CORONA_STAT_WORLD_FIXED_DATASET

May 4, 2020

1 TIME SERIES ANALYSIS ON THE COVID-19 (CORONAVIRUS) CASES AND DEATHS IN THE WORLD VIA A FIXED DATASET(1 JAN 2020-UP-TO-NOW)

[11]:
```
#importing relevant libraries
import pandas as pd
import numpy as nd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import requests
import datetime
from datetime import datetime, timedelta

#For using Google Drive path & Saving data to Google Drive
from google.colab import drive
drive.mount('drive')
```

[12]:
```
#This part is for daily update the dataset.
today = datetime.now()
today = today.strftime('%Y-%m-%d')

yesterday = datetime.now() - timedelta(days=1)
yesterday = yesterday.strftime('%Y-%m-%d')

try:
    url=f'https://www.ecdc.europa.eu/sites/default/files/documents/
 ↪COVID-19-geographic-disbtribution-worldwide-{today}.xlsx'
    df = pd.read_excel(url, parse_date=[0],index_col=0)#Parsing date is for␣
 ↪time series analysis
    df_domain = pd.DataFrame(df)

except:
    url=f'https://www.ecdc.europa.eu/sites/default/files/documents/
 ↪COVID-19-geographic-disbtribution-worldwide-{yesterday}.xlsx'
```

```
    df = pd.read_excel(url, parse_date=[0],index_col=0)#Parsing date is for␣
 ↪time series analysis
    df = pd.DataFrame(df)

#Data Wrangling
df.columns = ['Day','Month','Year','Cases','Deaths',␣
 ↪'Countries_and_territories','GeoId','Country_Code','Population','Continent']

df.drop('Day',1,inplace=True)
df.drop('Month',1,inplace=True)
df.drop('Year',1,inplace=True)
df.drop('Country_Code',1,inplace=True)
df.drop('Population',1,inplace=True)
#df = df[df['GeoId'] == 'NL']
df.drop('GeoId',1,inplace=True)
df.drop(df.tail(1).index,inplace=True) # drop last row 31-12-2019
#df.set_index(pd.DatetimeIndex(df['dateRep']), inplace=True)
#df.loc[:, 'Case_Fatality_Ratio'] =df['Deaths']/df.Cases
#df_NL=df

#Seave it to a csv file to a certain path
df.to_csv (r'C:/Users/lenovo/NLP_Homework/ECDC_data_world.csv',  index = True,␣
 ↪header=True)

df.head(10)
```

[12]:

dateRep	Cases	Deaths	Countries_and_territories	Continent
2020-05-04	235	13	Afghanistan	Asia
2020-05-03	134	4	Afghanistan	Asia
2020-05-02	164	4	Afghanistan	Asia
2020-05-01	222	4	Afghanistan	Asia
2020-04-30	122	0	Afghanistan	Asia
2020-04-29	124	3	Afghanistan	Asia
2020-04-28	172	0	Afghanistan	Asia
2020-04-27	68	10	Afghanistan	Asia
2020-04-26	112	4	Afghanistan	Asia
2020-04-25	70	1	Afghanistan	Asia

[13]:
```
df.tail(10)
```

[13]:

dateRep	Cases	Deaths	Countries_and_territories	Continent
2020-03-31	0	0	Zimbabwe	Africa
2020-03-30	0	0	Zimbabwe	Africa
2020-03-29	2	0	Zimbabwe	Africa
2020-03-28	2	0	Zimbabwe	Africa
2020-03-27	0	0	Zimbabwe	Africa

```
2020-03-26      1       0                   Zimbabwe    Africa
2020-03-25      0       0                   Zimbabwe    Africa
2020-03-24      0       1                   Zimbabwe    Africa
2020-03-23      0       0                   Zimbabwe    Africa
2020-03-22      1       0                   Zimbabwe    Africa
```

[14]:
```
print(df.isnull().sum().sort_values(ascending=False)) #Let's find out the␣
 ↪number of missing values in df and sort them down in descending order.
```

```
Continent                   0
Countries_and_territories   0
Deaths                      0
Cases                       0
dtype: int64
```

[15]:
```
type(df.index[0]) #Checking the dates whether they are timestamp or not.
print(df.index[0])
```

```
2020-05-04 00:00:00
```

[16]:
```
df.info() #general overview on dataset
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 14695 entries, 2020-05-04 to 2020-03-22
Data columns (total 4 columns):
Cases                       14695 non-null int64
Deaths                      14695 non-null int64
Countries_and_territories   14695 non-null object
Continent                   14695 non-null object
dtypes: int64(2), object(2)
memory usage: 574.0+ KB
```

[17]:
```
df.describe() #general descriptive statistics overview on dataset
```

[17]:

	Cases	Deaths
count	14695.000000	14695.000000
mean	233.996529	16.713848
std	1619.211535	124.539324
min	-1430.000000	0.000000
25%	0.000000	0.000000
50%	1.000000	0.000000
75%	30.000000	1.000000
max	48529.000000	4928.000000

[18]:
```
df_World=df[today] #A snapshot for current date of  the world
df_World
```

