In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import skew
import reliability as rel
import seaborn as sns
import math
```

1. Significant earthquakes since 2150 B.C.

In [2]:

```
Sig_Eqs = pd.read_csv("earthquakes-2023-11-06_10-35-19_+0800.tsv", sep='\t')
Sig_Eqs
```

Out[2]:

	Search Parameters	Id	Year	Mo	...	Total Houses Destroyed	Total Houses Destroyed Description	Total Houses Damaged	Total Houses Damaged Description
0	[]	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
1	NaN	1	-2150	NaN	...	NaN	NaN	NaN	NaN
2	NaN	2	-2000	NaN	...	NaN	NaN	NaN	NaN
3	NaN	3	-2000	NaN	...	NaN	1	NaN	NaN
4	NaN	5877	-1610	NaN	...	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...
6394	NaN	10708	2023	10	...	2862	4	250	3
6395	NaN	10711	2023	10	...	200	3	NaN	2
6396	NaN	10709	2023	10	...	NaN	NaN	NaN	NaN
6397	NaN	10710	2023	10	...	NaN	NaN	NaN	NaN
6398	NaN	10712	2023	10	...	NaN	2	NaN	NaN

6399 rows × 49 columns

1.1 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.

Total number of deaths:

In [3]:

```
Sig_Eqs.groupby('Country').sum().sort_values('Total Deaths')[['Total Deaths']]
```

Out[3]:

Country	Total Deaths
ZAMBIA	0
COTE D'IVOIRE	0
SRI LANKA	0
SOLOMON SEA	0
FRENCH GUIANA	0
FRENCH POLYNESIA	0
GABON	0
GERMANY	0
SLOVAKIA	0
GRENADA	0
SIERRA LEONE	0
SAUDI ARABIA	0
COMOROS	0

Country	Total Deaths
SAINT VINCENT AND THE GRENADINES	0
INDIAN OCEAN	0
PALAU	0
PACIFIC OCEAN	0
NORWAY	0
NORTH KOREA	0
JORDAN	0
KERMADEC ISLANDS (NEW ZEALAND)	0
KIRIBATI	0
MONTSERRAT	0
LAOS	0
LEBANON	0
HUNGARY	0
TOGO	0
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	0
MICRONESIA, FED. STATES OF	0
VIETNAM	0
CANARY ISLANDS	0
ANTARCTICA	0
ANTIGUA AND BARBUDA	0
URUGUAY	0
ATLANTIC OCEAN	0
CAMEROON	0
CENTRAL AFRICAN REPUBLIC	0
BERING SEA	0
BRITISH VIRGIN ISLANDS	0
KENYA	1
NETHERLANDS	1
AUSTRIA	1
TONGA	1
THAILAND	1
MADAGASCAR	2
SUDAN	2
CZECH REPUBLIC	2
TRINIDAD AND TOBAGO	2
BELGIUM	2
BURUNDI	3
BRAZIL	3
UK TERRITORY	3
MOZAMBIQUE	4
WALLIS AND FUTUNA (FRENCH TERRITORY)	5
BOSNIA-HERZEGOVINA	6
DJIBOUTI	6
PANAMA	7

Country	Total Deaths
SERBIA	7
FIJI	7
BHUTAN	11
ICELAND	11
UKRAINE	11
VANUATU	11
HONDURAS	12
AUSTRALIA	12
MALAWI	13
MALAYSIA	19
SLOVENIA	23
GHANA	25
NEW CALEDONIA	25
TANZANIA	28
POLAND	28
MONGOLIA	30
SOUTH SUDAN	31
CANADA	32
CYPRUS	42
CUBA	54
SOUTH AFRICA	55
CONGO	80
IRELAND	100
ERITREA	106
SWITZERLAND	109
BOLIVIA	111
MONTENEGRO	131
BULGARIA	138
SOUTH KOREA	151
KYRGYZSTAN	153
COSTA RICA	169
UGANDA	171
USA TERRITORY	172
SOLOMON ISLANDS	179
SAMOA	194
ETHIOPIA	200
GEORGIA	285
LIBYA	300
MARTINIQUE	391
GUINEA	443
KAZAKHSTAN	452
NEW ZEALAND	474
FRANCE	659
SAINT LUCIA	900

Country	Total Deaths
RWANDA	1045
USA	1083
MYANMAR (BURMA)	1083
MACEDONIA	1083
UK	1400
ROMANIA	1701
DOMINICAN REPUBLIC	1873
BANGLADESH	2136
PAPUA NEW GUINEA	2577
ALBANIA	2826
BARBADOS	3000
EL SALVADOR	3766
JAMAICA	4000
YEMEN	4071
UZBEKISTAN	4894
GUADELOUPE	5007
CROATIA	5014
SPAIN	5495
AZORES (PORTUGAL)	6353
COLOMBIA	6617
NICARAGUA	10180
RUSSIA	12081
AFGHANISTAN	13033
ARGENTINA	14520
PHILIPPINES	14750
TAJIKISTAN	15970
MEXICO	18683
NEPAL	21464
MOROCCO	22775
ISRAEL	35368
GUATEMALA	36189
ALGERIA	39339
EGYPT	41706
VENEZUELA	43981
TUNISIA	48013
TAIWAN	57705
INDIA	62396
CHILE	70174
IRAQ	70200
GREECE	81215
PORTUGAL	82572
PERU	96161
TURKMENISTAN	110412
ECUADOR	134444

	Total Deaths
Country	
PAKISTAN	143712
ARMENIA	189000
INDONESIA	282819
AZERBAIJAN	310119
HAITI	323776
JAPAN	356083
ITALY	422679
SYRIA	437700
IRAN	758650
TURKEY	995648
CHINA	2.04193e+06

Top 10:

In [4]:

```
Sig_Eqs.groupby('Country').sum().sort_values('Total Deaths', ascending=False)[['Total Deaths']].head(10)
```

Out[4]:

	Total Deaths
Country	
CHINA	2.04193e+06
TURKEY	995648
IRAN	758650
SYRIA	437700
ITALY	422679
JAPAN	356083
HAITI	323776
AZERBAIJAN	310119
INDONESIA	282819
ARMENIA	189000

1.2 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?

Total number of earthquakes > 6.0 each year:

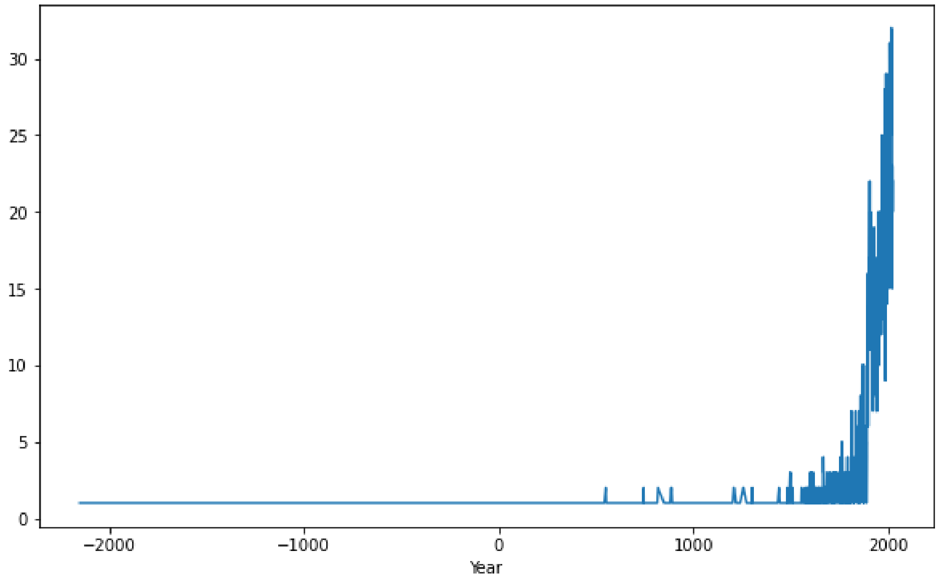
In [5]:

