

Name: Roshni Verma

Task 8: Timeline Analysis: Covid-19

Dataset: <https://bit.ly/3Dz2g6i>

Importing Needed Packages

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pylab as pl
import seaborn as sns
import matplotlib inline
sns.set()
```

Importing the Data

```
In [2]: df = pd.read_csv("ovid-covid-data.csv")
df.head()
```

```
Out[2]:
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per
0	AFG	Asia	Afghanistan	2020-02-24	5.0	5.0	NaN	NaN	NaN	NaN	...	NaN	NaN	37.746
1	AFG	Asia	Afghanistan	2020-02-25	5.0	0.0	NaN	NaN	NaN	NaN	...	NaN	NaN	37.746
2	AFG	Asia	Afghanistan	2020-02-26	5.0	0.0	NaN	NaN	NaN	NaN	...	NaN	NaN	37.746
3	AFG	Asia	Afghanistan	2020-02-27	5.0	0.0	NaN	NaN	NaN	NaN	...	NaN	NaN	37.746
4	AFG	Asia	Afghanistan	2020-02-28	5.0	0.0	NaN	NaN	NaN	NaN	...	NaN	NaN	37.746

5 rows × 15 columns

Data Exploration

```
In [3]: #Summarize the data
df.describe()
```

```
Out[3]:
```

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed_per_million	total_...
count	2.226960e+05	2.224350e+05	2.212440e+05	2.033600e+04	20334.000000	20160.000000	22160.000000	221427.000000	220233.000000	...
mean	4.60138e+06	1.22243e+04	1.226187e+04	7.035600e+06	135.244004	135.858975	62982.488874	186.968486	187.185080	...
std	2.814728e+07	8.439823e+04	8.244683e+04	3.866590e+06	720.162353	705.765002	110969.152924	1157.114358	685.835229	...
min	1.000000e+00	0.000000e+00	0.000000e+00	1.000000e+00	0.000000	0.000000	0.001000	0.000000	0.000000	...
25%	4.713750e+03	0.000000e+00	5.714000e+00	1.150000e+02	0.000000	0.000000	1136.444250	0.000000	1.234000	...
50%	5.228650e+04	5.000000e+01	9.028600e+01	1.250000e+03	1.000000	1.429000	10846.471500	6.7274000	18.390000	...
75%	5.465750e+05	9.280000e+02	1.101360e+03	1.064500e+04	13.000000	15.143000	75280.612250	93.842500	132.403000	...
max	6.375120e+08	4.081960e+06	3.430630e+06	6.019800e+06	17733.000000	14859.000000	688923.922000	228872.026000	36421.827000	...

8 rows × 10 columns

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 235924 entries, 0 to 235923
Data columns (total 67 columns):
 # Column                                Non-Null Count  Dtype
---  ---
 0 iso_code                              235924 non-null object
 1 continent                             222614 non-null object
 2 location                              235924 non-null object
 3 date                                  235924 non-null object
 4 total_cases                           222696 non-null float64
 5 new_cases                             222443 non-null float64
 6 new_cases_smoothed                    221244 non-null float64
 7 total_deaths                          20336 non-null float64
 8 new_deaths                            20341 non-null float64
 9 new_deaths_smoothed                  202160 non-null float64
10 total_cases_per_million               221680 non-null float64
11 new_cases_per_million                 221427 non-null float64
12 new_cases_smoothed_per_million        220233 non-null float64
13 total_deaths_per_million              202393 non-null float64
14 new_deaths_per_million                20238 non-null float64
15 new_deaths_smoothed_per_million        20162 non-null float64
16 reproduction_rate                    171567 non-null float64
17 icu_patients                          32863 non-null float64
18 icu_patients_per_million              32863 non-null float64
19 hosp_patients                         35886 non-null float64
20 hosp_patients_per_million             35886 non-null float64
21 weekly_icu_admissions                  8957 non-null float64
22 weekly_icu_admissions_per_million      8957 non-null float64
23 weekly_hosp_admissions                 19823 non-null float64
24 weekly_hosp_admissions_per_million     19823 non-null float64
25 total_tests                           79387 non-null float64
26 new_tests                             75493 non-null float64
27 total_tests_per_thousand              79387 non-null float64
28 new_tests_per_thousand                75493 non-null float64
29 new_tests_smoothed                    193965 non-null float64
30 new_tests_smoothed_per_thousand        193965 non-null float64
31 positive_rate                         89527 non-null float64
32 tests_unit                            16678 non-null object
33 test_per_case                         6758 non-null float64
34 total_vaccinations                    64489 non-null float64
35 people_vaccinated                     61787 non-null float64
36 people_fully_vaccinated                36821 non-null float64
37 total_boosters                        138487 non-null float64
38 new_vaccinations                      138487 non-null float64
39 new_vaccinations_smoothed              138487 non-null float64
40 total_vaccinations_per_hundred         67358 non-null float64
41 people_vaccinated_per_hundred           64489 non-null float64
42 people_fully_vaccinated_per_hundred     31837 non-null float64
43 total_boosters_per_hundred              36821 non-null float64
44 new_vaccinations_smoothed_per_million   138487 non-null float64
45 new_people_vaccinated_smoothed          138487 non-null float64
46 new_people_vaccinated_smoothed_per_hundred 138487 non-null float64
47 stringency_index                      174922 non-null float64
48 population_density                    209042 non-null float64
49 median_age                             19982 non-null float64
50 aged_65_and_over                       188591 non-null float64
51 aged_70_and_over                       189895 non-null float64
52 gdp_per_capita                         120832 non-null float64
53 extreme_poverty                       19874 non-null float64
54 cardiovasc_death_rate                  199913 non-null float64
55 diabetes_prevalence                   143966 non-null float64
56 female_smokers                         142925 non-null float64
57 male_smokers                           142925 non-null float64
58 handwashing_facilities                 21649 non-null float64
59 hospital_beds_per_thousand              18534 non-null float64
60 life_expectancy                        23498 non-null float64
61 human_development_index                7766 non-null float64
62 population                             7766 non-null float64
63 total_mortality_cumulative_absolute      7766 non-null float64
64 excess_mortality_cumulative             7766 non-null float64
65 excess_mortality_cumulative_per_million 7766 non-null float64
66 excess_mortality_cumulative_per_million float64(62), object(5)
dtypes: float64(62), object(5)
memory usage: 16.4+ MB
```

Dropping the Duplicates rows