[18]:

dateRep	Cases	Deaths	Countries_and_territories	Continent
2020-05-04	235	13	Afghanistan	Asia
2020-05-04	6	0	Albania	Europe
2020-05-04	179	4	Algeria	Africa
2020-05-04	1	1	Andorra	Europe
2020-05-04	0	0	Angola	Africa
2020-05-04	0	0	Anguilla	America
2020-05-04	0	0	Antigua_and_Barbuda	America
2020-05-04	102	9	Argentina	America
2020-05-04	113	2	Armenia	Europe
2020-05-04	0	0	Aruba	America
2020-05-04	18	2	Australia	Oceania
2020-05-04	39	2	Austria	Europe
2020-05-04	38	0	Azerbaijan	Europe
2020-05-04	0	0	Bahamas	America
2020-05-04	99	0	Bahrain	Asia
2020-05-04	665	2	Bangladesh	Asia
2020-05-04	1	0	Barbados	America
2020-05-04	877	2	Belarus	Europe
2020-05-04	389	79	Belgium	Europe
2020-05-04	0	0	Belize	America
2020-05-04	0	0	Benin	Africa
2020-05-04	1	0	Bermuda	America
2020-05-04	0	0	Bhutan	Asia
2020-05-04	124	5	Bolivia	America
2020-05-04	0	0	Bonaire, Saint Eustatius and Saba	America
2020-05-04	18	5	Bosnia_and_Herzegovina	Europe
2020-05-04	0	0	Botswana	Africa
2020-05-04	4588	275	Brazil	America
2020-05-04	0	0	British_Virgin_Islands	America
2020-05-04	0	0	Brunei_Darussalam	Asia
...	...	...	...	...
2020-05-04	13	0	Sri_Lanka	Asia
2020-05-04	86	0	Sudan	Africa
2020-05-04	0	0	Suriname	America
2020-05-04	235	10	Sweden	Europe
2020-05-04	88	6	Switzerland	Europe
2020-05-04	0	0	Syria	Asia
2020-05-04	4	0	Taiwan	Asia
2020-05-04	44	2	Tajikistan	Asia
2020-05-04	3	0	Thailand	Asia
2020-05-04	0	0	Timor_Leste	Asia
2020-05-04	1	0	Togo	Africa
2020-05-04	0	0	Trinidad_and_Tobago	America
2020-05-04	4	0	Tunisia	Africa
2020-05-04	1670	61	Turkey	Asia

```
2020-05-04      0       0          Turks_and_Caicos_islands   America
2020-05-04      1       0                           Uganda    Africa
2020-05-04    502       9                          Ukraine    Europe
2020-05-04    564       7              United_Arab_Emirates      Asia
2020-05-04   4339     315                   United_Kingdom    Europe
2020-05-04      0       0       United_Republic_of_Tanzania    Africa
2020-05-04  24972    1297          United_States_of_America   America
2020-05-04      0       0      United_States_Virgin_Islands   America
2020-05-04      3       0                           Uruguay   America
2020-05-04     22       1                        Uzbekistan      Asia
2020-05-04     12       0                         Venezuela   America
2020-05-04      1       0                           Vietnam      Asia
2020-05-04      0       0                    Western_Sahara    Africa
2020-05-04      0       0                             Yemen      Asia
2020-05-04      5       0                            Zambia    Africa
2020-05-04      0       0                          Zimbabwe    Africa

[207 rows x 4 columns]
```

[64]:
```
total_cases_uptonow= df['Cases'].sum()
print(f"Total Covid-19 Cases in the World:                              ␣
 ↪{total_cases_uptonow}")
total_deaths_uptonow = df['Deaths'].sum()
print(f"Total Covid-19 Deaths in the World:                             ␣
 ↪{total_deaths_uptonow}")
```

```
Total Covid-19 Cases in the World:                              3438579
Total Covid-19 Deaths in the World:                             245610
```

2 SNAPSHOT ON JANUARY 2020

[70]:
```
df_jan2020=df['2020-01'] #A snapshot for the first month, January 2020
```

[20]:
```
total_cases_jan2020 = df_jan2020['Cases'].sum()
print(f"Total Covid-19 cases in January in the World:                   ␣
 ↪ {total_cases_jan2020}")
total_deaths_jan2020 = df_jan2020['Deaths'].sum()
print(f"Total deaths due to Covid-19 cases in January in the World:     ␣
 ↪ {total_deaths_jan2020}")
```

```
Total Covid-19 cases in January in the World:                   9799
Total deaths due to Covid-19 cases in January in the World:     213
```

[21]:
```
avg_cases_jan2020 = df_jan2020['Cases'].mean()
print(f"Daily average Covid-19 cases in January in the World:           ␣
 ↪ {avg_cases_jan2020}")
avg_death_jan2020 = df_jan2020['Deaths'].mean()
```

```
print(f"Daily average deaths due to Covid-19 cases in January in the World:
 ↪ {avg_death_jan2020}")
```

```
Daily average Covid-19 cases in January in the World:
4.717862301396244
Daily average deaths due to Covid-19 cases in January in the World:
0.10255175734232065
```

3 SNAPSHOT ON FEBRUARY 2020

[71]:
```
df_feb2020=df['2020-02'] #A snapshot for the second month, February 2020
```

[23]:
```
total_cases_feb2020 = df_feb2020['Cases'].sum()
print(f"Total Covid-19 cases in February in the World:
 ↪ {total_cases_feb2020}")
total_deaths_feb2020 = df_feb2020['Deaths'].sum()
print(f"Total deaths due to Covid-19 cases in February in the World:
 ↪  {total_deaths_feb2020}")
```

```
Total Covid-19 cases in February in the World:                    75377
Total deaths due to Covid-19 cases in February in the World:       2708
```

[24]:
```
avg_cases_feb2020=df_feb2020['Cases'].mean()
print(f"Daily average Covid-19 cases in February in the World:
 ↪  {avg_cases_feb2020}")
avg_death_feb2020=df_feb2020['Deaths'].mean()
print(f"Daily average deaths due to Covid-19 cases in February in the World:
 ↪  {avg_death_feb2020}")
```

```
Daily average Covid-19 cases in February in the World:
38.79413278435409
Daily average deaths due to Covid-19 cases in February in the World:
1.3937210499227999
```

4 SNAPSHOT ON MARCH 2020

[72]:
```
df_mar2020=df['2020-03'] #A snapshot for the third month, March 2020
```

