In [71]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from unicodedata import normalize

#first download csv file from URL
#local path of csv file
USVaccURL="https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/
#read CSV file to pandas dataframe
dfUSVacc=pd.read_csv(USVaccURL)
#print dataframe
dfUSVacc
```

Out[71]:

	location	date	vaccine	source_url	total_vaccinations	people_vaccinated	people_
0	United States	2020-12-13	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	29430	24506	
1	United States	2020-12-14	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	33992	28909	
2	United States	2020-12-15	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	83686	76274	
3	United States	2020-12-16	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	243463	230662	
4	United States	2020-12-17	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	515903	495984	
...	...	...	...	...	...	...	
450	United States	2022-03-08	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556502259	254426780	
451	United States	2022-03-09	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556686776	254476000	
452	United States	2022-03-10	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556850740	254518093	
453	United States	2022-03-11	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556968093	254547815	
454	United States	2022-03-12	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556980091	254551489	

455 rows × 8 columns

In [72]:

```
dfUSVacc.columns
```

Out[72]:

```
Index(['location', 'date', 'vaccine', 'source_url', 'total_vaccinations',
       'people_vaccinated', 'people_fully_vaccinated', 'total_boosters'],
      dtype='object')
```

In [73]:

```
#Inspect data
```

```
dfUSVacc.head()
```

Out[73]:

	location	date	vaccine	source_url	total_vaccinations	people_vaccinated	people_fully_
0	United States	2020-12-13	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	29430	24506	
1	United States	2020-12-14	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	33992	28909	
2	United States	2020-12-15	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	83686	76274	
3	United States	2020-12-16	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	243463	230662	
4	United States	2020-12-17	Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	515903	495984	

In [74]:

```
dfUSVacc.tail()
```

Out[74]:

	location	date	vaccine	source_url	total_vaccinations	people_vaccinated	people_
450	United States	2022-03-08	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556502259	254426780	
451	United States	2022-03-09	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556686776	254476000	
452	United States	2022-03-10	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556850740	254518093	
453	United States	2022-03-11	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556968093	254547815	
454	United States	2022-03-12	Johnson&Johnson, Moderna, Pfizer/BioNTech	https://data.cdc.gov /Vaccinations /COVID-19-Vac...	556980091	254551489	

In [75]:

```
#to get information about column of dataset
dfUSVacc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 455 entries, 0 to 454
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype 
---  ------                   --------------  ----- 
 0   location                 455 non-null    object
 1   date                     455 non-null    object
 2   vaccine                  455 non-null    object
 3   source_url               455 non-null    object
 4   total_vaccinations       455 non-null    int64 
 5   people_vaccinated        455 non-null    int64 
 6   people_fully_vaccinated  455 non-null    int64 
 7   total_boosters           455 non-null    int64 
dtypes: int64(4), object(4)
memory usage: 28.6+ KB
```

In [76]:

```
#removed unecessary columns (vaccine,source_url)
```

```
dfUSVacc.drop(['location','vaccine','source_url','total_vaccinations','total_booste
```

In [77]:

```
#Columns after removed unecessary columns
dfUSVacc.columns
```

Out[77]:

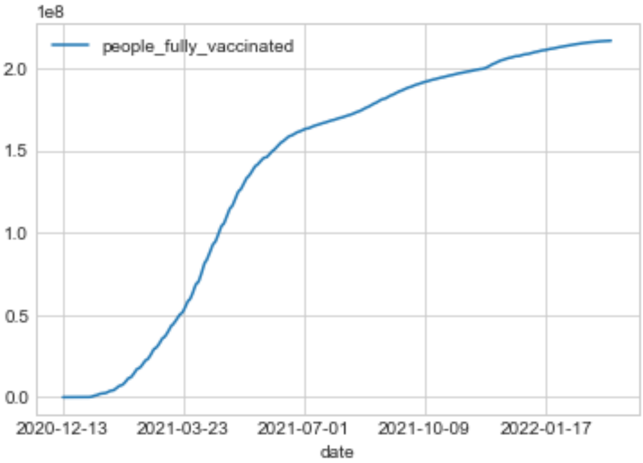
```
Index(['date', 'people_fully_vaccinated'], dtype='object')
```

In [78]:

```
plt.style.use('seaborn-whitegrid')
dfUSVacc.plot.line(x='date', y=['people_fully_vaccinated'])
```

Out[78]:

```
<AxesSubplot:xlabel='date'>
```



In [79]:

```
#Columns information after removed unecessary columns
dfUSVacc.info
```

Out[79]:

```
<bound method DataFrame.info of             date  people_fully_vaccinated
0    2020-12-13                     5662
1    2020-12-14                     5781
2    2020-12-15                     6041
3    2020-12-16                     6513
4    2020-12-17                     7257
..          ...                      ...
450  2022-03-08                216512109
451  2022-03-09                216563462
452  2022-03-10                216609285
453  2022-03-11                216643718
454  2022-03-12                216647869

[455 rows x 2 columns]>
```

In [80]:

```
dfUSVacc
```

Out[80]:

	date	people_fully_vaccinated
0	2020-12-13	5662
1	2020-12-14	5781
2	2020-12-15	6041
3	2020-12-16	6513
4	2020-12-17	7257
...	...	...
450	2022-03-08	216512109

	date	people_fully_vaccinated
451	2022-03-09	216563462
452	2022-03-10	216609285
453	2022-03-11	216643718
454	2022-03-12	216647869

In [81]:
```
dfUSVacc.describe()
```

Out[81]:

	people_fully_vaccinated
count	4.550000e+02
mean	1.387127e+08
std	7.510954e+07
min	5.662000e+03
25%	7.237704e+07
50%	1.691354e+08
75%	1.987061e+08
max	2.166479e+08

In [82]:
```
dfUSVacc.mean()
```

```
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/4170092832.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=No
ne') is deprecated; in a future version this will raise TypeError.  Select only
valid columns before calling the reduction.
  dfUSVacc.mean()
```

Out[82]:
```
people_fully_vaccinated    1.387127e+08
dtype: float64
```

In [83]:
```
dfUSVacc.median()
```

```
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/3812337837.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=No
ne') is deprecated; in a future version this will raise TypeError.  Select only
valid columns before calling the reduction.
  dfUSVacc.median()
```

Out[83]:
```
people_fully_vaccinated    169135405.0
dtype: float64
```

In [84]:
```
dfUSVacc.std()
```

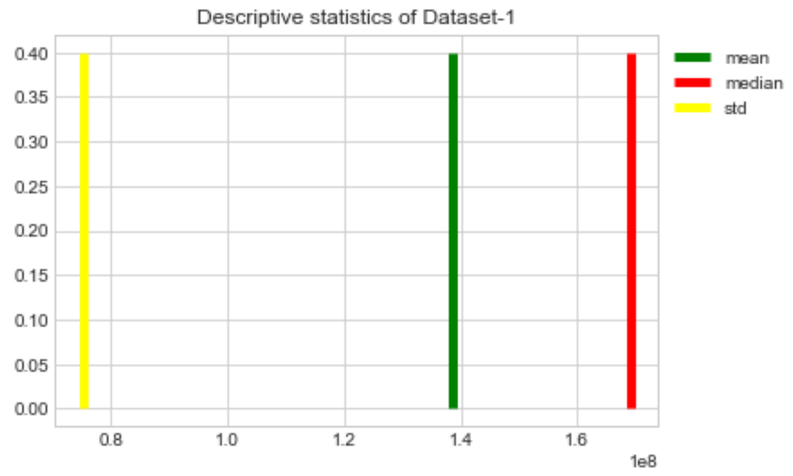
```
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/2934334280.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=No
ne') is deprecated; in a future version this will raise TypeError.  Select only
valid columns before calling the reduction.
  dfUSVacc.std()
```

Out[84]:
```
people_fully_vaccinated    7.510954e+07
dtype: float64
```

In [85]:
```
arr1 = plt.vlines(dfUSVacc.mean(), # Plot green line at mean
 ymin=0,
 ymax=0.4,
 linewidth=5.0,
 colors = "green"
 );
arr2 = plt.vlines(dfUSVacc.median(), # Plot red line at median
 ymin=0,
 ymax=0.4,
 linewidth=5.0,
 colors="red",
 );
arr3 = plt.vlines(dfUSVacc.std(), # Plot yellow line at std
 ymin=0,
 ymax=0.4,
 linewidth=5.0,
 colors="yellow"
 );
# place legend outside
#plt.legend(bbox_to_anchor=(1.0, 1), loc='upper left', title=('green - mean\nred -

plt.legend([arr1, arr2, arr3], ['mean','median','std'],bbox_to_anchor=(1.0, 1),loc=
plt.title('Descriptive statistics of Dataset-1')
plt.show()
```