```
print(
    Sig_Eqs.loc[Sig_Eqs['Mag']>6.0].groupby(['Year']).count()['Mag']
)
```

```
Year
-2150     1
-2000     1
-1250     1
-1050     1
-479      1
         ..
2019     27
2020     15
2021     23
2022     20
2023     22
Name: Mag, Length: 536, dtype: int64
```

In [7]:

```
Sig_Eqs.loc[Sig_Eqs['Mag']>6.0].groupby(['Year']).count()['Mag'].plot()
plt.gcf().set_size_inches(10,6)
plt.show()
```



In the past 1,000 years, earthquakes have occurred with increasing frequency.

The reason of the increasing earthquakes may be due to modern technology and improved monitoring leading to more recorded earthquakes.

1.3 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.

In [8]:

```
#All records by each country
Eqs_c = Sig_Eqs.groupby('Country').count()
Eqs_c
```

Out[8]:

	Search Parameters	Id	Year	Mo	...	Total Houses Destroyed	Total Houses Destroyed Description	Total Houses Damaged	Total Houses Damaged Description
Country									
AFGHANISTAN	0	66	66	65	...	15	23	16	20
ALBANIA	0	56	56	48	...	6	17	3	4
ALGERIA	0	57	57	57	...	12	25	1	8
ANTARCTICA	0	5	5	5	...	0	0	0	0
ANTIGUA AND BARBUDA	0	3	3	3	...	0	0	0	0
ARGENTINA	0	21	21	21	...	3	7	1	2
ARMENIA	0	13	13	9	...	0	3	0	0
ATLANTIC OCEAN	0	6	6	6	...	0	0	0	0
AUSTRALIA	0	24	24	24	...	0	0	1	2
AUSTRIA	0	7	7	7	...	0	0	0	0
AZERBAIJAN	0	16	16	14	...	2	4	3	5
AZORES (PORTUGAL)	0	27	27	27	...	2	8	0	4
BANGLADESH	0	17	17	17	...	2	3	0	3
BARBADOS	0	1	1	1	...	0	0	0	0
BELGIUM	0	1	1	1	...	0	1	0	0

Country	Search Parameters	Id	Year	Mo	...	Total Houses Destroyed	Total Houses Destroyed Description	Total Houses Damaged	Total Houses Damaged Description
BERING SEA	0	1	1	1	...	0	0	0	0
BHUTAN	0	5	5	5	...	2	3	2	3
BOLIVIA	0	6	6	6	...	0	1	0	0
BOSNIA-HERZEGOVINA	0	11	11	10	...	0	0	1	1
BRAZIL	0	8	8	8	...	2	2	1	2
BRITISH VIRGIN ISLANDS	0	1	1	1	...	0	0	0	0
BULGARIA	0	18	18	16	...	4	8	2	2
BURUNDI	0	1	1	1	...	1	1	0	0
CAMEROON	0	2	2	2	...	0	0	0	0
CANADA	0	20	20	19	...	1	1	0	2
CANARY ISLANDS	0	2	2	1	...	0	0	0	0
CENTRAL AFRICAN REPUBLIC	0	1	1	1	...	0	0	0	0
CHILE	0	198	198	195	...	13	55	7	18
CHINA	0	620	620	601	...	206	438	60	76
COLOMBIA	0	80	80	79	...	15	33	8	24
COMOROS	0	1	1	1	...	0	0	0	1
CONGO	0	7	7	6	...	2	4	2	3
COSTA RICA	0	36	36	35	...	4	9	5	6
COTE D'IVOIRE	0	2	2	1	...	0	0	0	0
CROATIA	0	53	53	48	...	3	5	2	4
CUBA	0	14	14	12	...	1	2	1	2
CYPRUS	0	7	7	4	...	2	3	1	1
CZECH REPUBLIC	0	1	1	1	...	0	0	0	0
DJIBOUTI	0	1	1	1	...	0	0	0	0
DOMINICAN REPUBLIC	0	18	18	14	...	0	3	0	2
ECUADOR	0	68	68	67	...	6	20	9	17
EGYPT	0	15	15	13	...	4	7	1	3
EL SALVADOR	0	38	38	35	...	5	11	8	11
ERITREA	0	6	6	6	...	0	2	0	0
ETHIOPIA	0	9	9	9	...	0	2	0	1
FIJI	0	19	19	19	...	0	1	0	1
FRANCE	0	43	43	38	...	0	2	2	4
FRENCH GUIANA	0	2	2	1	...	0	1	0	1
FRENCH POLYNESIA	0	1	1	1	...	0	0	0	0
GABON	0	1	1	1	...	0	0	0	0
GEORGIA	0	15	15	12	...	2	3	2	3
GERMANY	0	9	9	7	...	0	0	1	1
GHANA	0	5	5	5	...	1	3	0	0
GREECE	0	270	270	242	...	36	89	17	39
GRENADA	0	1	1	1	...	0	0	0	0
GUADELOUPE	0	9	9	9	...	0	2	1	2
GUATEMALA	0	39	39	36	...	7	19	6	11
GUINEA	0	1	1	1	...	1	1	0	0

Country	Search Parameters	Id	Year	Mo	...	Total Houses Destroyed	Total Houses Destroyed Description	Total Houses Damaged	Total Houses Damaged Description
HAITI	0	20	20	20	...	4	7	4	6
HONDURAS	0	13	13	13	...	3	7	4	4
HUNGARY	0	5	5	5	...	0	1	0	0
ICELAND	0	17	17	12	...	2	4	1	1
INDIA	0	100	100	97	...	10	33	11	32
INDIAN OCEAN	0	3	3	3	...	0	0	0	0
INDONESIA	0	411	411	408	...	90	157	82	125
IRAN	0	384	384	340	...	50	107	24	62
IRAQ	0	24	24	20	...	3	5	0	0
IRELAND	0	1	1	0	...	0	0	0	0
ISRAEL	0	24	24	10	...	0	4	0	0
ITALY	0	331	331	299	...	13	55	4	18
JAMAICA	0	19	19	17	...	0	1	0	1
JAPAN	0	414	414	411	...	101	120	30	37
JORDAN	0	5	5	3	...	0	0	0	2
KAZAKHSTAN	0	10	10	9	...	3	3	0	1
KENYA	0	3	3	3	...	0	0	0	1
KERMADEC ISLANDS (NEW ZEALAND)	0	21	21	21	...	0	0	0	0
KIRIBATI	0	1	1	1	...	0	0	0	0
KYRGYZSTAN	0	14	14	12	...	6	7	3	3
LAOS	0	2	2	2	...	0	0	1	1
LEBANON	0	14	14	6	...	0	0	0	1
LIBYA	0	1	1	1	...	0	1	0	0
MACEDONIA	0	12	12	9	...	1	2	0	2
MADAGASCAR	0	1	1	1	...	0	0	0	1
MALAWI	0	4	4	4	...	1	1	2	2
MALAYSIA	0	3	3	3	...	0	0	1	1
MARTINIQUE	0	10	10	10	...	0	1	0	0
MEXICO	0	209	209	200	...	9	45	13	38
MICRONESIA, FED. STATES OF	0	4	4	4	...	0	0	0	0
MONGOLIA	0	6	6	6	...	0	0	0	0
MONTENEGRO	0	10	10	10	...	0	1	0	2
MONTSERRAT	0	1	1	1	...	0	1	0	0
MOROCCO	0	21	21	13	...	2	4	1	2
MOZAMBIQUE	0	3	3	3	...	3	3	1	2
MYANMAR (BURMA)	0	34	34	31	...	2	7	5	10
NEPAL	0	19	19	19	...	4	12	3	5
NETHERLANDS	0	3	3	3	...	0	0	1	2
NEW CALEDONIA	0	25	25	25	...	0	1	0	1
NEW ZEALAND	0	71	71	69	...	4	12	3	12
NICARAGUA	0	39	39	37	...	6	9	4	6
NORTH KOREA	0	6	6	5	...	0	1	0	0

Country	Search Parameters	Id	Year	Mo	...	Total Houses Destroyed	Total Houses Destroyed Description	Total Houses Damaged	Total Houses Damaged Description
NORWAY	0	1	1	1	...	0	0	0	0
PACIFIC OCEAN	0	2	2	2	...	0	0	0	0
PAKISTAN	0	53	53	52	...	13	25	9	17
PALAU	0	1	1	1	...	0	0	0	0
PANAMA	0	23	23	23	...	0	5	3	6
PAPUA NEW GUINEA	0	101	101	99	...	10	34	2	13
PERU	0	190	190	181	...	32	67	20	37
PHILIPPINES	0	224	224	216	...	30	76	25	58
POLAND	0	7	7	7	...	0	0	0	0
PORTUGAL	0	28	28	23	...	5	15	0	2
ROMANIA	0	15	15	15	...	0	0	3	6
RUSSIA	0	152	152	150	...	2	11	7	10
RWANDA	0	5	5	5	...	3	3	1	2
SAINT LUCIA	0	2	2	2	...	0	0	0	0
SAINT VINCENT AND THE GRENADINES	0	1	1	1	...	0	0	0	0
SAMOA	0	8	8	8	...	0	2	0	0
SAUDI ARABIA	0	3	3	3	...	0	1	0	0
SERBIA	0	15	15	15	...	1	3	2	2
SIERRA LEONE	0	1	1	1	...	0	0	0	0
SLOVAKIA	0	3	3	3	...	0	0	0	0
SLOVENIA	0	22	22	18	...	0	4	0	1
SOLOMON ISLANDS	0	62	62	62	...	7	12	4	6
SOLOMON SEA	0	2	2	2	...	0	1	0	0
SOUTH AFRICA	0	14	14	14	...	1	2	1	2
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	0	7	7	7	...	0	0	0	0
SOUTH KOREA	0	21	21	14	...	2	10	3	3
SOUTH SUDAN	0	3	3	3	...	0	0	0	2
SPAIN	0	34	34	30	...	4	12	3	7
SRI LANKA	0	1	1	1	...	0	0	0	0
SUDAN	0	1	1	1	...	0	0	0	0
SWITZERLAND	0	31	31	28	...	0	2	0	2
SYRIA	0	33	33	14	...	0	6	0	0
TAIWAN	0	100	100	99	...	40	58	3	12
TAJIKISTAN	0	27	27	27	...	7	14	6	6
TANZANIA	0	8	8	8	...	4	5	3	4
THAILAND	0	4	4	4	...	2	2	0	1
TOGO	0	2	2	1	...	0	0	0	0
TONGA	0	24	24	23	...	0	0	0	0
TRINIDAD AND TOBAGO	0	8	8	7	...	2	3	0	1
TUNISIA	0	9	9	6	...	0	2	0	0
TURKEY	0	335	335	282	...	40	80	26	45

Country	Search Parameters	Id	Year	Mo	...	Total Houses Destroyed	Total Houses Destroyed Description	Total Houses Damaged	Total Houses Damaged Description
TURKMENISTAN	0	11	11	7	...	0	3	0	0
UGANDA	0	4	4	4	...	0	2	0	0
UK	0	14	14	10	...	1	2	0	3
UK TERRITORY	0	2	2	2	...	0	0	0	0
UKRAINE	0	12	12	11	...	0	3	0	4
URUGUAY	0	1	1	1	...	0	0	0	0
USA	0	276	276	274	...	17	32	13	37
USA TERRITORY	0	40	40	38	...	2	7	1	4
UZBEKISTAN	0	14	14	12	...	1	2	0	0
VANUATU	0	54	54	54	...	0	5	1	7
VENEZUELA	0	66	66	61	...	2	9	7	12
VIETNAM	0	5	5	5	...	0	0	0	3
WALLIS AND FUTUNA (FRENCH TERRITORY)	0	1	1	1	...	0	1	0	1
YEMEN	0	10	10	5	...	3	5	2	2
ZAMBIA	0	1	1	1	...	0	1	0	0

156 rows × 48 columns

In [9]:

```
#Get the counts of earthquake
def Eqs_count(country):
    return Eqs_c.loc['%s'%(country)]['Year']
```

In [10]:

```
def CountEq_LargestEq(country):
    Eqs_largest = Sig_Eqs[['Country','Mag']].groupby('Country').max()['Mag'][country]


    if pd.isna(Eqs_largest) :
        df1 = Sig_Eqs[(Sig_Eqs['Country']==country)]

    else:
        df1 = Sig_Eqs[(Sig_Eqs['Mag']==Eqs_largest) & (Sig_Eqs['Country']==country)]


    # Output
    df2 = df1[['Year','Mo','Dy','Country','Mag']].groupby('Country').max()


    df2['Earthquake_Number'] = Eqs_count(country)

    return df2
```

In [11]:

```
Country_List = Sig_Eqs.groupby('Country').count().index

result = CountEq_LargestEq(Country_List[0])

for i in range(1,156):
    result = pd.concat([result,CountEq_LargestEq(Country_List[i])])

# Show dataframe
result.sort_values('Mag',ascending = False)
```

Out[11]:

Country	Year	Mo	Dy	Mag	Earthquake_Number
CHILE	1960	5	22	9.5	198

Country	Year	Mo	Dy	Mag	Earthquake_Number
USA	1964	3	28	9.2	276
INDONESIA	2004	12	26	9.1	411
JAPAN	2011	3	11	9.1	414
RUSSIA	1952	11	4	9	152
PERU	1716	2	6	8.8	190
ECUADOR	1906	1	31	8.8	68
PHILIPPINES	1897	9	21	8.7	224
INDIA	1950	8	15	8.6	100
CHINA	1668	7	25	8.5	620
PORTUGAL	1761	11	30	8.5	28
MONGOLIA	1905	7	9	8.4	6
MEXICO	1899	1	24	8.4	209
KAZAKHSTAN	1889	7	11	8.3	10
PANAMA	1882	9	7	8.3	23
GUADELOUPE	1843	2	8	8.3	9
FIJI	1919	1	1	8.3	19
SAMOA	1917	6	26	8.3	8
COLOMBIA	1826	6	18	8.2	80
BOLIVIA	1994	6	9	8.2	6
TURKMENISTAN	1895	7	8	8.2	11
VENEZUELA	1894	4	29	8.2	66
TAIWAN	1920	6	5	8.2	100
NEPAL	1505	6	6	8.2	19
PAPUA NEW GUINEA	1919	5	6	8.2	101
AUSTRALIA	1989	5	23	8.2	24
INDIAN OCEAN	1928	3	9	8.1	3
HAITI	1842	5	7	8.1	20
KERMADEC ISLANDS (NEW ZEALAND)	2021	10	20	8.1	21
TONGA	1919	4	30	8.1	24
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	2021	8	27	8.1	7
AFGHANISTAN	1909	7	7	8.1	66
SOLOMON ISLANDS	2007	4	21	8.1	62
VANUATU	1913	10	14	8.1	54
CANADA	1949	8	22	8.1	20
ANTARCTICA	1998	3	25	8.1	5
USA TERRITORY	1902	9	22	8.1	40
PAKISTAN	1945	11	27	8	53
MYANMAR (BURMA)	1912	5	23	8	34
NEW CALEDONIA	1875	3	28	8	25
NEW ZEALAND	1855	1	23	8	71
ANTIGUA AND BARBUDA	1690	4	16	8	3
FRANCE	1817	3	11	8	43
GREECE	1303	8	21	8	270
GUATEMALA	1942	8	6	7.9	39

	Year	Mo	Dy	Mag	Earthquake_Number
Country					
MARTINIQUE	1906	12	3	7.9	10
NICARAGUA	1898	4	29	7.9	39
IRAN	856	12	22	7.9	384
EL SALVADOR	1915	9	7	7.9	38
DOMINICAN REPUBLIC	1946	8	4	7.9	18
SOUTH AFRICA	1942	11	10	7.9	14
JAMAICA	1899	6	14	7.8	19
TURKEY	2023	12	26	7.8	335
ATLANTIC OCEAN	1975	11	26	7.8	6
ARGENTINA	1944	1	15	7.8	21
BULGARIA	1904	4	4	7.8	18
COSTA RICA	1950	10	5	7.7	36
MICRONESIA, FED. STATES OF	1911	8	16	7.7	4
CUBA	2020	1	28	7.7	14
BANGLADESH	1918	7	8	7.6	17
SYRIA	1202	5	20	7.6	33
BRAZIL	1963	11	9	7.6	8
KIRIBATI	1905	6	30	7.6	1
AZORES (PORTUGAL)	1968	2	28	7.6	27
UK TERRITORY	1983	11	30	7.6	2
PALAU	1914	10	23	7.6	1
KYRGYZSTAN	1946	11	2	7.6	14
PACIFIC OCEAN	1932	11	2	7.5	2
SPAIN	881	5	26	7.5	34
ROMANIA	1977	3	4	7.5	15
ALBANIA	1893	6	14	7.5	56
HONDURAS	2018	8	10	7.5	13
ICELAND	1912	5	6	7.5	17
ITALY	1915	1	13	7.5	331
GEORGIA	1905	10	21	7.5	15
TRINIDAD AND TOBAGO	1888	1	10	7.5	8
TAJIKISTAN	1949	10	21	7.4	27
LEBANON	551	7	9	7.3	14
TANZANIA	1910	12	13	7.3	8
SOLOMON SEA	1895	3	6	7.3	2
JORDAN	-2150	NaN	NaN	7.3	5
EGYPT	1995	11	22	7.2	15
CROATIA	1667	4	6	7.2	53
ALGERIA	1980	10	10	7.1	57
SOUTH SUDAN	1990	5	20	7.1	3
UZBEKISTAN	1984	5	19	7	14
CONGO	1992	9	11	7	7
UKRAINE	1650	4	19	7	12
ISRAEL	1546	9	18	7	24

	Year	Mo	Dy	Mag	Earthquake_Number
Country					
MOZAMBIQUE	2006	2	22	7	3
AZERBAIJAN	1902	11	13	6.9	16