[26]:
```
total_cases_mar2020 = df_mar2020['Cases'].sum()
print(f"Total Covid-19 cases in March in the World:
↪{total_cases_mar2020}")
total_deaths_mar2020 = df_mar2020['Deaths'].sum()
print(f"Total deaths due to Covid-19 cases in March in the World:
↪{total_deaths_mar2020}")
```

```
Total Covid-19 cases in March in the World:                    663242
Total deaths due to Covid-19 cases in March in the World:       33621
```

[27]:
```
avg_cases_mar2020=df_mar2020['Cases'].mean()
print(f"Daily average Covid-19 cases in March in the World:
 ↪{avg_cases_mar2020}")
avg_death_mar2020=df_mar2020['Deaths'].mean()
print(f"Daily average deaths due to Covid-19 cases in March in the World:
 ↪{avg_death_mar2020}")
```

```
Daily average Covid-19 cases in March in the World:
181.56090884204764
Daily average deaths due to Covid-19 cases in March in the World:
9.203668217903093
```

5 SNAPSHOT ON APRIL 2020

[73]:
```
df_april2020=df['2020-04'] #A snapshot for the forth month, April 2020
```

[55]:
```
total_cases_april2020 = df_april2020['Cases'].sum()
print(f"Total Covid-19 cases in April in the World:
 ↪{total_cases_april2020}")
total_deaths_april2020 = df_april2020['Deaths'].sum()
print(f"Total deaths due to Covid-19 cases in April in the World:
 ↪{total_deaths_april2020}")
```

```
Total Covid-19 cases in April in the World:                    2354115
Total deaths due to Covid-19 cases in April in the World:       189408
```

[56]:
```
avg_cases_april2020=df_april2020['Cases'].mean()
print(f"Daily average Covid-19 cases in April in the World:
 ↪{avg_cases_april2020}")
avg_death_april2020=df_april2020['Deaths'].mean()
print(f"Daily average deaths due to Covid-19 cases in April in the World:
 ↪{avg_death_april2020}")
```

```
Daily average Covid-19 cases in April in the World:
384.34530612244896
Daily average deaths due to Covid-19 cases in April in the World:
30.923755102040815
```

6 Plots and statistics with Time Series Analysis

7 WEEKLY(1JAN-Up to Date)

[69]:
```
df.Deaths.resample("D").mean() #Weekly average of the Covid-19 death in 2020
```

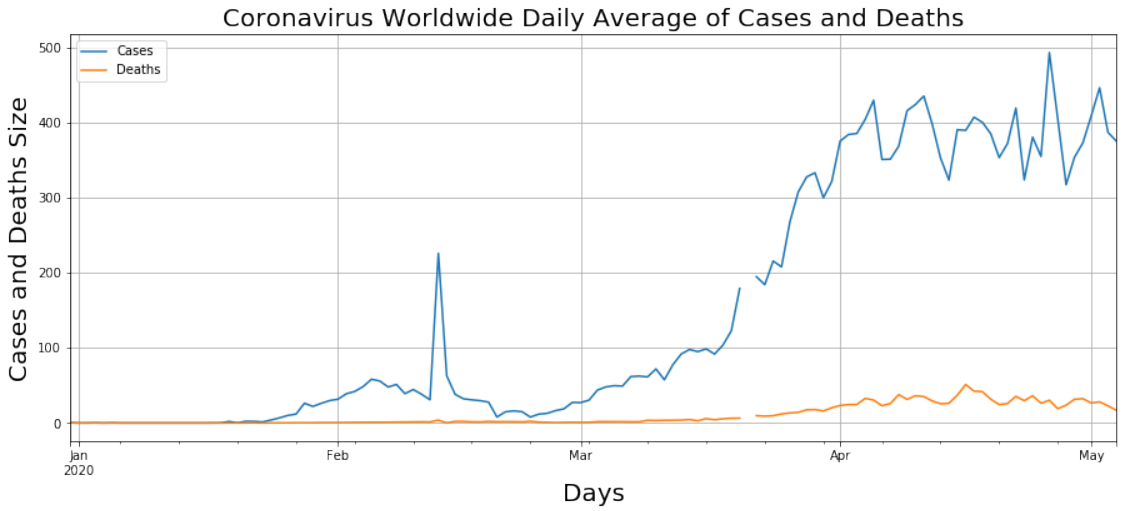
[69]:
```
dateRep
2019-12-31     0.000000
2020-01-01     0.000000
2020-01-02     0.000000
2020-01-03     0.000000
2020-01-04     0.000000
2020-01-05     0.000000
2020-01-06     0.000000
2020-01-07     0.000000
2020-01-08     0.000000
2020-01-09     0.000000
2020-01-10     0.000000
2020-01-11     0.014925
2020-01-12     0.000000
2020-01-13     0.000000
2020-01-14     0.000000
2020-01-15     0.014925
2020-01-16     0.000000
2020-01-17     0.000000
2020-01-18     0.000000
2020-01-19     0.014925
2020-01-20     0.000000
2020-01-21     0.044776
2020-01-22     0.164179
2020-01-23     0.000000
2020-01-24     0.134328
2020-01-25     0.223881
2020-01-26     0.223881
2020-01-27     0.373134
2020-01-28     0.373134
2020-01-29     0.388060
                  ...
2020-04-05    30.572139
2020-04-06    23.044335
2020-04-07    25.940887
2020-04-08    37.748768
2020-04-09    31.549020
2020-04-10    36.317073
2020-04-11    35.224390
2020-04-12    29.395122
2020-04-13    25.687805
2020-04-14    26.200000
2020-04-15    37.092683
2020-04-16    51.317073
2020-04-17    42.482927
2020-04-18    41.795122
2020-04-19    31.673171
```

```
2020-04-20    24.565854
2020-04-21    26.029268
2020-04-22    35.531707
2020-04-23    29.760976
2020-04-24    36.312195
2020-04-25    26.370732
2020-04-26    30.334951
2020-04-27    19.058252
2020-04-28    23.771845
2020-04-29    31.616505
2020-04-30    32.427184
2020-05-01    26.724638
2020-05-02    28.153846
2020-05-03    22.927885
2020-05-04    16.922705
Freq: D, Name: Deaths, Length: 126, dtype: float64
```