```
In [6]: df.shape
```

```
Out[6]: (235924, 67)
```

```
In [7]: duplicate_rows_df = df[df.duplicated()]
df.shape
```

```
Out[7]: (235924, 67)
```

Number of duplicate rows: 0, 67

```
In [8]: df.count()
```

```
Out[8]:
```

	iso_code	continent	location	date	total_cases	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
count	235924	222614	235924	235924	222696	234988	7766	7766	7817	7766

Length: 67, dtype: int64

Checking the Missing or Null values

```
In [9]: df.isnull().sum()
```

```
Out[9]:
```

	iso_code	continent	location	date	total_cases	population	excess_mortality_cumulative_absolute	excess_mortality_cumulative	excess_mortality	excess_mortality_cumulative_per_million
count	0	13310	0	0	13228	1016	228158	228158	228158	228158

```
In [10]: df = df[['date', 'total_cases', 'new_cases', 'total_deaths', 'new_tests', 'total_tests']]
df.head()
```

```
Out[10]:
```

	date	total_cases	new_cases	total_deaths	new_tests	total_tests
0	2020-02-24	5.0	5.0	NaN	NaN	NaN
1	2020-02-25	5.0	0.0	NaN	NaN	NaN
2	2020-02-26	5.0	0.0	NaN	NaN	NaN
3	2020-02-27	5.0	0.0	NaN	NaN	NaN
4	2020-02-28	5.0	0.0	NaN	NaN	NaN

```
In [11]: df.date = pd.to_datetime(df.date)
df.set_index('date', inplace = True)
```

```
Out[11]:
```

C:\Users\sony\anaconda3\lib\site-packages\pandas\core\generic.py:5516: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
self[name] = value

```
In [12]: df.head()
```

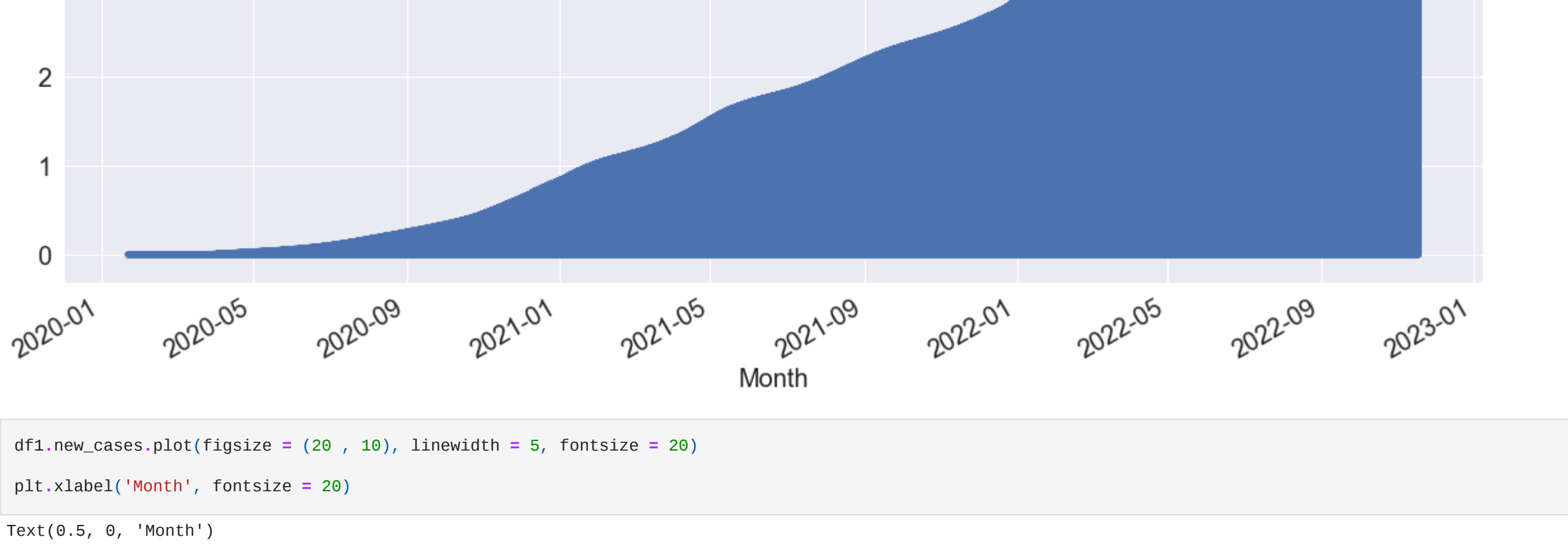
```
Out[12]:
```

	date	total_cases	new_cases	total_deaths	new_tests	total_tests
0	2020-02-24	5.0	5.0	NaN	NaN	NaN
2020-02-25	5.0	0.0	NaN	NaN	NaN	NaN
2020-02-26	5.0	0.0	NaN	NaN	NaN	NaN
2020-02-27	5.0	0.0	NaN	NaN	NaN	NaN
2020-02-28	5.0	0.0	NaN	NaN	NaN	NaN