KENYA	1928	1	6	6.9	3
FRENCH GUIANA	1885	8	4	6.9	2
MONTENEGRO	1979	4	15	6.9	10
HUNGARY	1834	10	15	6.8	5
UGANDA	1912	7	9	6.8	4
ARMENIA	1988	12	7	6.8	13
VIETNAM	1935	11	1	6.8	5
MOROCCO	2023	9	8	6.8	21
NORTH KOREA	1518	7	2	6.7	6
BERING SEA	1991	2	21	6.7	1
AUSTRIA	1590	9	15	6.6	7
GHANA	1862	7	10	6.5	5
FRENCH POLYNESIA	1848	7	12	6.5	1
SOUTH KOREA	1700	12	25	6.5	21
ETHIOPIA	1906	8	25	6.5	9
SLOVENIA	1511	3	26	6.5	22
IRAQ	1864	12	2	6.4	24
WALLIS AND FUTUNA (FRENCH TERRITORY)	1993	3	12	6.4	1
BOSNIA-HERZEGOVINA	1969	10	27	6.4	11
LAOS	2007	5	16	6.3	2
CYPRUS	1953	9	10	6.3	7
DJIBOUTI	1989	8	20	6.3	1
GUINEA	1983	12	22	6.2	1
GABON	1974	9	23	6.2	1
CAMEROON	1945	9	12	6.2	2
MALAYSIA	1976	7	26	6.2	3
UK	1580	4	6	6.2	14
SWITZERLAND	1601	9	18	6.2	31
MACEDONIA	1979	5	24	6.2	12
ERITREA	1915	11	23	6.2	6
BHUTAN	2009	9	21	6.1	5
MALAWI	1989	3	10	6.1	4
THAILAND	2014	5	5	6.1	4
SERBIA	1922	3	24	6	15
YEMEN	1982	12	13	6	10
COMOROS	2018	5	15	5.9	1
ZAMBIA	2017	2	24	5.9	1
NORWAY	1819	8	31	5.8	1
RWANDA	2015	8	7	5.8	5
COTE D'IVOIRE	1879	2	11	5.7	2
SAUDI ARABIA	2009	5	19	5.7	3
TUNISIA	1957	2	20	5.6	9

Country	Year	Mo	Dy	Mag	Earthquake_Number
TOGO	1788	NaN	NaN	5.6	2
MADAGASCAR	2017	1	11	5.5	1
SUDAN	1993	8	1	5.5	1
LIBYA	1963	2	21	5.4	1
GERMANY	1978	9	3	5.3	9
SIERRA LEONE	1795	5	20	5.2	1
NETHERLANDS	1992	4	13	5.2	3
BELGIUM	1983	11	8	5	1
CENTRAL AFRICAN REPUBLIC	1921	9	16	4.8	1
POLAND	2004	9	21	4.8	7
BURUNDI	2004	2	24	4.7	1
CZECH REPUBLIC	2008	11	22	4.1	1
SLOVAKIA	2004	1	10	2.2	3
BARBADOS	1831	8	11	NaN	1
BRITISH VIRGIN ISLANDS	1871	9	NaN	NaN	1
CANARY ISLANDS	1810	3	20	NaN	2
GRENADA	1822	12	1	NaN	1
IRELAND	1490	NaN	NaN	NaN	1
MONTSERRAT	1897	4	25	NaN	1
SAINT LUCIA	1906	10	16	NaN	2
SAINT VINCENT AND THE GRENADINES	1844	8	30	NaN	1
SRI LANKA	1882	1	NaN	NaN	1
URUGUAY	1888	6	5	NaN	1

2. Wind speed in Shenzhen during the past 10 years

In [12]:

```
file = pd.read_csv('2281305.csv')
file
```

```
/Users/xujiayu/opt/anaconda3/envs/research/lib/python3.9/site-packages/IPython/core/interactiveshell.py:3444: 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.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Out[12]:

	STATION	DATE	SOURCE	REPORT_TYPE	...	SOURCE.1	TMP	VIS	WND
0	59493099999	2010-01-02T00:00:00	4	SY-MT	...	4	+0161,1	004000,1,N,1	040,1,N,0020,1
1	59493099999	2010-01-02T01:00:00	4	FM-15	...	4	+0170,1	002600,1,N,1	999,9,V,0010,1
2	59493099999	2010-01-02T02:00:00	4	FM-15	...	4	+0180,1	002600,1,N,1	999,9,C,0000,1
3	59493099999	2010-01-02T03:00:00	4	SY-MT	...	4	+0192,1	005000,1,N,1	140,1,N,0010,1
4	59493099999	2010-01-02T04:00:00	4	FM-15	...	4	+0180,1	002100,1,N,1	300,1,N,0040,1
...	...	...	...	...	...	...	...	...	...
111979	59493099999	2020-09-11T17:00:00	4	FM-15	...	4	+0290,1	009999,1,9,9	170,1,N,0030,1
111980	59493099999	2020-09-11T18:00:00	4	FM-15	...	4	+0290,1	009999,1,9,9	180,1,N,0040,1
111981	59493099999	2020-09-11T19:00:00	4	FM-15	...	4	+0290,1	009999,1,9,9	220,1,V,0030,1
111982	59493099999	2020-09-11T20:00:00	4	FM-15	...	4	+0290,1	009999,1,9,9	260,1,N,0030,1
111983	59493099999	2020-09-11T21:00:00	4	FM-15	...	4	+0290,1	009999,1,9,9	310,1,V,0020,1

111984 rows × 43 columns

In [13]:

```
def quality(ws):
    return ws[-1]
```

In [14]:

```
#get wind speed

def cal_Speed(ws):
    data = int(ws[8])*100 + int(ws[9])*10 + int(ws[10]) + int(ws[11])*0.1
    return data
```

In [15]:

```
#Get the month from original file

def newMonth(DATE):
    return DATE.split('T')[0].split('-')[1]

def newDate(DATE):
    return DATE.split('T')[0].split('-')[0] + '-' + DATE.split('T')[0].split('-')[1]
```