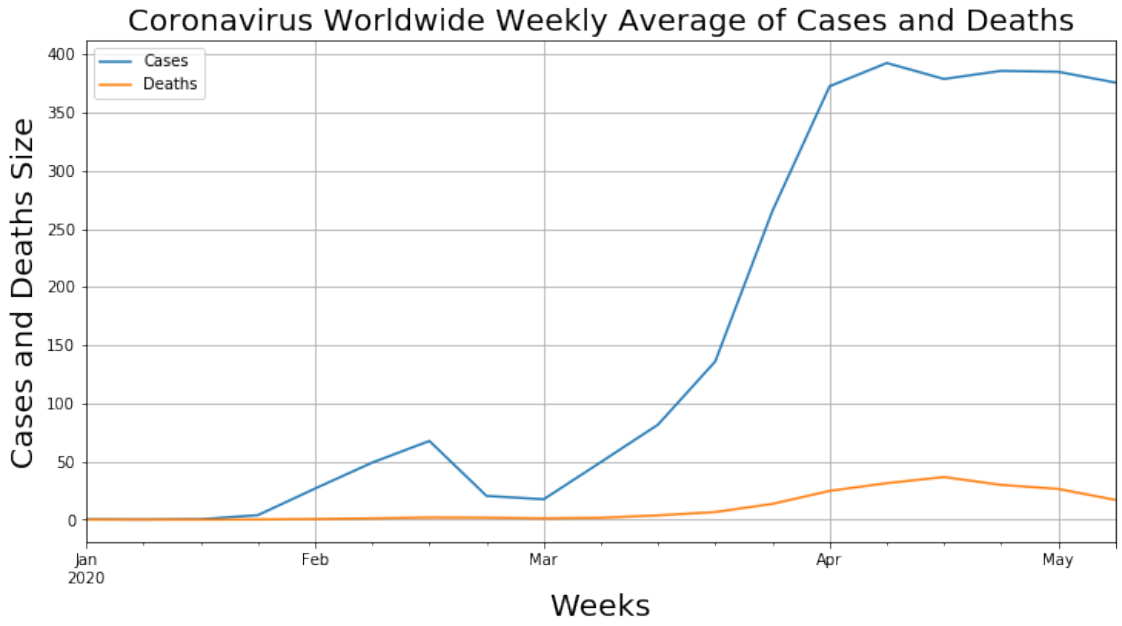
[33]:
```
df.Deaths.resample("W").mean() #Weekly average of the Covid-19 death in 2020
```

[33]:
```
dateRep
2020-01-05     0.000000
2020-01-12     0.002132
2020-01-19     0.004264
2020-01-26     0.113006
2020-02-02     0.530917
2020-02-09     1.083156
2020-02-16     1.825160
2020-02-23     1.692964
2020-03-01     1.100213
2020-03-08     1.594737
2020-03-15     3.644407
2020-03-22     6.512849
2020-03-29    13.526038
2020-04-05    24.688985
2020-04-12    31.327031
2020-04-19    36.606969
2020-04-26    29.844011
2020-05-03    26.380788
2020-05-10    16.922705
Freq: W-SUN, Name: Deaths, dtype: float64
```

[34]:
```
df[['Cases','Deaths']].resample('D').mean().plot(figsize=(15,6), grid=True,)␣
 ↪#Line plot Daily both cases and deaths due to Covid-19
plt.title('Coronavirus Worldwide Daily Average of Cases and Deaths', size=20);
plt.xlabel('Days', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```



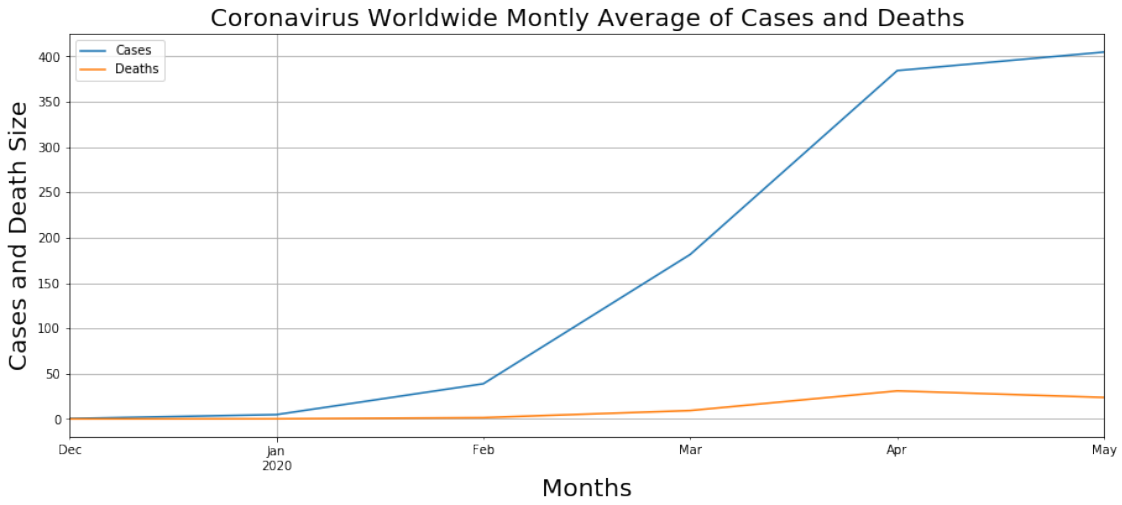
[35]:
```
df[['Cases','Deaths']].resample("W").mean().plot(figsize=(12, 6),␣
 ↪grid=True,)#Line plot weekly both cases and deaths due to Covid-19
plt.title('Coronavirus Worldwide Weekly Average of Cases and Deaths', size=20);
plt.xlabel('Weeks', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```