```
In [13]: df1.plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20)
```

```
Out[13]:
```

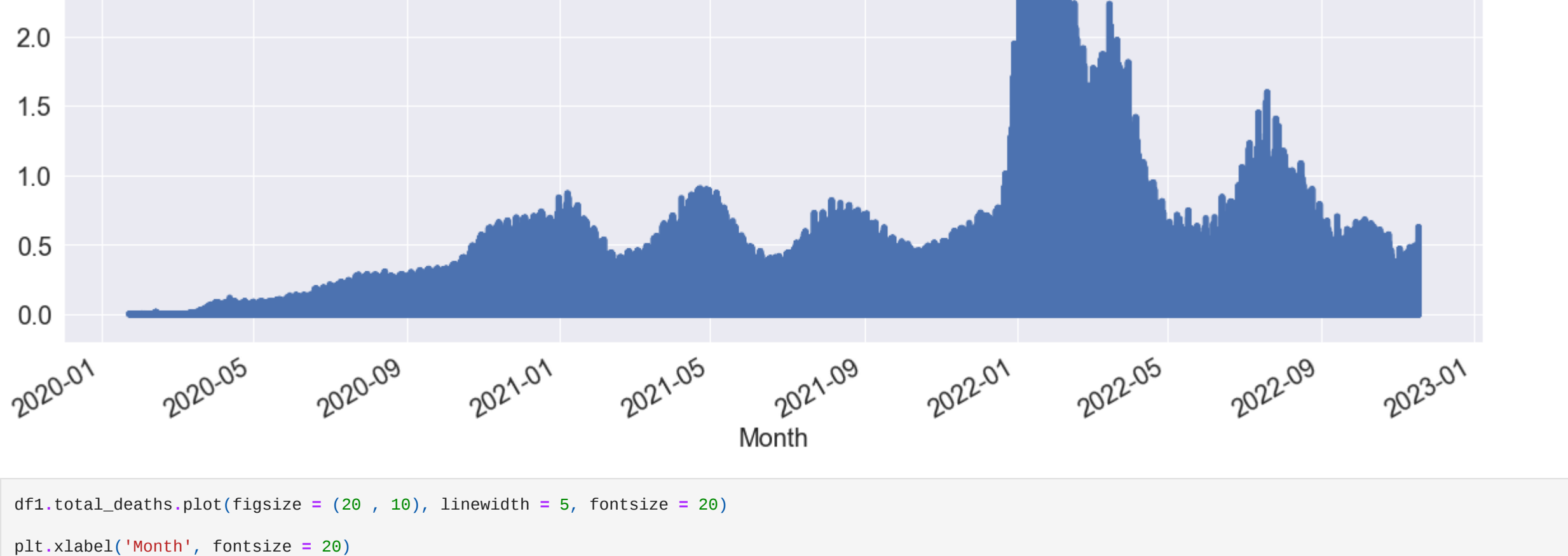
Text(0.5, 0, 'Month')



```
In [14]: df1.total_cases.plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20)
```

```
Out[14]:
```

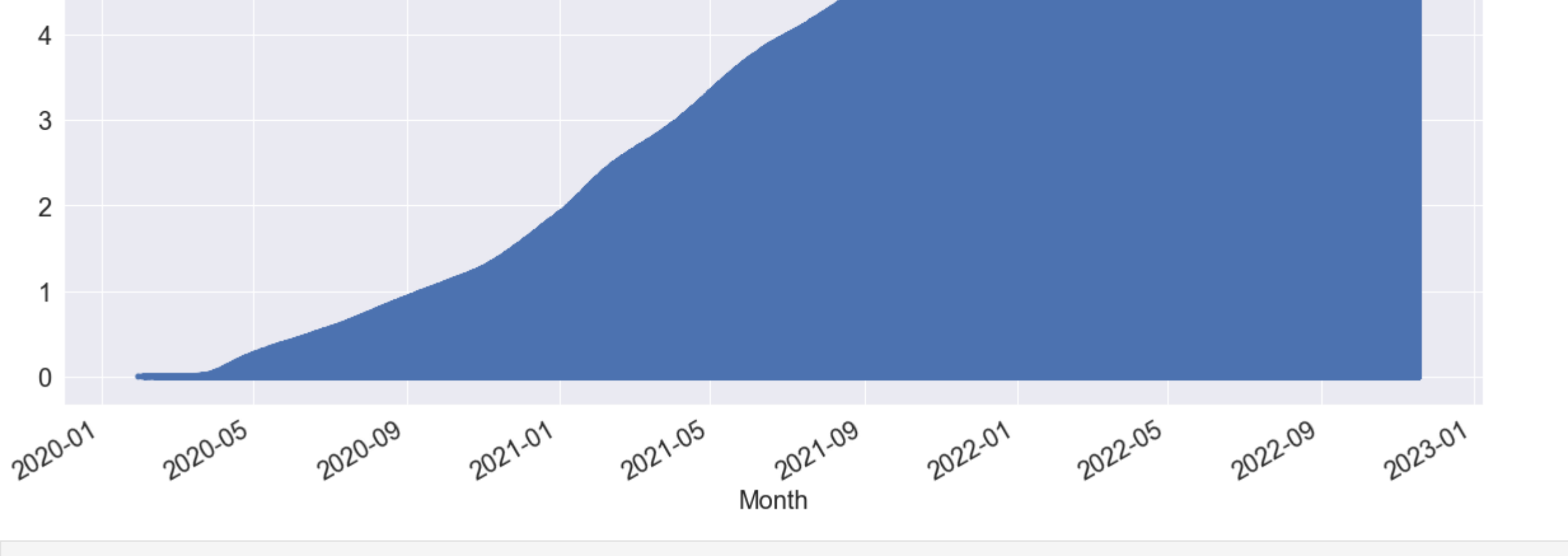
Text(0.5, 0, 'Month')