In [16]:

```
# Create a new column to store Temperature number.
file['WS'] = file.apply(lambda col: cal_Speed(col['WND']), axis=1)
```

In [17]:

```
file
```

Out[17]:

	STATION	DATE	SOURCE	REPORT_TYPE	...	TMP	VIS	WND	WS
0	59493099999	2010-01-02T00:00:00	4	SY-MT	...	+0161,1	004000,1,N,1	040,1,N,0020,1	2
1	59493099999	2010-01-02T01:00:00	4	FM-15	...	+0170,1	002600,1,N,1	999,9,V,0010,1	1
2	59493099999	2010-01-02T02:00:00	4	FM-15	...	+0180,1	002600,1,N,1	999,9,C,0000,1	0
3	59493099999	2010-01-02T03:00:00	4	SY-MT	...	+0192,1	005000,1,N,1	140,1,N,0010,1	1
4	59493099999	2010-01-02T04:00:00	4	FM-15	...	+0180,1	002100,1,N,1	300,1,N,0040,1	4
...	...	...	...	...	...	...	...	...	...
111979	59493099999	2020-09-11T17:00:00	4	FM-15	...	+0290,1	009999,1,9,9	170,1,N,0030,1	3
111980	59493099999	2020-09-11T18:00:00	4	FM-15	...	+0290,1	009999,1,9,9	180,1,N,0040,1	4
111981	59493099999	2020-09-11T19:00:00	4	FM-15	...	+0290,1	009999,1,9,9	220,1,V,0030,1	3
111982	59493099999	2020-09-11T20:00:00	4	FM-15	...	+0290,1	009999,1,9,9	260,1,N,0030,1	3
111983	59493099999	2020-09-11T21:00:00	4	FM-15	...	+0290,1	009999,1,9,9	310,1,V,0020,1	2

111984 rows × 44 columns

In [18]:

```
file['Q'] = file.apply(lambda col: quality(col['WND']), axis=1)

file
```

Out[18]:

	STATION	DATE	SOURCE	REPORT_TYPE	...	VIS	WND	WS	Q
0	59493099999	2010-01-02T00:00:00	4	SY-MT	...	004000,1,N,1	040,1,N,0020,1	2	1
1	59493099999	2010-01-02T01:00:00	4	FM-15	...	002600,1,N,1	999,9,V,0010,1	1	1
2	59493099999	2010-01-02T02:00:00	4	FM-15	...	002600,1,N,1	999,9,C,0000,1	0	1
3	59493099999	2010-01-02T03:00:00	4	SY-MT	...	005000,1,N,1	140,1,N,0010,1	1	1
4	59493099999	2010-01-02T04:00:00	4	FM-15	...	002100,1,N,1	300,1,N,0040,1	4	1
...	...	...	...	...	...	...	...	...	...
111979	59493099999	2020-09-11T17:00:00	4	FM-15	...	009999,1,9,9	170,1,N,0030,1	3	1
111980	59493099999	2020-09-11T18:00:00	4	FM-15	...	009999,1,9,9	180,1,N,0040,1	4	1
111981	59493099999	2020-09-11T19:00:00	4	FM-15	...	009999,1,9,9	220,1,V,0030,1	3	1
111982	59493099999	2020-09-11T20:00:00	4	FM-15	...	009999,1,9,9	260,1,N,0030,1	3	1
111983	59493099999	2020-09-11T21:00:00	4	FM-15	...	009999,1,9,9	310,1,V,0020,1	2	1

111984 rows × 45 columns

In [19]:
```
#Check the quality of wind speed

file2 = file.drop(index=file[file['Q']==2&3&6&7].index)
```

In [20]:
```
file2['Month'] = file2.apply(lambda col: newMonth(col['DATE']),axis=1)


file2['newDate'] = file2.apply(lambda col: newDate(col['DATE']),axis=1)
```

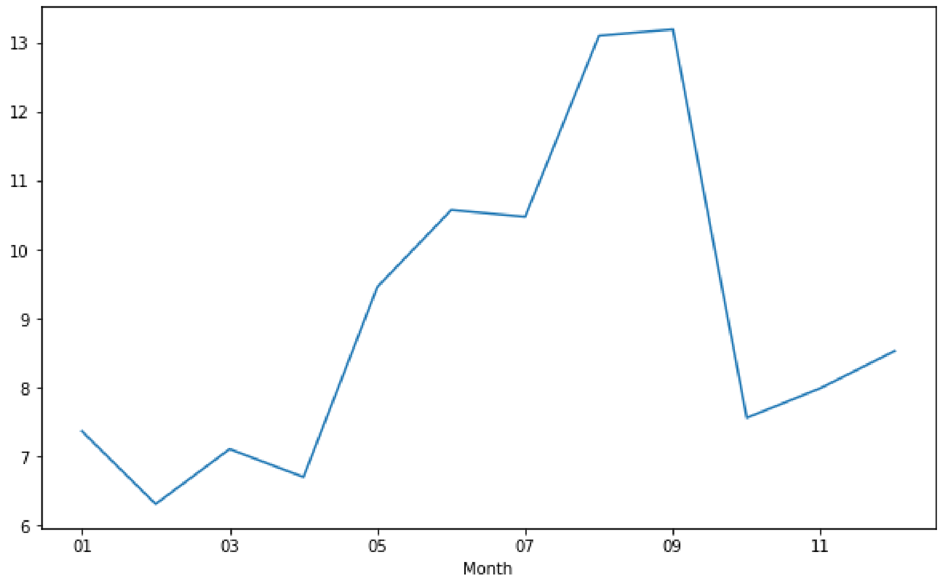
In [21]:
```
file2
```

Out[21]:

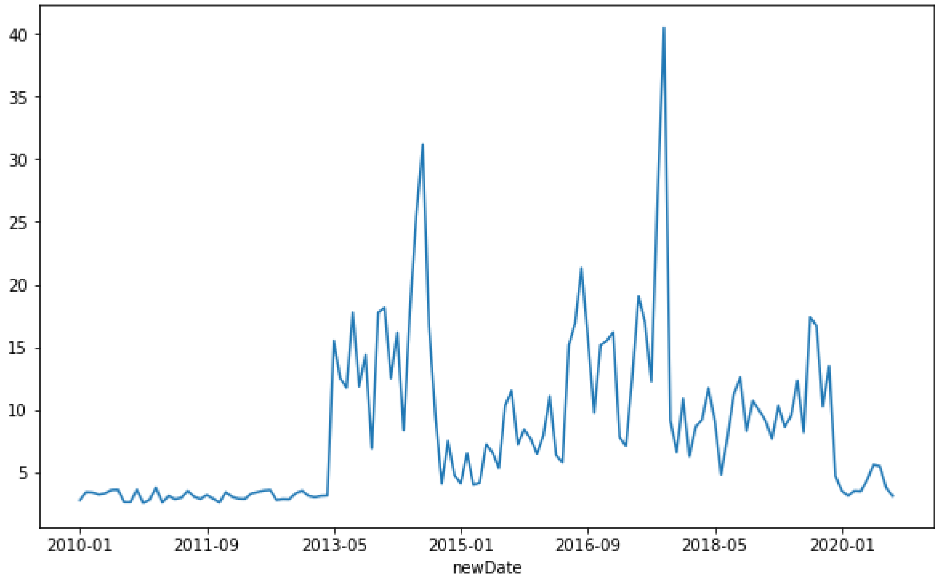
	STATION	DATE	SOURCE	REPORT_TYPE	...	WS	Q	Month	newDate
0	59493099999	2010-01-02T00:00:00	4	SY-MT	...	2	1	01	2010-01
1	59493099999	2010-01-02T01:00:00	4	FM-15	...	1	1	01	2010-01
2	59493099999	2010-01-02T02:00:00	4	FM-15	...	0	1	01	2010-01
3	59493099999	2010-01-02T03:00:00	4	SY-MT	...	1	1	01	2010-01
4	59493099999	2010-01-02T04:00:00	4	FM-15	...	4	1	01	2010-01
...	...	...	...	...	...	...	...	...	...
111979	59493099999	2020-09-11T17:00:00	4	FM-15	...	3	1	09	2020-09
111980	59493099999	2020-09-11T18:00:00	4	FM-15	...	4	1	09	2020-09
111981	59493099999	2020-09-11T19:00:00	4	FM-15	...	3	1	09	2020-09
111982	59493099999	2020-09-11T20:00:00	4	FM-15	...	3	1	09	2020-09
111983	59493099999	2020-09-11T21:00:00	4	FM-15	...	2	1	09	2020-09