[36]:
```
df[['Cases','Deaths']].resample('M').mean().plot(figsize=(15,6), grid=True,)␣
 ↪#Line plot Montly both cases and deaths due to Covid-19
plt.title('Coronavirus Worldwide Montly Average of Cases and Deaths', size=20);
plt.xlabel('Months', size=20)
```

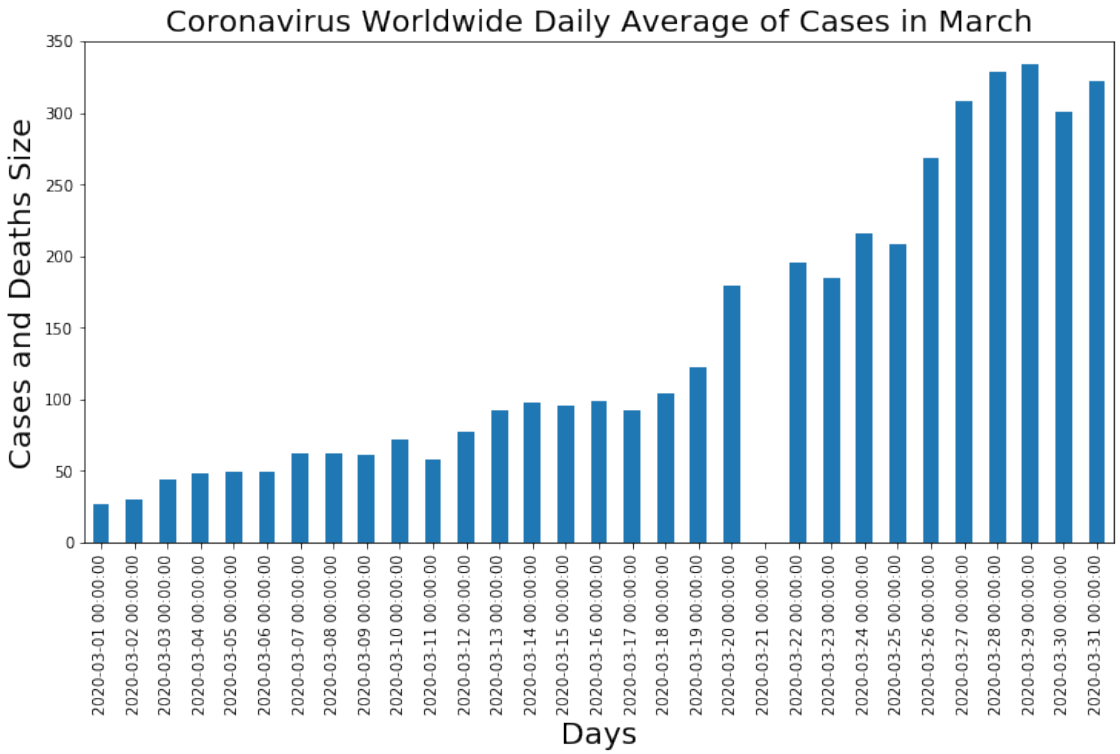
```
plt.ylabel('Cases and Death Size', size=20)
```