```
In [15]: df1.new_cases.plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20)
```

```
Out[15]:
```

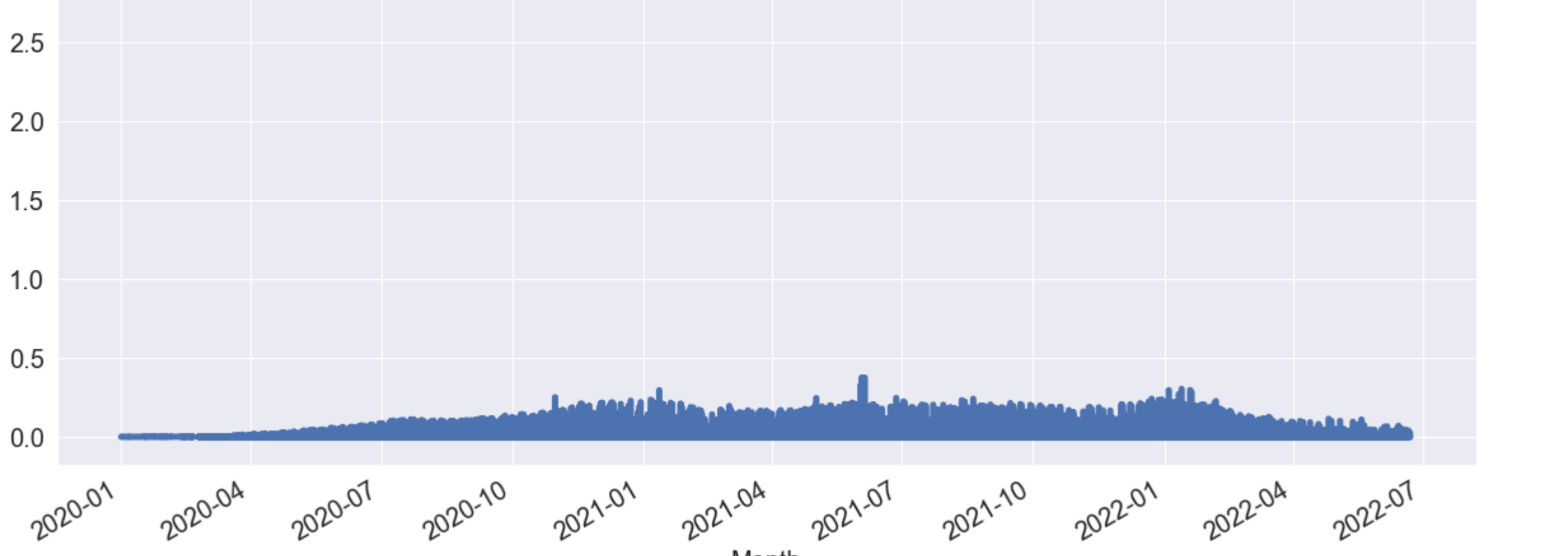
Text(0.5, 0, 'Month')



```
In [16]: df1.total_deaths.plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20)
```

```
Out[16]:
```

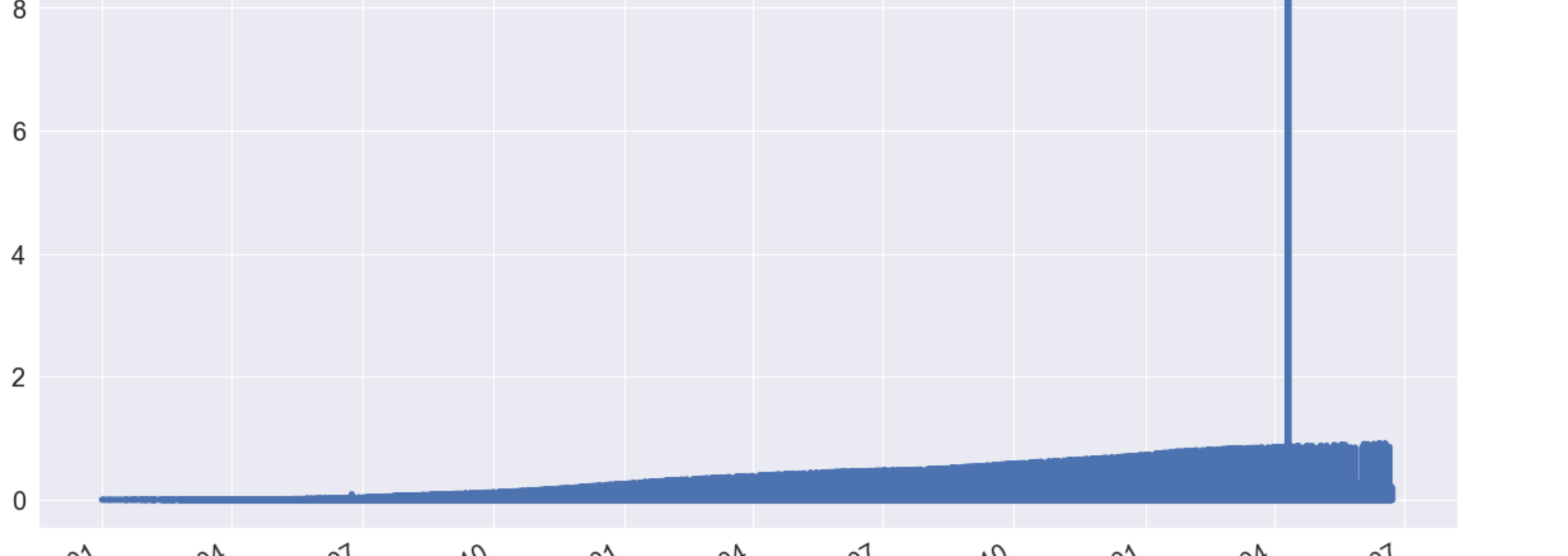
Text(0.5, 0, 'Month')



