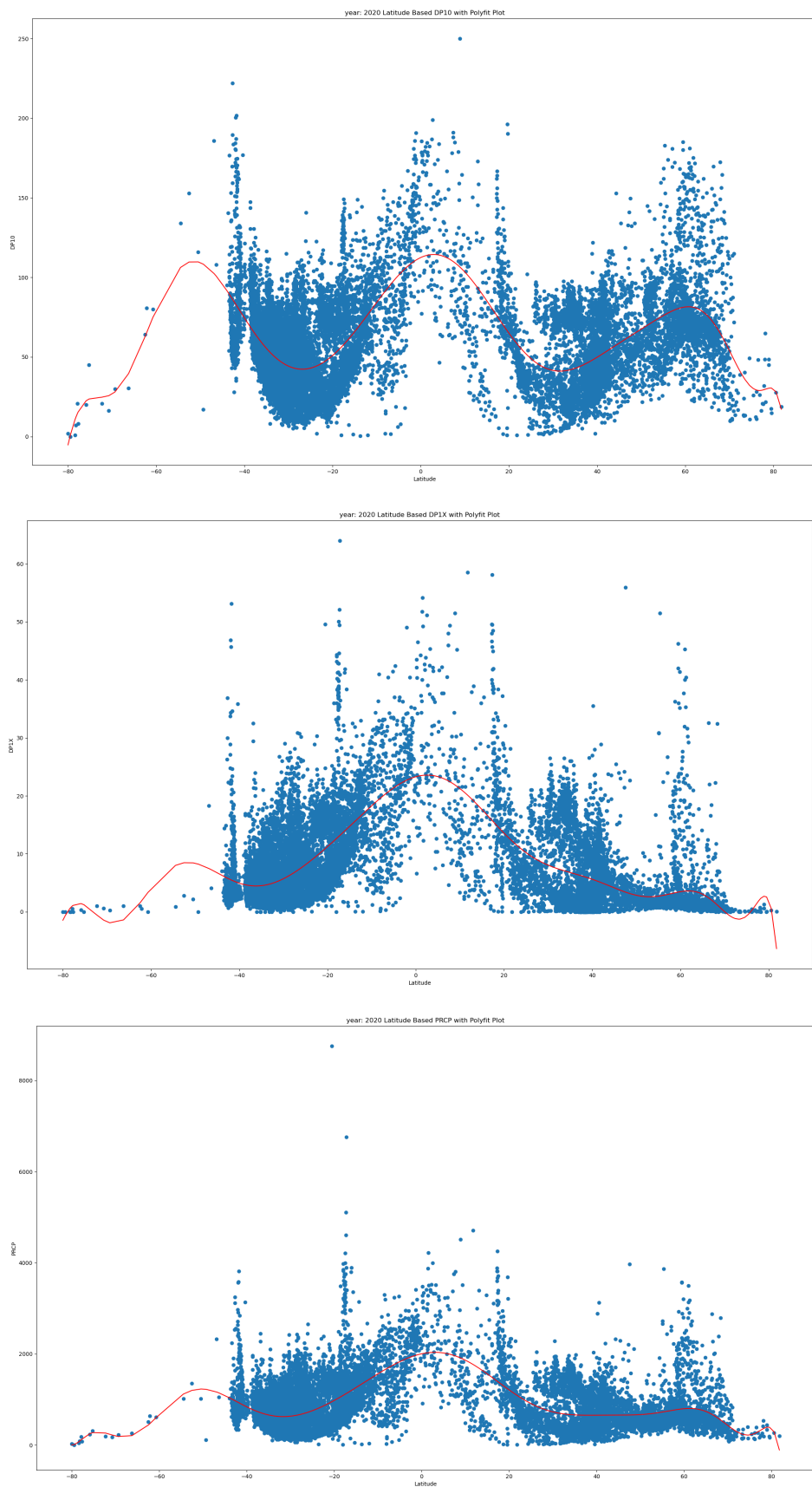接下来我挑选了 **2020** 这一年份，分别以**经度**，**纬度**为横坐标绘制 2020 年全球各地站点统计得到的 DP10，DP1X 和 PRCP

此外为了方便地观察数据趋势，我使用了 numpy 提供的多项式拟合函数（[numpy.polyfit()](#)），对数据进行了 **degree=15** 的多项式拟合

最后得到的结果如下：

首先是以经度为基准的：

year: 2020 Longitude Based DP10 with Polyfit Plot

year: 2020 Longitude Based DP1X with Polyfit Plot

year: 2020 Longitude Based PRCP with Polyfit Plot

首先是以纬度为基准的：

year: 2020 Latitude Based DP10 with Polyfit Plot

year: 2020 Latitude Based DP1X with Polyfit Plot

year: 2020 Latitude Based PRCP with Polyfit Plot

观察上面的图像可以得到几个简单的现象规律（2020年）：

1. 大约西经40°至西经80°的区域，无论降水总量还是有大量降雨的天数（DP1X）都明显高于其他地区，该区域恰好囊括了南美洲，而南美洲拥有世界上最大的雨林生态系统，这应该可以成为该现象的可行解释之一。

2. 从纬度来看热带和南北温带全年降雨量适中的天数（DP10）都较多，大致呈现 w 的形态。如果看全年降水总量的话，赤道附近地区的降水总量则明显多于其他地区。