[36]:
```
Text(0, 0.5, 'Cases and Death Size')
```

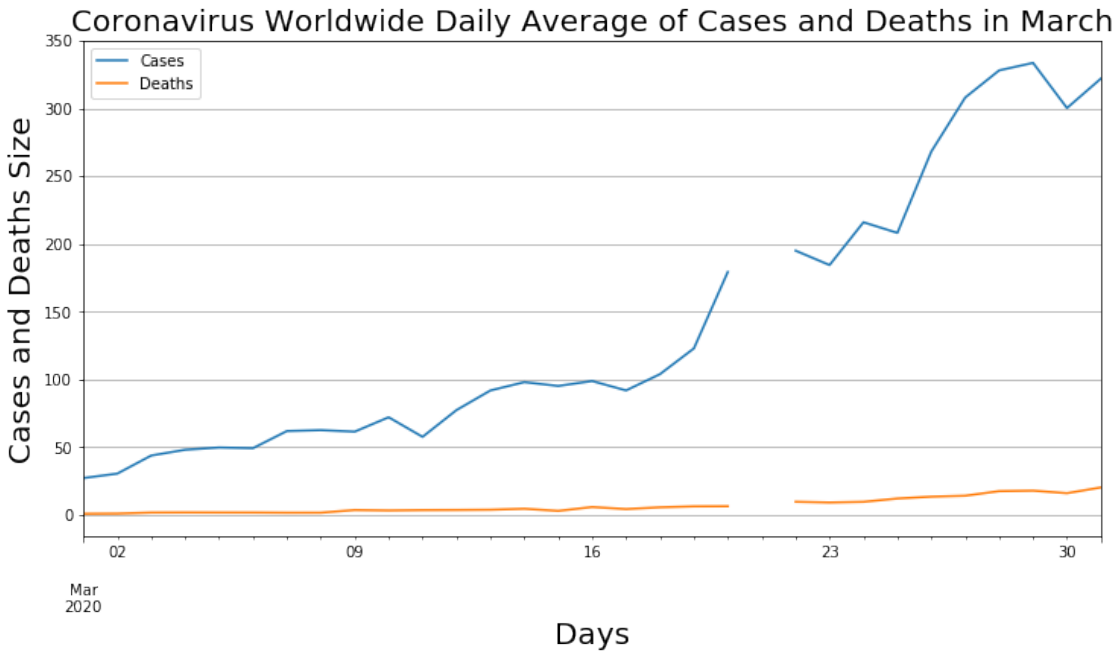


8 March Analysis

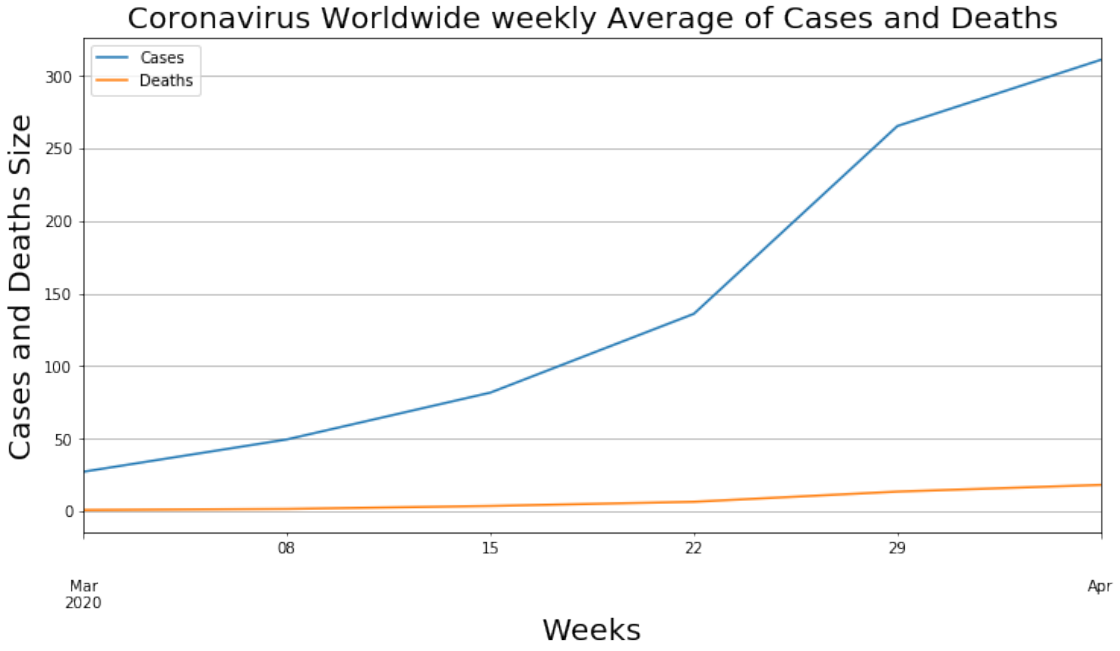
[37]:
```
df['2020-03'].Cases.resample('D').mean().plot(kind='bar',figsize=(12, 6))␣
 ↪#Daily basis just cases in bar plot in March 2020
plt.title('Coronavirus Worldwide Daily Average of Cases in March', size=20)
plt.xlabel('Days', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```



[38]:
```
df['2020-03'].resample('D').mean().plot(figsize=(12, 6), grid=True,)#Daily␣
 ↪basis both cases and deaths in bar plot in March 2020
plt.title('Coronavirus Worldwide Daily Average of Cases and Deaths in March',␣
 ↪size=20);
plt.xlabel('Days', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```