```
In [17]: df1.new_tests.plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20)
```

```
Out[17]:
```

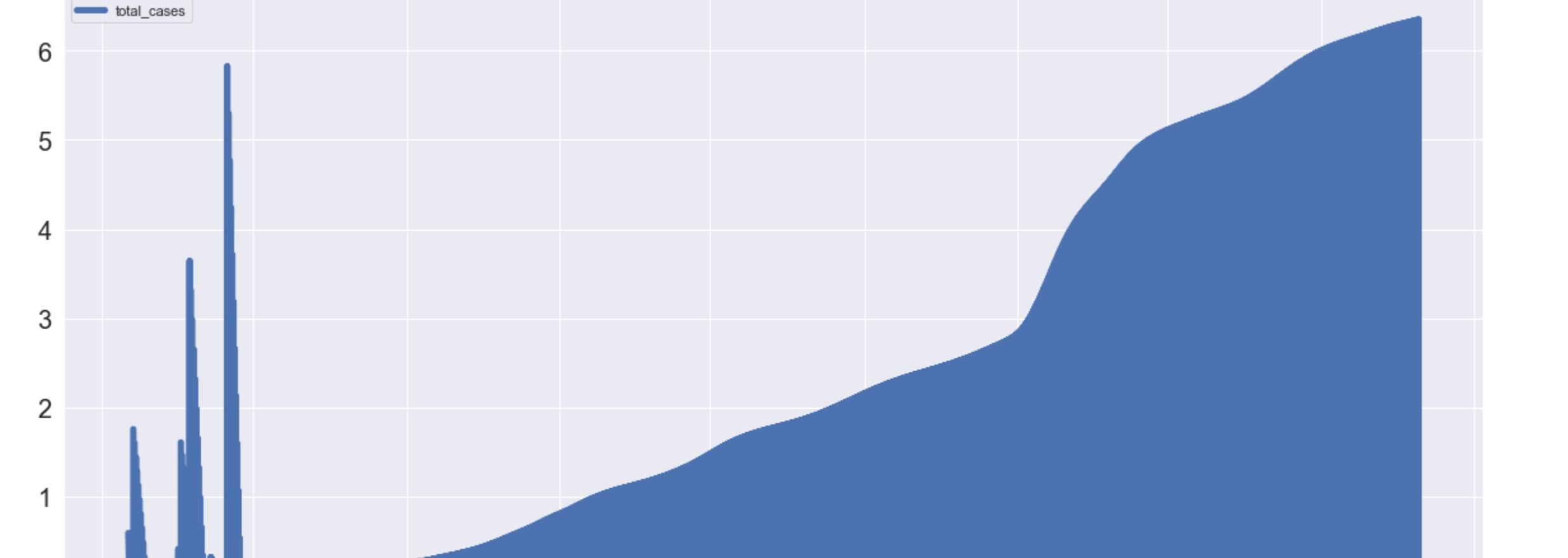
Text(0.5, 0, 'Month')



```
In [18]: df1.total_tests.plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20)
```

```
Out[18]:
```

Text(0.5, 0, 'Month')

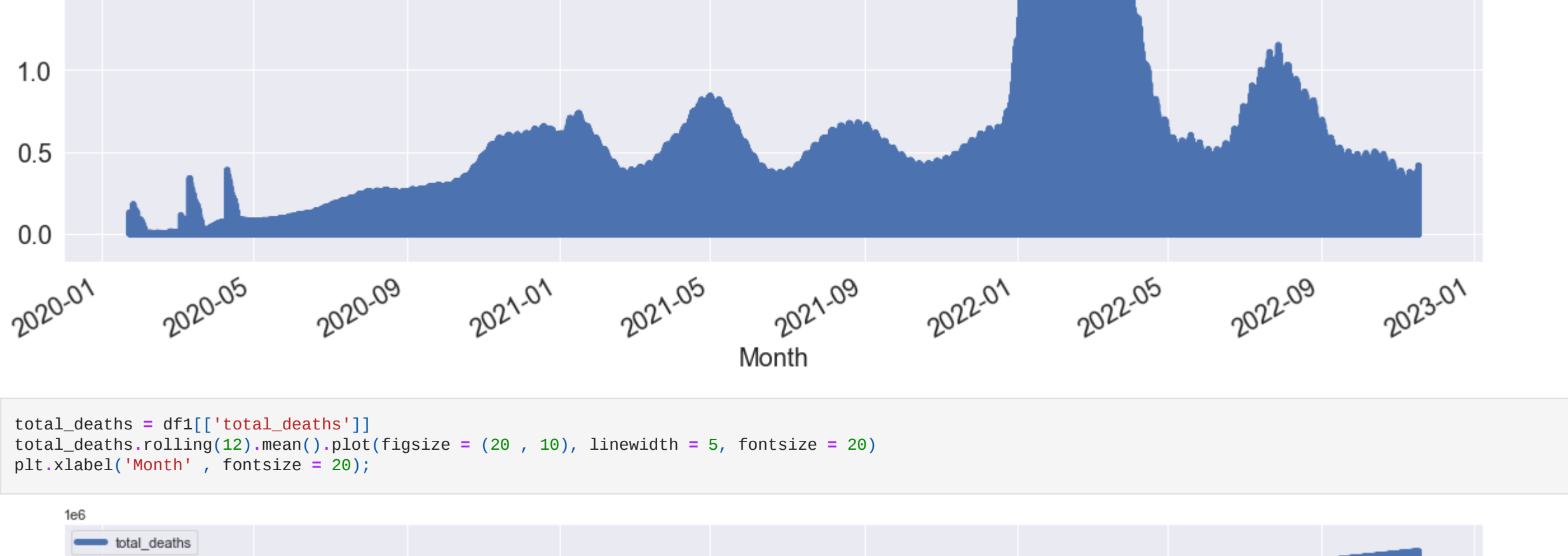


Trends and Seasonality in TimeSeries Data