111984 rows × 47 columns

In [22]:
```
#plot the monthly averaged in the last decades
file2.groupby('Month').mean()['WS'].plot()
plt.gcf().set_size_inches(10,6)
plt.show()
```



In [23]:
```
#plot the monthly averaged wind speed by time series
file2.groupby('newDate').mean()['WS'].plot()
plt.gcf().set_size_inches(10,6)
plt.show()
```



In Shenzhen, the wind speed is higher in summer (from July to September) while lower in spring (from Janunary to April). The inter-annual variation is different in last decades, with the highest wind speed appeared in 2017-09 and 2014-07.

3. Explore a data set

The selected dataset is from my group. It's a wind station located at the site (Latitude = 32.8°N， Longitude = 120.9°E), which recorded the 10-min wind speed data at different hub height in the year 2014 and 2015.

In [24]:

```
data = pd.read_csv('windstation.csv')
data
```

Out[24]:

	Date/Time	Speed 100 m [m/s]	Speed 100 m SD [m/s]	Speed 100 m Max [m/s]	...	Speed 10 m [m/s]	Speed 10 m SD [m/s]	Speed 10 m Max [m/s]	Speed 10 m Min [m/s]
0	2014/1/1 0:00	6.7	0.5	7.9	...	2.3	0.3	3.5	1.6
1	2014/1/1 0:10	7.7	0.4	8.7	...	2.8	0.3	3.8	1.9
2	2014/1/1 0:20	7.2	0.5	8.7	...	2.6	0.4	3.8	1.9
3	2014/1/1 0:30	6.8	0.5	7.9	...	2.3	0.3	3.1	1.6
4	2014/1/1 0:40	6.5	0.4	7.2	...	2.5	0.3	3.5	1.9
...	...	...	...	...	...	...	...	...	...
105115	2015/12/31 23:10	1.7	0.3	2.2	...	0.6	0.2	1.2	0.4
105116	2015/12/31 23:20	1.4	0.4	2.2	...	0.5	0.1	0.8	0.4
105117	2015/12/31 23:30	2.1	0.4	3	...	0.5	0.2	1.2	0.4
105118	2015/12/31 23:40	2.2	0.2	2.6	...	0.7	0.3	1.2	0.4
105119	2015/12/31 23:50	2.3	0.3	3	...	0.6	0.2	1.2	0.4

105120 rows × 53 columns

In [25]:

```
# clean the data
data_clean = data.drop(index=data[(np.isnan(data['Speed 100 m [m/s]']))
                                  & (np.isnan(data['Speed 10 m [m/s]']))
                                  & (np.isnan(data['Speed 30 m [m/s]']))
                                  & (np.isnan(data['Speed 50 m [m/s]']))
                                  & (np.isnan(data['Speed 70 m [m/s]']))
                                  & (np.isnan(data['Speed 80 m [m/s]']))
                                  & (np.isnan(data['Speed 90 m [m/s]']))].index)
```

In [26]:
```
def setNewDate(DATE):

    return DATE.split('/')[0] + '-' + DATE.split('/')[1].zfill(2) + '-' + DATE.split('/')[2].split(' ')[0].zf
```

In [27]:
```
data_clean['newDate'] = data_clean.apply(lambda col: setNewDate(col['Date/Time']),axis=1)
```

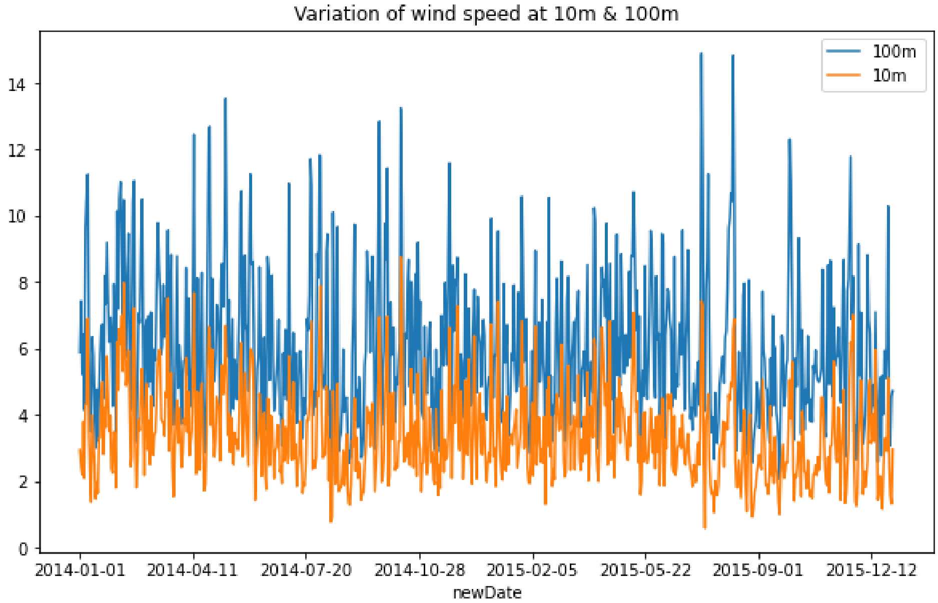
In [28]:
```
data_clean
```

Out[28]:

	Date/Time	Speed 100 m [m/s]	Speed 100 m SD [m/s]	Speed 100 m Max [m/s]	...	Speed 10 m SD [m/s]	Speed 10 m Max [m/s]	Speed 10 m Min [m/s]	newDate
0	2014/1/1 0:00	6.7	0.5	7.9	...	0.3	3.5	1.6	2014-01-01
1	2014/1/1 0:10	7.7	0.4	8.7	...	0.3	3.8	1.9	2014-01-01
2	2014/1/1 0:20	7.2	0.5	8.7	...	0.4	3.8	1.9	2014-01-01
3	2014/1/1 0:30	6.8	0.5	7.9	...	0.3	3.1	1.6	2014-01-01
4	2014/1/1 0:40	6.5	0.4	7.2	...	0.3	3.5	1.9	2014-01-01
...	...	...	...	...	...	...	...	...	...
105115	2015/12/31 23:10	1.7	0.3	2.2	...	0.2	1.2	0.4	2015-12-31
105116	2015/12/31 23:20	1.4	0.4	2.2	...	0.1	0.8	0.4	2015-12-31
105117	2015/12/31 23:30	2.1	0.4	3	...	0.2	1.2	0.4	2015-12-31
105118	2015/12/31 23:40	2.2	0.2	2.6	...	0.3	1.2	0.4	2015-12-31
105119	2015/12/31 23:50	2.3	0.3	3	...	0.2	1.2	0.4	2015-12-31

103162 rows × 54 columns

In [29]:
```
data_clean.groupby('newDate')['Speed 100 m [m/s]'].mean().plot()
plt.gcf().set_size_inches(10,6)


data_clean.groupby('newDate')['Speed 10 m [m/s]'].mean().plot()
plt.gcf().set_size_inches(10,6)
plt.legend(['100m','10m'])
plt.title('Variation of wind speed at 10m & 100m')

plt.show()
```