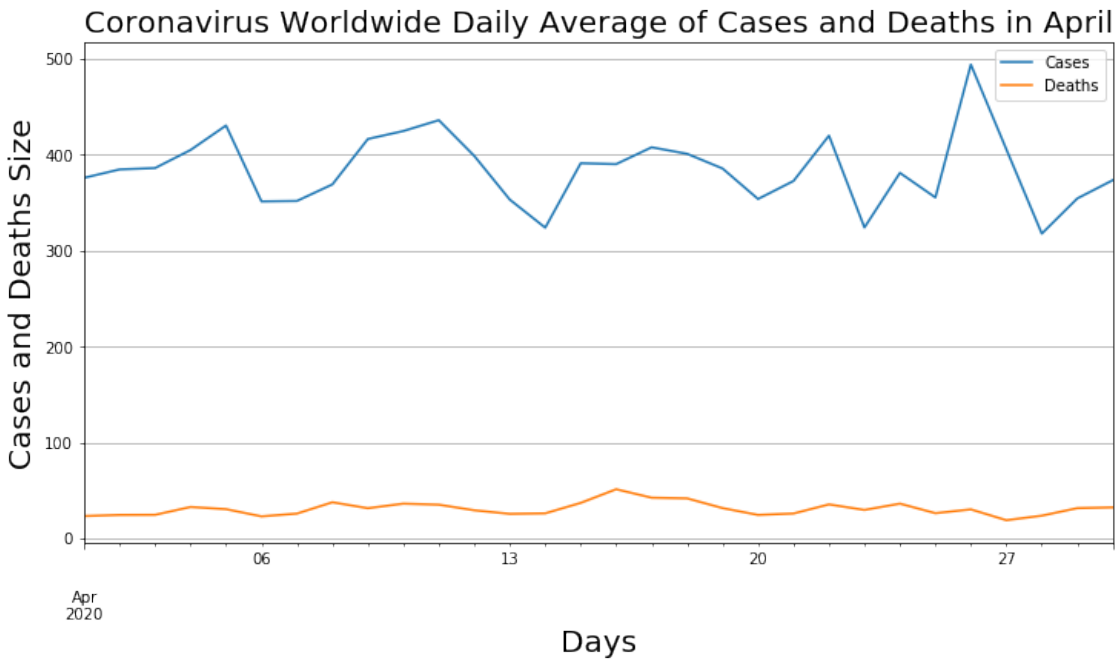


[39]:
```
df['2020-03'].resample('W').mean().plot(figsize=(12, 6), grid=True,) #Weekly␣
 ↪basis both cases and deaths in line plot in March 2020
plt.title('Coronavirus Worldwide weekly Average of Cases and Deaths', size=20)
plt.xlabel('Weeks', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```

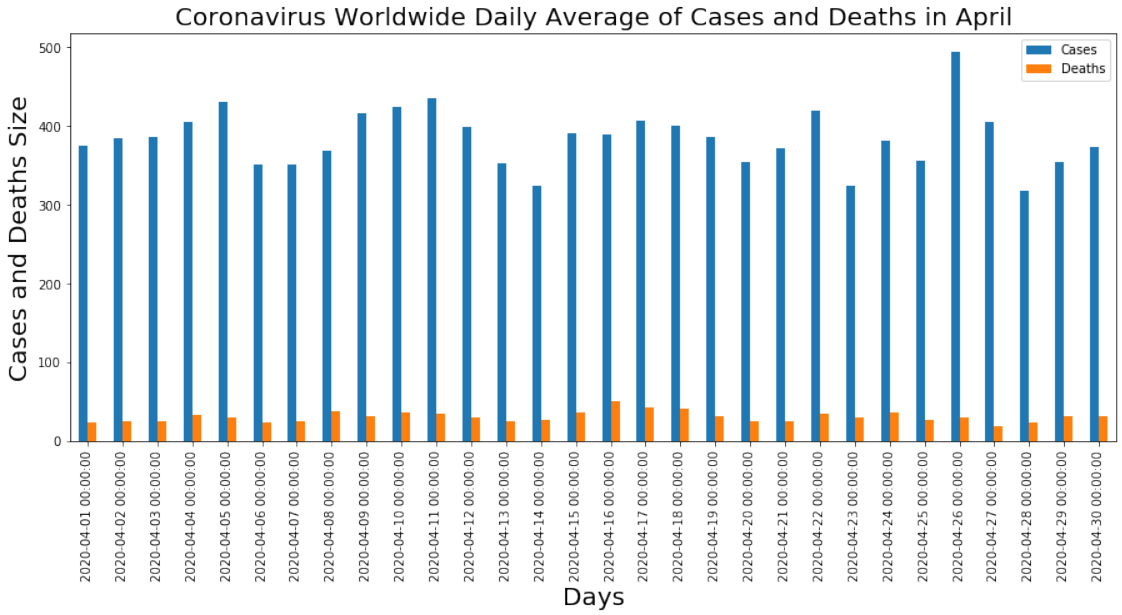


9 April Analysis

[40]:
```
df['2020-04'].resample('D').mean().plot(figsize=(12, 6), grid=True,) ##Daily␣
 ↪basis cases and deaths in line plot in April 2020
plt.title('Coronavirus Worldwide Daily Average of Cases and Deaths in April',␣
 ↪size=20)
plt.xlabel('Days', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```

