Excel and CSV file Reading

In [7]:

```
import pandas as pd
```

In [31]:

```
data=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\database-A section_2020_Assignment a
# for reading Csv file just replace the read_excel with read_csv
```

In [9]:

```
data[16:17]
```

Out[9]:

	sl_no	Rg_no	student_name	Total	Total (roundup)
16	17.0	17ETCS002004	ABHILASH RAMU	7.0	7.0

In [10]:

```
 data[69:70]
```

Out[10]:

	sl_no	Rg_no	student_name	Total	Total (roundup)
69	70.0	17ETCS002066	DINESH N B	6.5	7.0

In [11]:

```
sum1=data["Total"].sum()
print("sum is :",sum1)
```

```
sum is : 486.5
```

In [12]:

```
sum2=data["Total (roundup)"].sum()
print("sum is : ",sum2)
```

```
sum is :  508.0
```

In [13]:

```
data.shape
```

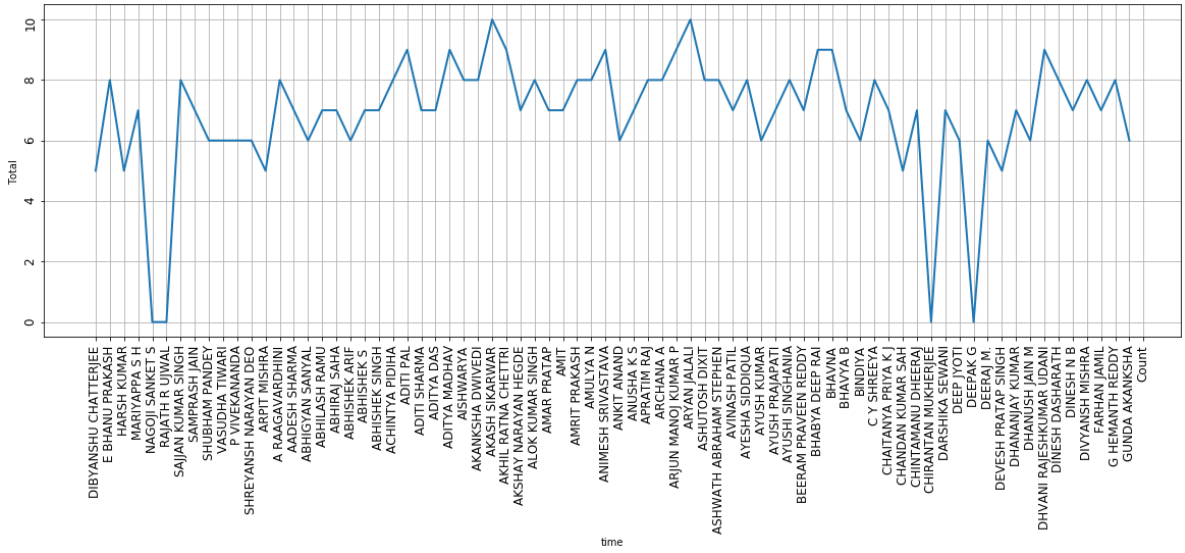
Out[13]:

```
(75, 5)
```

Plotting Exam result Data

In [19]:

```
import matplotlib.pyplot as plt
fig=plt.figure()
plt.xlabel("time")
plt.ylabel("Total")
plt.plot(data["student_name"],data["Total (roundup)"],linewidth=2)
plt.tick_params(labelrotation=90,labelsize=12)
plt.grid()
plt.rcParams["figure.figsize"]=(20,6)
fig.savefig(r"C:\Users\Lenovo\OneDrive\Pictures\saved_figure\Data_base_A_section_.png"
```



In [20]:

```
import matplotlib.pyplot as plt
```

In [21]:

```
wind=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Project_V1.
wind.head(10)
```

Out[21]:

	Days	(Multiple Items)	Unnamed: 2	Unnamed: 3	Unnamed: 4
0	NaN	NaN	NaN	NaN	NaN
1	Row Labels	Average of Wind (MW)	Average of Solar (MW)	Average of Battery (MW)	Average of Target (MW)
2	12 AM	44.799	0	15.201	30
3	1 AM	41.4458	0	18.5542	30
4	2 AM	27.127	0	32.873	30
5	3 AM	37.1483	0	22.8517	30
6	4 AM	32.8214	0	27.1786	30
7	5 AM	27.0539	0	32.9461	30
8	6 AM	31.0651	0.258696	28.6762	30
9	7 AM	31.4016	6.85506	21.7434	30