```
In [19]: total_cases = df1[['total_cases']]
total_cases.rolling(12).mean().plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[19]:
```

Text(0.5, 0, 'Month')



```
In [20]: new_cases = df1[['new_cases']]
new_cases.rolling(12).mean().plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[20]:
```

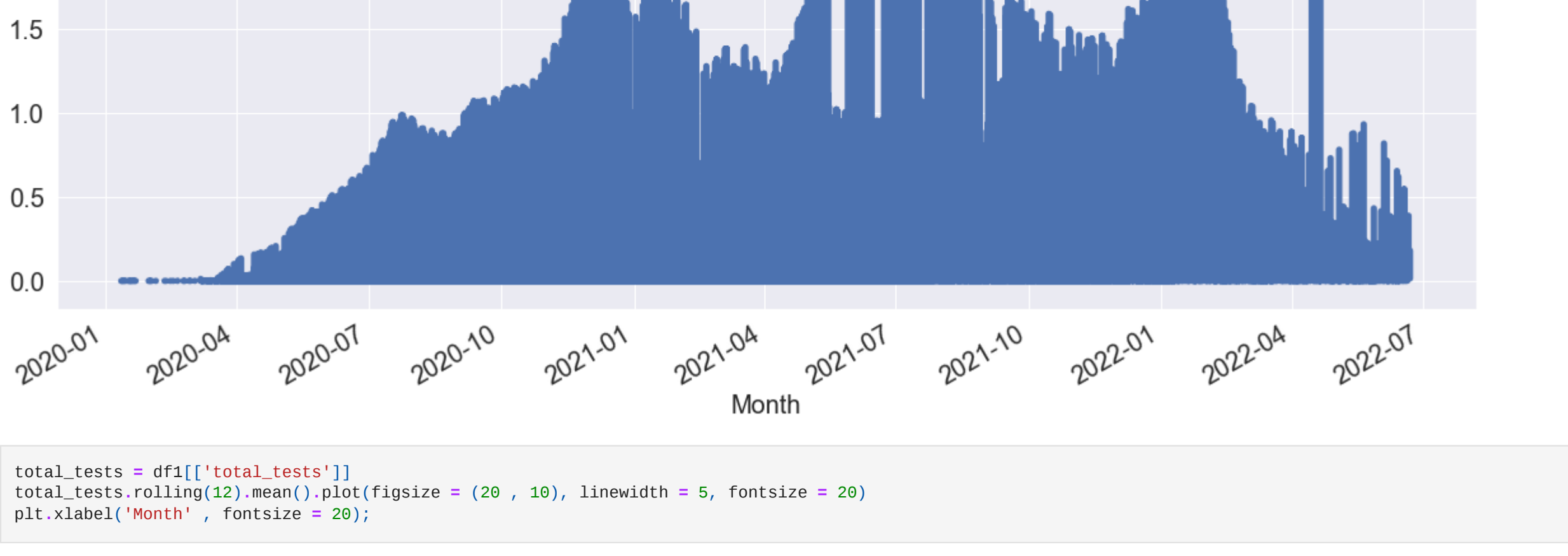
Text(0.5, 0, 'Month')