In [30]:

```
#simple statistical checks
mean100 = np.mean(data_clean['Speed 100 m [m/s]'])
mean10 = np.mean(data_clean['Speed 10 m [m/s]'])

median100 = np.median(data_clean['Speed 100 m [m/s]'])
median10 = np.median(data_clean['Speed 10 m [m/s]'])

variance100 = np.var(data_clean['Speed 100 m [m/s]'])
variance10 = np.var(data_clean['Speed 10 m [m/s]'])

#standard deviation
std100 = np.std(data_clean['Speed 100 m [m/s]'])
std10 = np.std(data_clean['Speed 10 m [m/s]'])

skewness100 = skew(data_clean['Speed 100 m [m/s]'])
skewness10 = skew(data_clean['Speed 10 m [m/s]'])

print(f'Mean wind speed at 100m and 10m : {round(mean100,2)} m/s, {round(mean10,2)} m/s;')
print(f'Median wind speed at 100m and 10m : {round(median100,2)} m/s, {round(median10,2)} m/s;')
print(f'Variance of wind speed at 100m and 10m : {round(variance100,2)} m/s, {round(variance10,2)} m/s;')
print(f'Standard deviation of wind speed at 100m and 10m : {round(std100,2)} m/s, {round(std10,2)} m/s;')
print(f'Skewness of wind speed at 100m and 10m : {round(skewness100,2)} m/s, {round(skewness10,2)} m/s;')
```

```
Mean wind speed at 100m and 10m : 6.06 m/s, 3.44 m/s;
Median wind speed at 100m and 10m : 5.7 m/s, 3.2 m/s;
Variance of wind speed at 100m and 10m : 8.41 m/s, 3.64 m/s;
Standard deviation of wind speed at 100m and 10m : 2.9 m/s, 1.91 m/s;
Skewness of wind speed at 100m and 10m : 0.77 m/s, 0.8 m/s;
```

In [31]:

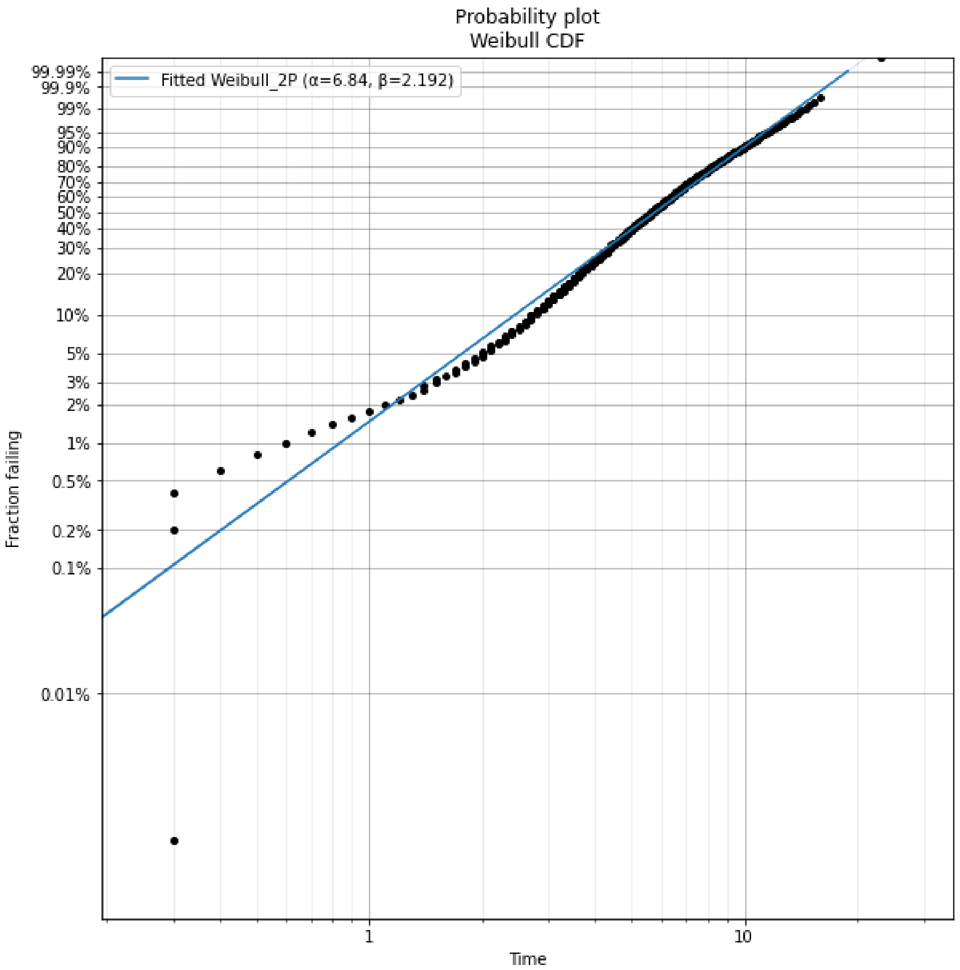
```
#Check the distribution of wind speed: 2-Weibull distribution
def f_weibull(x, beta, alpha):
    return (beta/alpha)*((x/alpha)**(beta-1))*math.exp(-(x/alpha)**beta)
```

Distribution of 100m wind speed:

In [32]:

```
wb_100 = rel.Fitters.Fit_Weibull_2P(failures=data_clean['Speed 100 m [m/s]'].values,print_results=True,show_pr
```

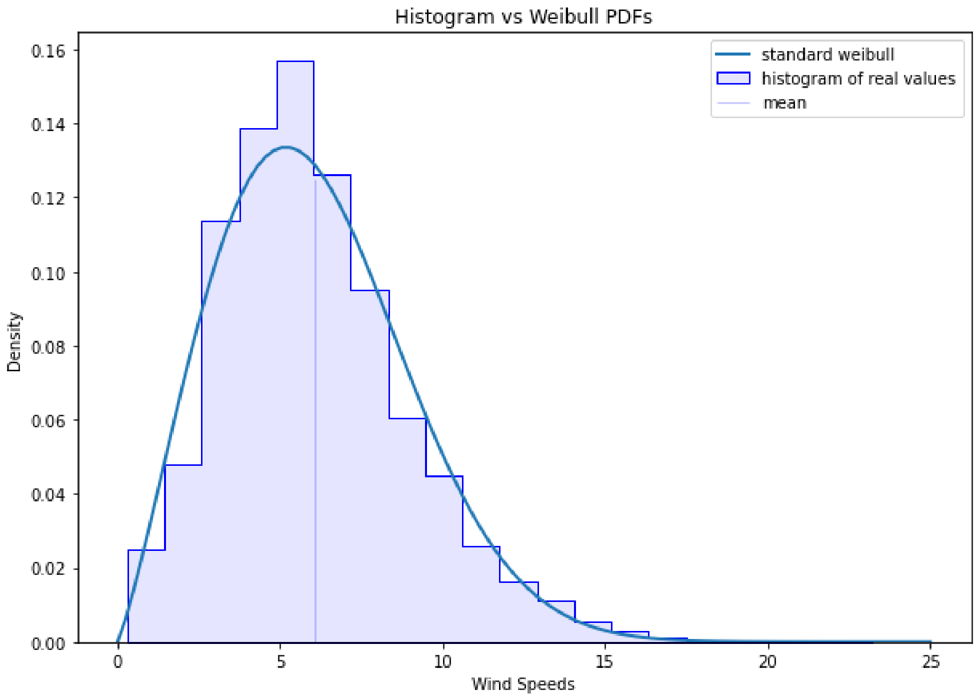
```
Results from Fit_Weibull_2P (95% CI):
Analysis method: Maximum Likelihood Estimation (MLE)
Optimizer: TNC
Failures / Right censored: 103162/0 (0% right censored)

Parameter  Point Estimate  Standard Error  Lower CI  Upper CI
    Alpha         6.83996       0.0102418   6.81991   6.86006
     Beta         2.19151      0.00518093   2.18138   2.20169

Goodness of fit    Value
 Log-likelihood -252077
           AICc   504159
            BIC   504178
             AD  220.552
```