```
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/3248225235.py:1: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=No
ne') is deprecated; in a future version this will raise TypeError.  Select only
valid columns before calling the reduction.
  arr1 = plt.vlines(dfUSVacc.mean(), # Plot green line at mean
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/3248225235.py:7: FutureWarn
ing: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=No
ne') is deprecated; in a future version this will raise TypeError.  Select only
valid columns before calling the reduction.
  arr2 = plt.vlines(dfUSVacc.median(), # Plot red line at median
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/3248225235.py:13: FutureWar
ning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=N
one') is deprecated; in a future version this will raise TypeError.  Select only
valid columns before calling the reduction.
  arr3 = plt.vlines(dfUSVacc.std(), # Plot yellow line at std
```



In [86]:

```
#Monthly grouping
dfUSVacc['date'] = pd.to_datetime(dfUSVacc['date'])
dfUSVacc = dfUSVacc.sort_values(by='date')
dfUSVacc=dfUSVacc.groupby(pd.DatetimeIndex(dfUSVacc.date).to_period('M')).nth([-1])
dfUSVacc.set_index('date', inplace=True)
#dfUSVacc.to_csv('dfUSVacc.csv')
#dfUSVacc
```

In [87]:
```
dfUSVacc
```

Out[87]:

	people_fully_vaccinated
date	
2020-12-31	40617
2021-01-31	7348908
2021-02-28	29858974
2021-03-31	64155921
2021-04-30	114120988
2021-05-31	145938973
2021-06-30	162447516
2021-07-31	169858404
2021-08-31	179616223
2021-09-30	189466803
2021-10-31	195718966
2021-11-30	200727384
2021-12-31	208380521
2022-01-31	213096238
2022-02-28	216059627
2022-03-12	216647869

In [88]:
```
#load covid19 death,confirmed and recovered data
urlCovid19='https://raw.githubusercontent.com/datasets/covid-19/master/data/countri
dfUSCovid19Data=pd.read_csv(urlCovid19)
dfUSCovid19Data.rename(columns={'Date': 'date'}, inplace=True)
# Print Initial dataset of covid-19
dfUSCovid19Data
```

Out[88]:

	date	Country	Confirmed	Recovered	Deaths
0	2020-01-22	Afghanistan	0	0	0
1	2020-01-23	Afghanistan	0	0	0
2	2020-01-24	Afghanistan	0	0	0
3	2020-01-25	Afghanistan	0	0	0
4	2020-01-26	Afghanistan	0	0	0
...	...	...	...	...	...
154831	2022-03-09	Zimbabwe	240343	0	5400
154832	2022-03-10	Zimbabwe	241548	0	5408
154833	2022-03-11	Zimbabwe	241548	0	5408
154834	2022-03-12	Zimbabwe	242069	0	5412
154835	2022-03-13	Zimbabwe	242515	0	5414

154836 rows × 5 columns

In [89]:
```
#Initial total columns of dfUSCovid19Data dataset
```

```
dfUSCovid19Data.columns
```

Out[89]:
```
Index(['date', 'Country', 'Confirmed', 'Recovered', 'Deaths'], dtype='object')
```

In [90]:
```
#Initial columns info of dfUSCovid19Data dataset
dfUSCovid19Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 154836 entries, 0 to 154835
Data columns (total 5 columns):
 #   Column     Non-Null Count   Dtype 
---  ------     --------------   ----- 
 0   date       154836 non-null  object
 1   Country    154836 non-null  object
 2   Confirmed  154836 non-null  int64 
 3   Recovered  154836 non-null  int64 
 4   Deaths     154836 non-null  int64 
dtypes: int64(3), object(2)
memory usage: 5.9+ MB
```

In [91]:
```
#filtering data for considering only USA data
dfUSCovid19Data=dfUSCovid19Data.loc[(dfUSCovid19Data['Country'] == 'US') & (dfUSCov
```

In [92]:
```
# Printing Covid-19 USA Data
dfUSCovid19Data
```

Out[92]:

	date	Country	Confirmed	Recovered	Deaths
143116	2020-02-01	US	8	0	0
143117	2020-02-02	US	8	0	0
143118	2020-02-03	US	11	0	0
143119	2020-02-04	US	11	0	0
143120	2020-02-05	US	11	0	0
...	...	...	...	...	...
143870	2022-02-24	US	78812640	0	946099
143871	2022-02-25	US	78887236	0	948130
143872	2022-02-26	US	78934671	0	948826
143873	2022-02-27	US	78950518	0	949018
143874	2022-02-28	US	79047371	0	951114

759 rows × 5 columns

In [93]:
```
dfUSCovid19Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 759 entries, 143116 to 143874
Data columns (total 5 columns):
 #   Column     Non-Null Count  Dtype 
---  ------     --------------  ----- 
 0   date       759 non-null    object
 1   Country    759 non-null    object
 2   Confirmed  759 non-null    int64 
 3   Recovered  759 non-null    int64 
 4   Deaths     759 non-null    int64 
dtypes: int64(3), object(2)
memory usage: 35.6+ KB
```

In [94]:

```
#removed unnecessary columns, calculating death rate, and grouping
dfUSCovid19Data.drop(['Recovered'],axis=1,inplace=True)
dfUSCovid19Data['Deaths'] =  (dfUSCovid19Data['Deaths']/dfUSCovid19Data['Confirmed'
dfUSCovid19Data['date'] = pd.to_datetime(dfUSCovid19Data['date'])
dfUSCovid19Data=dfUSCovid19Data.groupby(pd.Grouper(key='date', axis=0, freq='M')).s
```

```
C:\Users\Taslima Akter\anaconda3\lib\site-packages\pandas\core\frame.py:4906: Se
ttingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydat
a.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  return super().drop(
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/2776449457.py:3: SettingWit
hCopyWarning:
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/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydat
a.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  dfUSCovid19Data['Deaths'] =  (dfUSCovid19Data['Deaths']/dfUSCovid19Data['Confi
rmed'])*1
C:\Users\TASLIM~1\AppData\Local\Temp/ipykernel_11472/2776449457.py:4: SettingWit
hCopyWarning:
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/stab
le/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydat
a.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-cop
y)
  dfUSCovid19Data['date'] = pd.to_datetime(dfUSCovid19Data['date'])
```

In [95]:

```
# Printing covid-19 Data of USA after removing unnecessary columns and grouping
dfUSCovid19Data
```

Out[95]:

	Confirmed	Deaths
date		
2020-02-29	402	0.040000
2020-03-31	1121455	1.073644
2020-04-30	19835424	1.508744
2020-05-31	45294659	1.917086
2020-06-30	64822529	1.663059
2020-07-31	111086834	1.225709
2020-08-31	166531654	0.981105
2020-09-30	199608857	0.884212
2020-10-31	251226672	0.839544
2020-11-30	338160262	0.671164
2020-12-31	527988498	0.570083

	Confirmed	Deaths
date		
2021-01-31	733787758	0.524889
2021-02-28	776848931	0.490191
2021-03-31	919661679	0.559992
2021-04-30	947668934	0.536053
2021-05-31	1023439204	0.549538
2021-06-30	1007947426	0.535629
2021-07-31	1061524357	0.550731
2021-08-31	1151033303	0.521806
2021-09-30	1250238382	0.482108
2021-10-31	1392850378	0.500566
2021-11-30	1419741886	0.485748
2021-12-31	1578107885	0.491237

In [96]:

```
#before vaccination in the year 2020 , death rate
dfUSCovid19Data.drop(['Confirmed'],axis=1,inplace=True)
print(dfUSCovid19Data)
#dfUSCovid19Data.to_csv('dfUSCovid19Data.csv')
#dfUSCovid19Data.rename(columns={'Confirmed': 'covid_postive'}, inplace=True)
```

```
              Deaths
date
2020-02-29  0.040000
2020-03-31  1.073644
2020-04-30  1.508744
2020-05-31  1.917086
2020-06-30  1.663059
2020-07-31  1.225709
2020-08-31  0.981105
2020-09-30  0.884212
2020-10-31  0.839544
2020-11-30  0.671164
2020-12-31  0.570083
2021-01-31  0.524889
2021-02-28  0.490191
2021-03-31  0.559992
2021-04-30  0.536053
2021-05-31  0.549538
2021-06-30  0.535629
2021-07-31  0.550731
2021-08-31  0.521806
2021-09-30  0.482108
2021-10-31  0.500566
2021-11-30  0.485748
2021-12-31  0.491237
2022-01-31  0.406118
2022-02-28  0.333173
```

In [97]:

```
dfUSCovid19Data.describe()
```

Out[97]:

	Deaths
count	25.000000
mean	0.733685

	Deaths
std	0.440583
min	0.040000
25%	0.491237
50%	0.549538
75%	0.884212