In [32]:

```
wind1=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Project_V1
wind2=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Project_V1
wind3=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Project_V1
wind4=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Project_V1
wind5=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Project_V1
wind6=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Project_V1
data3=wind3[2:15]
data3
```

Out[32]:

	Days	(Multiple Items)	Unnamed: 2	Unnamed: 3	Unnamed: 4
2	12 AM	44.799	0	15.201	30
3	1 AM	41.4458	0	18.5542	30
4	2 AM	27.127	0	32.873	30
5	3 AM	37.1483	0	22.8517	30
6	4 AM	32.8214	0	27.1786	30
7	5 AM	27.0539	0	32.9461	30
8	6 AM	31.0651	0.258696	28.6762	30
9	7 AM	31.4016	6.85506	21.7434	30
10	8 AM	28.8227	19.6763	11.501	30
11	9 AM	24.6572	31.6208	3.72206	30
12	10 AM	21.9488	40.0881	-2.03696	30
13	11 AM	20.4033	44.3137	-4.71692	30
14	12 PM	16.6897	44.9787	-1.66846	30

Removing Unwanted Cell or Empty Cell and Renameing them

In [25]:

```
data3.rename(columns={'Days':'Row Labels','(Multiple Items)':'Average of Wind (MW)',
                      'Unnamed: 2':'Average of Solar (MW)',
                      'Unnamed: 3':'Average of Battery (MW)','Unnamed: 4':'Average of
data3.filter(regex='\d')
data3
```

```
C:\Users\Lenovo\Anaconda3\lib\site-packages\pandas\core\frame.py:4025: Setti
ngWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/s
table/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pand
as-docs/stable/indexing.html#indexing-view-versus-copy)
  return super(DataFrame, self).rename(**kwargs)
```

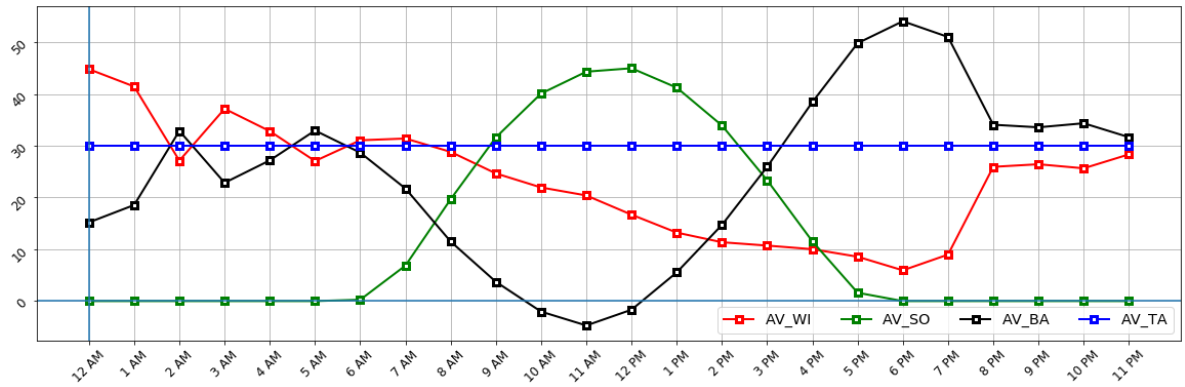
Out[25]:

	Row Labels	Average of Wind (MW)	Average of Solar (MW)	Average of Battery (MW)	Average of Target (MW)
2	12 AM	44.799	0	15.201	30
3	1 AM	41.4458	0	18.5542	30
4	2 AM	27.127	0	32.873	30
5	3 AM	37.1483	0	22.8517	30
6	4 AM	32.8214	0	27.1786	30
7	5 AM	27.0539	0	32.9461	30
8	6 AM	31.0651	0.258696	28.6762	30
9	7 AM	31.4016	6.85506	21.7434	30
10	8 AM	28.8227	19.6763	11.501	30
11	9 AM	24.6572	31.6208	3.72206	30
12	10 AM	21.9488	40.0881	-2.03696	30
13	11 AM	20.4033	44.3137	-4.71692	30
14	12 PM	16.6897	44.9787	-1.66846	30
15	1 PM	13.2276	41.2309	5.54146	30
16	2 PM	11.3634	33.9169	14.7196	30
17	3 PM	10.7191	23.3358	25.9451	30
18	4 PM	10.023	11.5312	38.4457	30
19	5 PM	8.57096	1.61023	49.8188	30
20	6 PM	5.94049	0	54.0595	30
21	7 PM	8.97982	0	51.0202	30
22	8 PM	25.9332	0.00158492	34.0652	30
23	9 PM	26.4326	0.000855021	33.5666	30
24	10 PM	25.6462	0.00252335	34.3513	30
25	11 PM	28.3387	0	31.6613	30

Analysing the Real time Data

In [26]:

```
fig=plt.figure()
plt.plot(data3['Row Labels'],data3['Average of Wind (MW)'],'s-',color='r',linewidth=2,
         markersize=6,markerfacecolor='white',markeredgecolor='r',markeredgewidth=3)
plt.plot(data3['Row Labels'],data3['Average of Solar (MW)'],'s-',color='g',linewidth=2
         markersize=6,markerfacecolor='white',markeredgecolor='g',markeredgewidth=3)
plt.plot(data3['Row Labels'],data3['Average of Battery (MW)'],'s-',color='k',linewidth
         markersize=6,markerfacecolor='white',markeredgecolor='k',markeredgewidth=3)
plt.plot(data3['Row Labels'],data3['Average of Target (MW)'],'s-',color='b',linewidth=
         markersize=6,markerfacecolor='white',markeredgecolor='b',markeredgewidth=3)
plt.legend(loc='best',frameon='true',ncol=4,fontsize='x-large')
plt.grid()
plt.axhline()
plt.axvline()
plt.tick_params(labelrotation=45,labelsize=12)
fig.savefig(r"C:\Users\Lenovo\OneDrive\Pictures\saved_figure\Wind_data1.png")
```



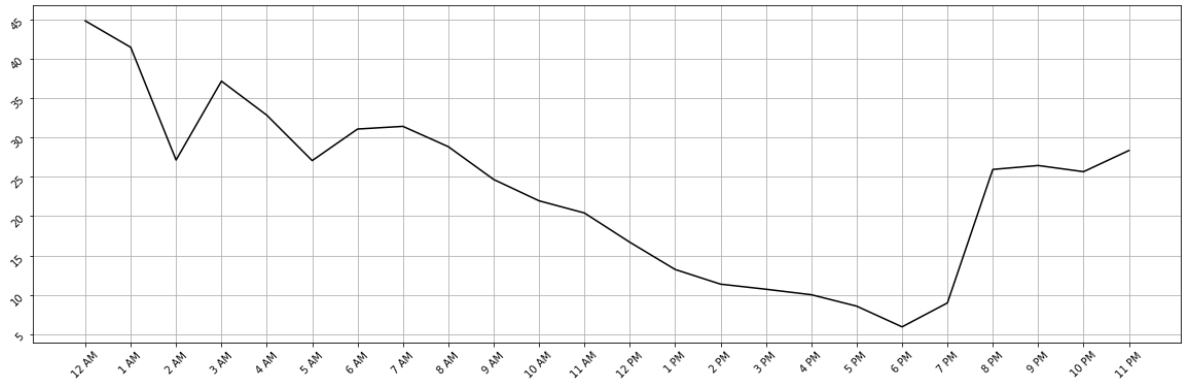
In [27]:

```
import sqlite3
from sqlalchemy import create_engine
import pandas as pd
import matplotlib.pyplot as plt
```

In [28]:

```
entry=int(input("Enter the entry:"))
wind_data=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Projec
wind_data.columns=wind_data.iloc[1]
wind_data.drop([0,1,26],axis=0,inplace=True)
plt.plot(wind_data['Row Labels'],wind_data['Average of Wind (MW)'],'-k')
plt.rcParams['figure.figsize']=(12,6)
plt.tick_params(labelrotation=45)
plt.grid()
```

```
Enter the entry:3
```