```
In [21]: total_deaths = df1[['total_deaths']]
total_deaths.rolling(12).mean().plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[21]:
```

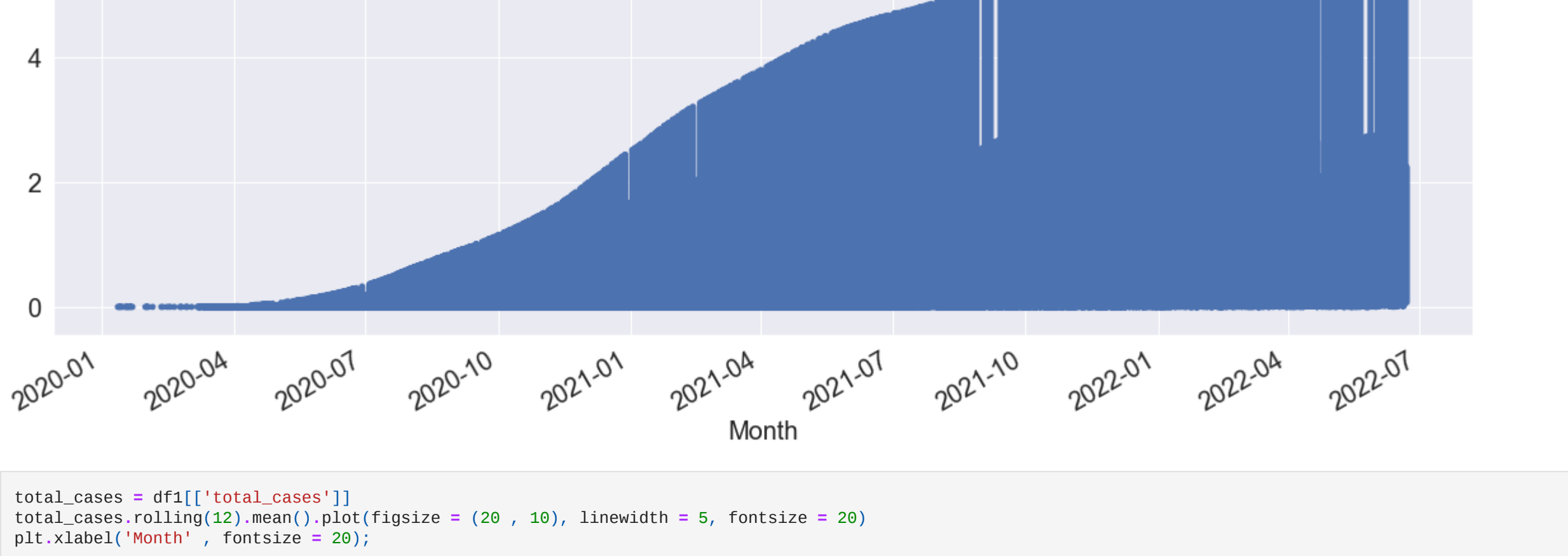
Text(0.5, 0, 'Month')



```
In [22]: new_tests = df1[['new_tests']]
new_tests.rolling(12).mean().plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[22]:
```

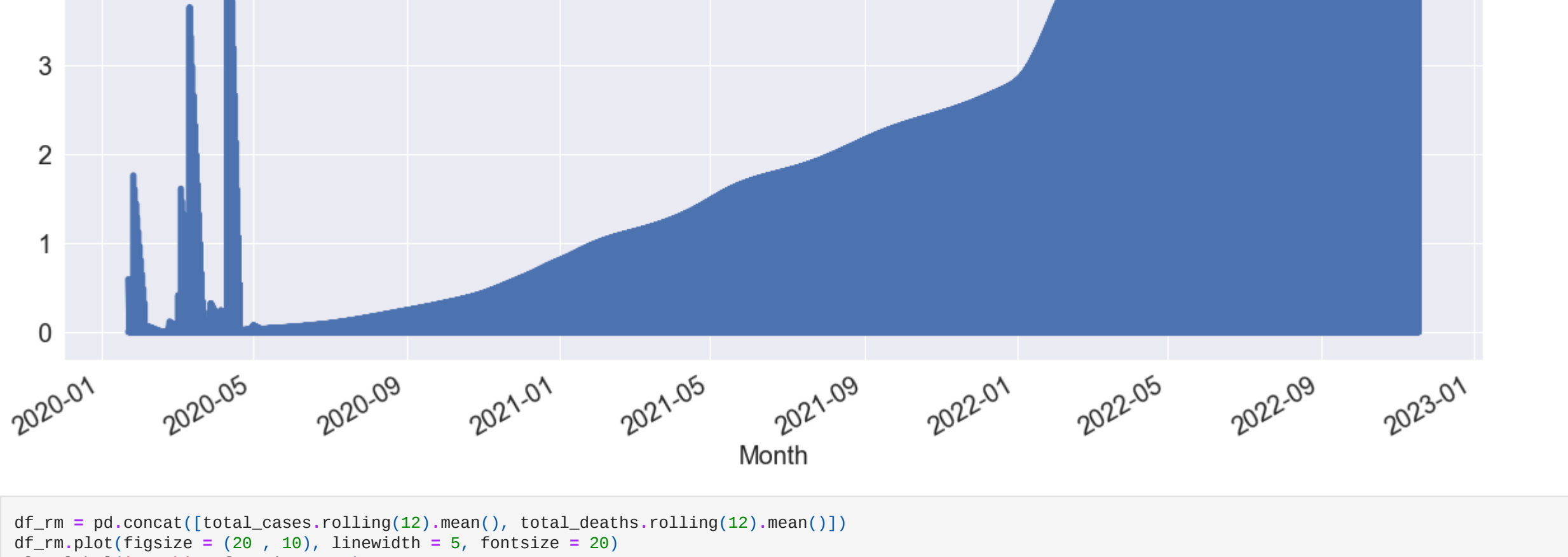
Text(0.5, 0, 'Month')