In [33]:

```
bin_count=100
#we sample bin_count number of times from a space of (0,25)
#the maximum wind speed is less than 25 m/s
tries = np.linspace(0,25,bin_count)
beta = wb_100.beta
alpha = wb_100.alpha
glob_pdf = [f_weibull(i, beta, alpha) for i in tries]


fix, ax = plt.subplots(1,1, figsize=(10,7))

ax.set_title('Histogram vs Weibull PDFs')

d1, = ax.plot(tries, glob_pdf, linewidth='2', alpha=1, label='standard weibull')
d2 = sns.histplot(data_clean['Speed 100 m [m/s]'], bins = 20, stat='density', element='step', alpha=0.1, label=
d4 = plt.vlines(np.mean(data_clean['Speed 100 m [m/s]']), ymin=0, ymax=0.125, linewidth=1, alpha=0.2, color='b

ax.set_xlabel('Wind Speeds')

ax.legend()
```

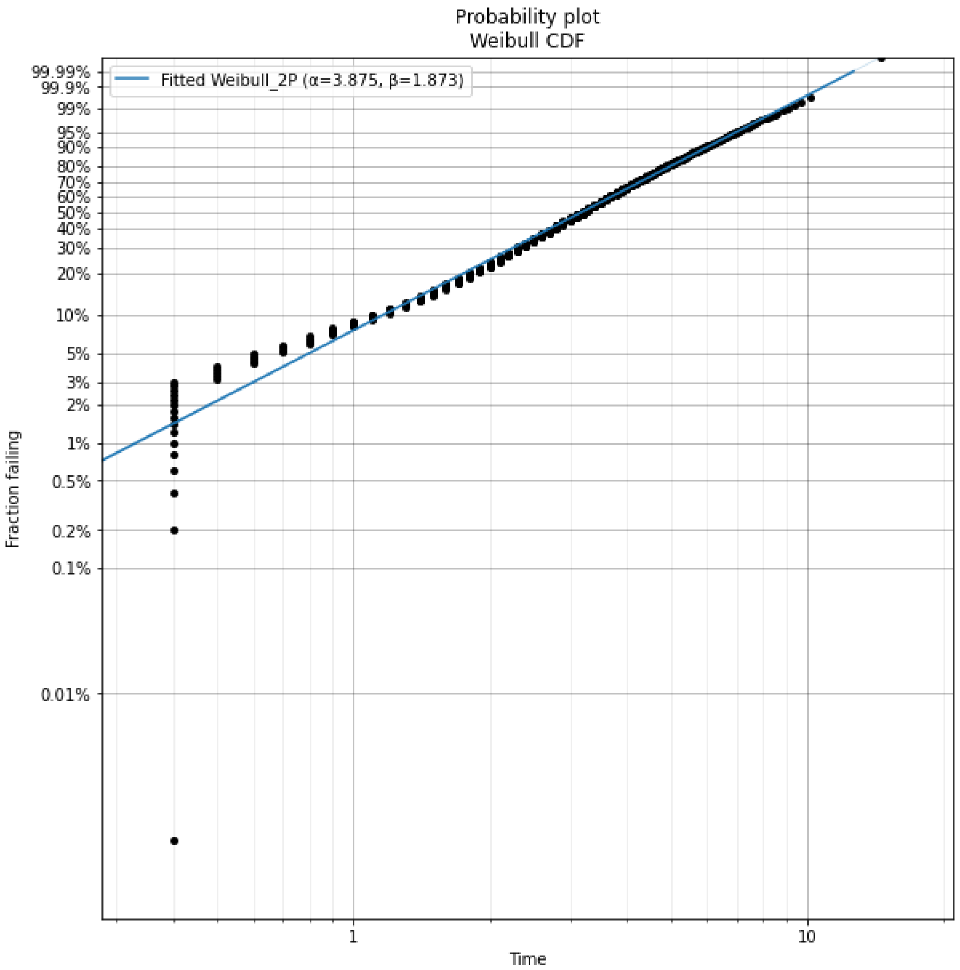
Out[33]: <matplotlib.legend.Legend at 0x148321a90>



Distribution of 10m wind speed:

In [34]:
```
wb_10 = rel.Fitters.Fit_Weibull_2P(failures=data_clean['Speed 10 m [m/s]'].values,print_results=True,show_prob
```

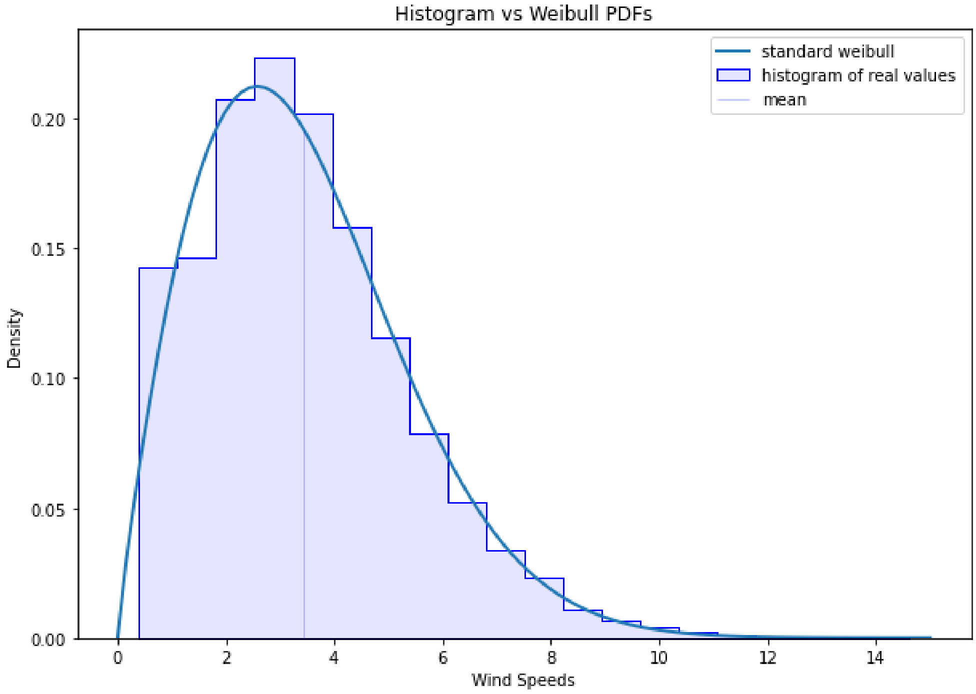
```
Results from Fit_Weibull_2P (95% CI):
Analysis method: Maximum Likelihood Estimation (MLE)
Optimizer: TNC
Failures / Right censored: 103162/0 (0% right censored)

Parameter  Point Estimate  Standard Error  Lower CI  Upper CI
    Alpha         3.87474      0.00678067   3.86147   3.88805
     Beta         1.87301      0.00452186   1.86416   1.88189

Goodness of fit    Value
 Log-likelihood -205831
           AICc  411666
            BIC  411685
             AD 113.843
```



In [35]:

```
bin_count=100
tries = np.linspace(0,15,bin_count)
beta = wb_10.beta
alpha = wb_10.alpha
glob_pdf = [f_weibull(i, beta, alpha) for i in tries]


fix, ax = plt.subplots(1,1, figsize=(10,7))

ax.set_title('Histogram vs Weibull PDFs')

d1, = ax.plot(tries, glob_pdf, linewidth='2', alpha=1, label='standard weibull')
d2 = sns.histplot(data_clean['Speed 10 m [m/s]'], bins = 20, stat='density', element='step', alpha=0.1, label='
d4 = plt.vlines(np.mean(data_clean['Speed 10 m [m/s]']), ymin=0, ymax=0.195, linewidth=1, alpha=0.2, color='blu

ax.set_xlabel('Wind Speeds')

ax.legend()
```

Out[35]: <matplotlib.legend.Legend at 0x148ab0340>



The wind speed becomes higher when the hub height is higher.

In general, the distribution of wind speeds conforms to the 2-parameter-Weibull distribution. Comparing the distribution of wind speed data at different heights for the same site, the two parameters (Alpha, Beta) of the Weibull distribution are 6.8 and 2.2 at 100 meters and 3.9 and 1.9 at 10 meters.