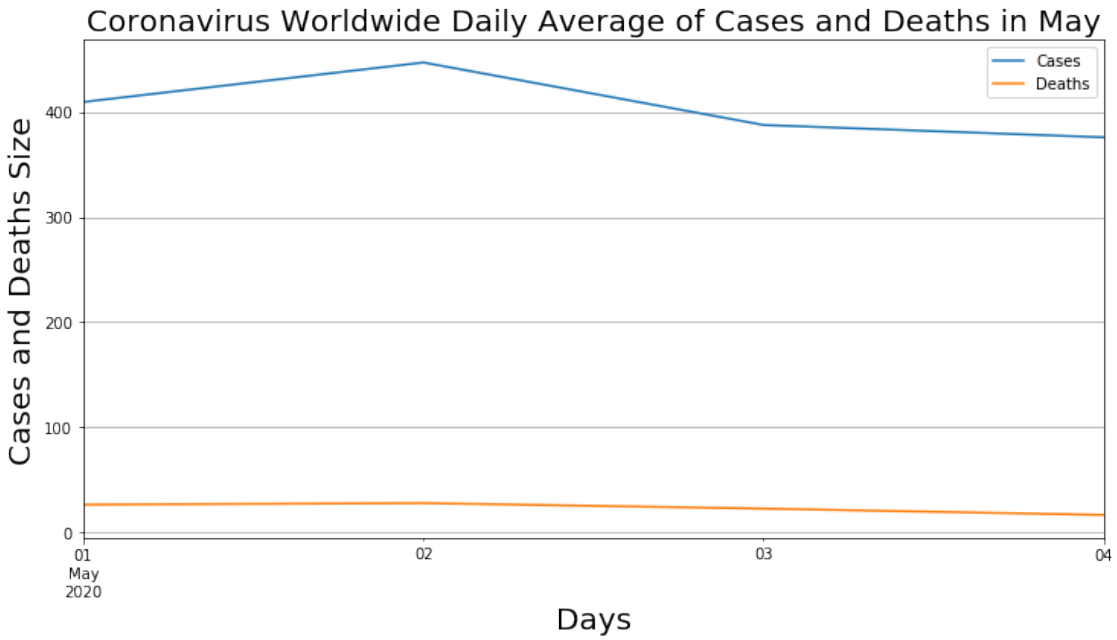


[41]:
```
df['2020-04'].resample('D').mean().plot(kind='bar',figsize=(15, 6)) #Daily␣
 ↪basis cases and deaths in bar plot in April 2020
plt.title('Coronavirus Worldwide Daily Average of Cases and Deaths in April',␣
 ↪size=20)
plt.xlabel('Days', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```

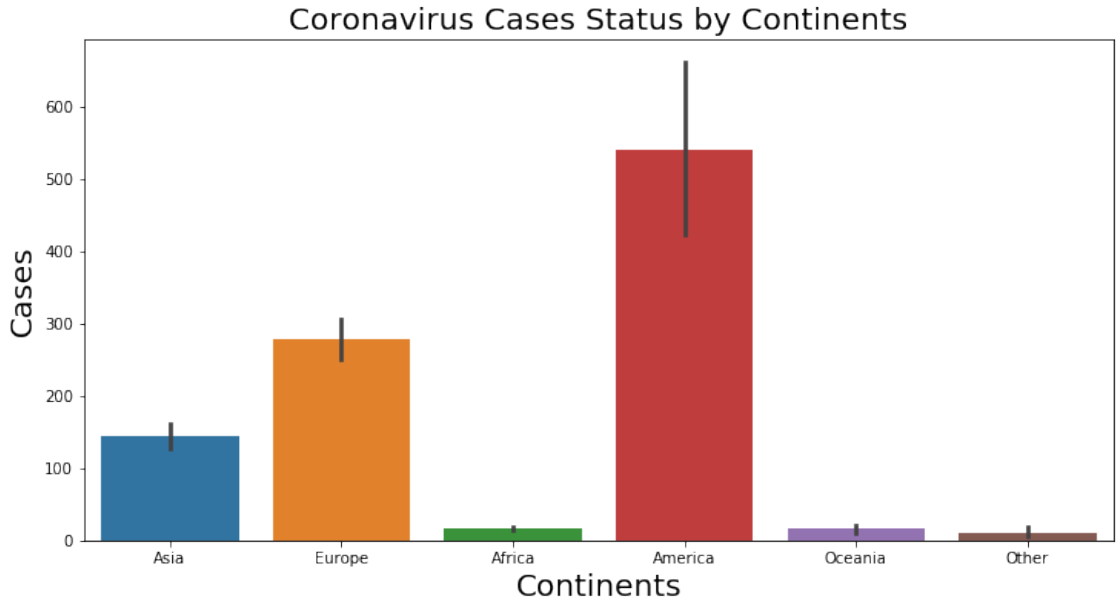


10 May Analysis

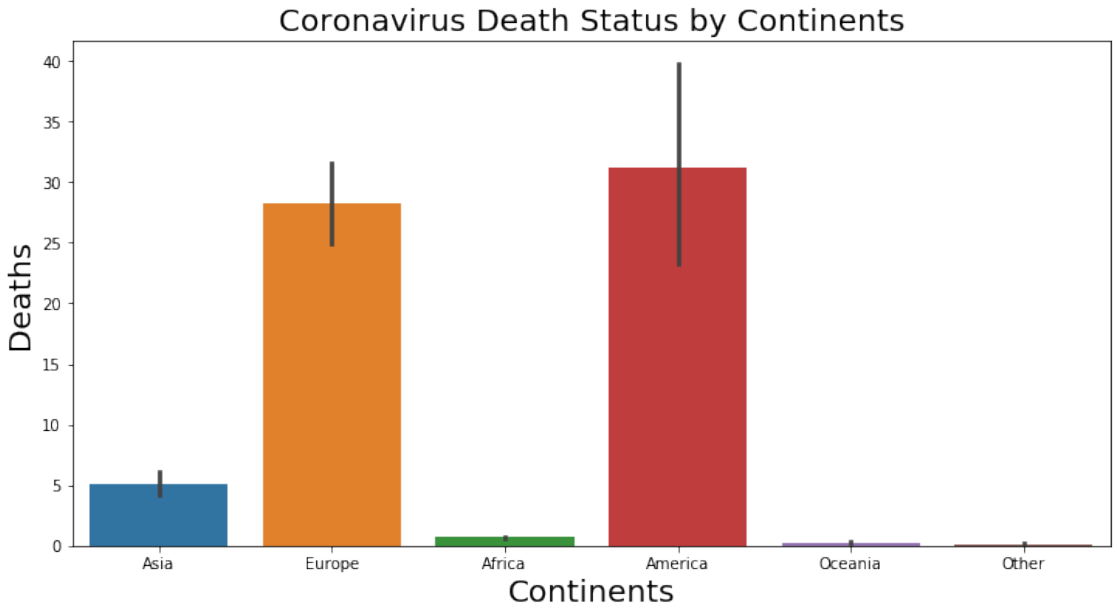
[42]:
```
df['2020-05'].resample('D').mean().plot(figsize=(12, 6), grid=True,) ##Daily␣
 ↪basis cases and deaths in line plot in May 2020
plt.title('Coronavirus Worldwide Daily Average of Cases and Deaths in May',␣
 ↪size=20)
plt.xlabel('Days', size=20)
plt.ylabel('Cases and Deaths Size', size=20);
```



[43]:
```
plt.figure(figsize=(12,6))
sns.barplot(x='Continent',y="Cases",data=df);
plt.title('Coronavirus Cases Status by Continents', size=20);
plt.xlabel('Continents', size=20)
plt.ylabel('Cases', size=20);
```



[44]:
```
plt.figure(figsize=(12,6))
sns.barplot(x='Continent',y="Deaths",data=df);
plt.title('Coronavirus Death Status by Continents', size=20);
plt.xlabel('Continents', size=20)
plt.ylabel('Deaths', size=20);
```



[45]:
```
%%html
<iframe src= "https://who.maps.arcgis.com/apps/opsdashboard/index.html#/
 ↪ead3c6475654481ca51c248d52ab9c61?:display_count=y&:origin=viz_share_link"␣
 ↪width="1400" height="1000">
</iframe>
```

```
<IPython.core.display.HTML object>
```