```
In [23]: total_tests = df1[['total_tests']]
total_tests.rolling(12).mean().plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[23]:
```

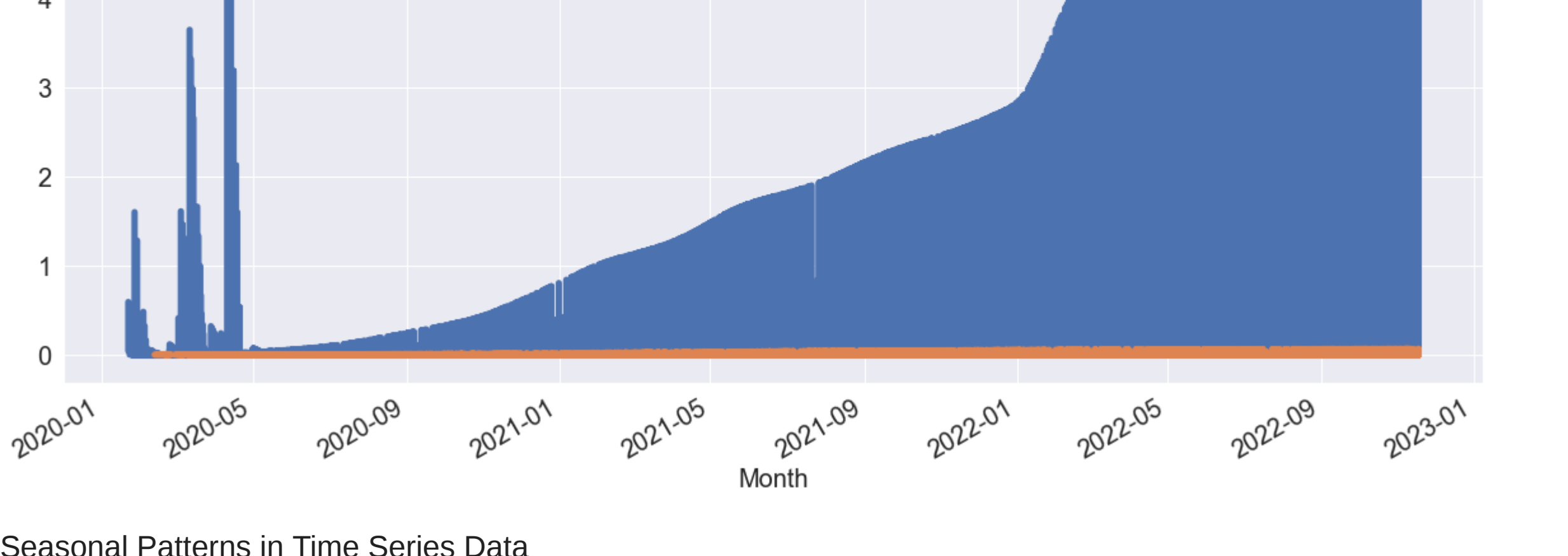
Text(0.5, 0, 'Month')



```
In [24]: total_cases = df1[['total_cases']]
total_tests.rolling(12).mean().plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[24]:
```

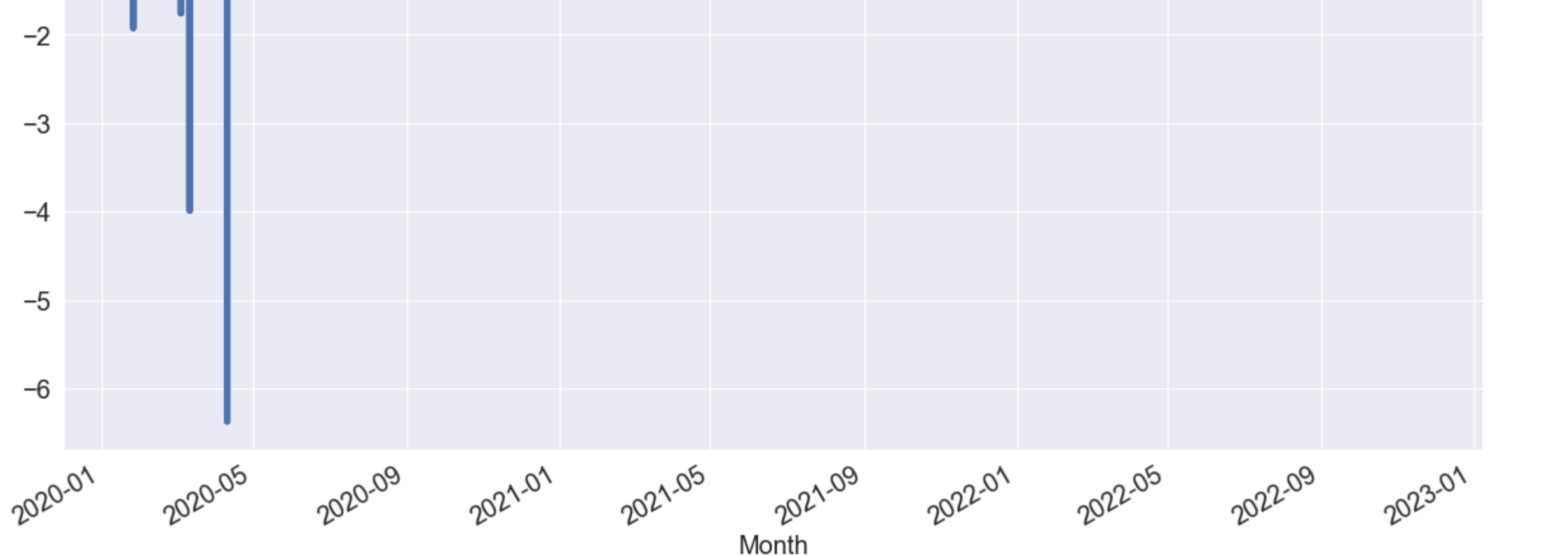
Text(0.5, 0, 'Month')



```
In [25]: df_res = pd.concat([total_cases.rolling(12).mean(), total_deaths.rolling(12).mean()])
df_res.plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[25]:
```

Text(0.5, 0, 'Month')



Seasonal Patterns in Time Series Data

```
In [26]: total_cases.diff().plot(figsize = (20 , 10), linewidth = 5, fontsize = 20)
plt.xlabel('Month', fontsize = 20);
```

```
Out[26]:
```

Text(0.5, 0, 'Month')