```
In [98]: dfUSCovid19Data.mean()
```

```
Out[98]: Deaths    0.733685
         dtype: float64
```

```
In [99]: dfUSCovid19Data.median()
```

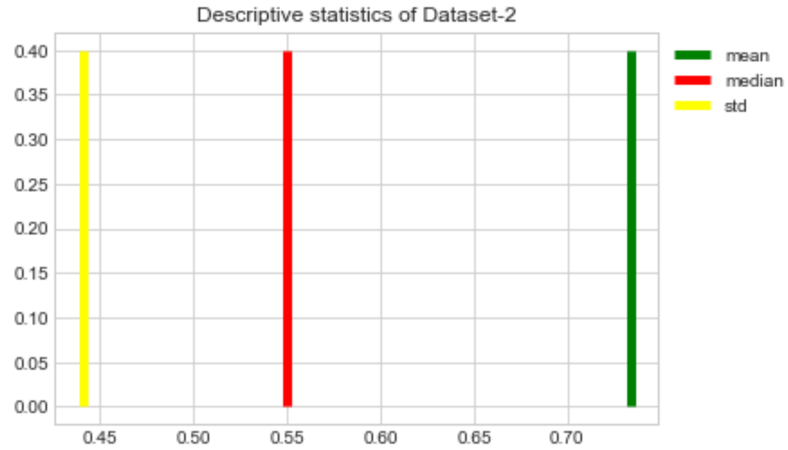
```
Out[99]: Deaths    0.549538
         dtype: float64
```

```
In [100]: dfUSCovid19Data.std()
```

```
Out[100]: Deaths    0.440583
          dtype: float64
```

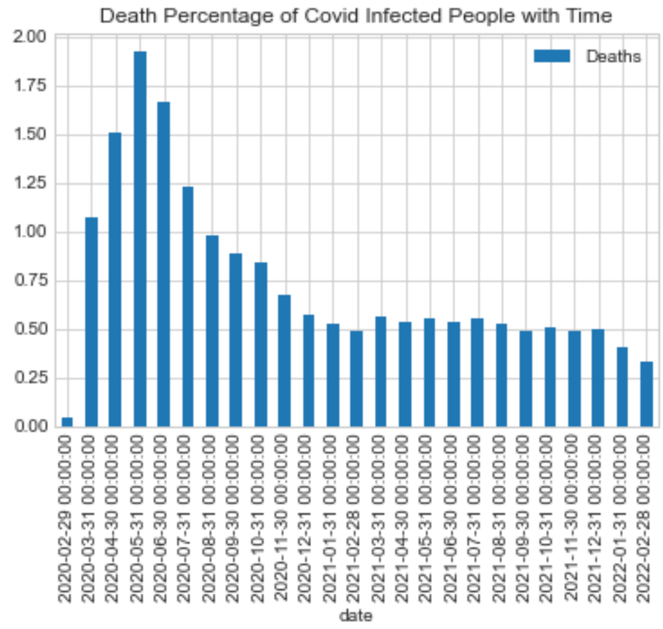
```
In [101]: arr1 = plt.vlines(dfUSCovid19Data.mean(), # Plot green line at mean
           ymin=0,
           ymax=0.4,
           linewidth=5.0,
           colors = "green"
           );
          arr2 = plt.vlines(dfUSCovid19Data.median(), # Plot red line at median
           ymin=0,
           ymax=0.4,
           linewidth=5.0,
           colors="red",
           );
          arr3 = plt.vlines(dfUSCovid19Data.std(), # Plot yellow line at std
           ymin=0,
           ymax=0.4,
           linewidth=5.0,
           colors="yellow"
           );
          # place legend outside
          #plt.legend(bbox_to_anchor=(1.0, 1), loc='upper left', title=('green - mean\nred -

          plt.legend([arr1, arr2, arr3], ['mean','median','std'],bbox_to_anchor=(1.0, 1),loc=
          plt.title('Descriptive statistics of Dataset-2')
          plt.show()
```



```
In [102]: plt.style.use('seaborn-whitegrid')
          dfUSCovid19Data.plot.bar()
          plt.title("Death Percentage of Covid Infected People with Time")

Out[102]: Text(0.5, 1.0, 'Death Percentage of Covid Infected People with Time')
```


Death Percentage of Covid Infected People with Time
Deaths
2020-02-29 00:00:00
2020-03-31 00:00:00
2020-04-30 00:00:00
2020-05-31 00:00:00
2020-06-30 00:00:00
2020-07-31 00:00:00
2020-08-31 00:00:00
2020-09-30 00:00:00
2020-10-31 00:00:00
2020-11-30 00:00:00
2020-12-31 00:00:00
2021-01-31 00:00:00
2021-02-28 00:00:00
2021-03-31 00:00:00
2021-04-30 00:00:00
2021-05-31 00:00:00
2021-06-30 00:00:00
2021-07-31 00:00:00
2021-08-31 00:00:00
2021-09-30 00:00:00
2021-10-31 00:00:00
2021-11-30 00:00:00
2021-12-31 00:00:00
2022-01-31 00:00:00
2022-02-28 00:00:00
date