Extracting YEAR, MONTH, DAY, HOUR, MINUTES, SECONDS from the Date and Time

In [30]:

```
wind_data=pd.read_excel(r"C:\Users\Lenovo\Desktop\data\Wind _ Solar _ Battery _ Projec
wind_data.drop({wind_data.columns[i] for i in range(11,15)},axis=1,inplace=True)
wind_data.drop({i for i in range(4464,4467)},axis=0,inplace=True)
wind_data.insert(1,'Year',pd.DatetimeIndex(wind_data['Date']).year)
wind_data.insert(3,'Day',pd.DatetimeIndex(wind_data['Date']).day)
wind_data.insert(4,'Time',pd.DatetimeIndex(wind_data['Date']).time)
wind_data.insert(5,'Hour',pd.DatetimeIndex(wind_data['Date']).hour)
wind_data.insert(6,'minute',pd.DatetimeIndex(wind_data['Date']).minute)
wind_data.insert(7,'second',pd.DatetimeIndex(wind_data['Date']).second)
wind_data.head(20)
#wind_data_transposed = wind_data.T
#wind_data_transposed


```

Out[30]:

	Date	Year	Month	Day	Time	Hour	minute	second	AP	GJ	KA	MH	MP	R.
0	2019-01-01 00:00:00	2019	January	1	00:00:00	0	0	0	3.5	1.2	9.6	0.0	6.3	5.(
1	2019-01-01 00:10:00	2019	January	1	00:10:00	0	10	0	3.5	1.4	12.1	2.3	7.4	5.(
2	2019-01-01 00:20:00	2019	January	1	00:20:00	0	20	0	3.4	1.6	13.7	7.9	7.2	5.7
3	2019-01-01 00:30:00	2019	January	1	00:30:00	0	30	0	4.0	2.2	15.4	9.9	7.9	5.(
4	2019-01-01 00:40:00	2019	January	1	00:40:00	0	40	0	4.4	3.3	15.7	11.6	8	5.(
5	2019-01-01 00:50:00	2019	January	1	00:50:00	0	50	0	4.3	5.8	15.1	9.9	7.6	5.(
6	2019-01-01 01:00:00	2019	January	1	01:00:00	1	0	0	4.2	5.6	14.2	7.9	8.1	4.2
7	2019-01-01 01:10:00	2019	January	1	01:10:00	1	10	0	5.1	5.5	13.1	2.0	8.3	3.4
8	2019-01-01 01:20:00	2019	January	1	01:20:00	1	20	0	4.5	4.6	11.1	0.4	8.1	3.(
9	2019-01-01 01:30:00	2019	January	1	01:30:00	1	30	0	5.0	3.7	9.8	0.4	7.9	3.5
10	2019-01-01 01:40:00	2019	January	1	01:40:00	1	40	0	5.3	4.0	9.4	0.0	9.1	3.7
11	2019-01-01 01:50:00	2019	January	1	01:50:00	1	50	0	5.0	4.7	8.5	0.0	9.5	5.(

	Date	Year	Month	Day	Time	Hour	minute	second	AP	GJ	KA	MH	MP	R.
12	2019-01-01 02:00:00	2019	January	1	02:00:00	2	0	0	3.8	4.4	7.7	0.0	9	5.6
13	2019-01-01 02:10:00	2019	January	1	02:10:00	2	10	0	3.9	4.7	5.8	0.0	8.5	5.1
14	2019-01-01 02:20:00	2019	January	1	02:20:00	2	20	0	3.9	5.3	6.7	0.0	8.1	4.9
15	2019-01-01 02:30:00	2019	January	1	02:30:00	2	30	0	3.4	5.5	6.9	0.0	7.5	4.7
16	2019-01-01 02:40:00	2019	January	1	02:40:00	2	40	0	3.4	5.8	5.9	0.0	7.5	4.6
17	2019-01-01 02:50:00	2019	January	1	02:50:00	2	50	0	3.5	5.9	7.1	0.0	8.4	4.5
18	2019-01-01 03:00:00	2019	January	1	03:00:00	3	0	0	4.6	5.6	8.4	0.0	8.8	5.2
19	2019-01-01 03:10:00	2019	January	1	03:10:00	3	10	0	3.9	5.3	9.8	0.0	9.7	5.0

Analysing Covid -19 Report in different Country

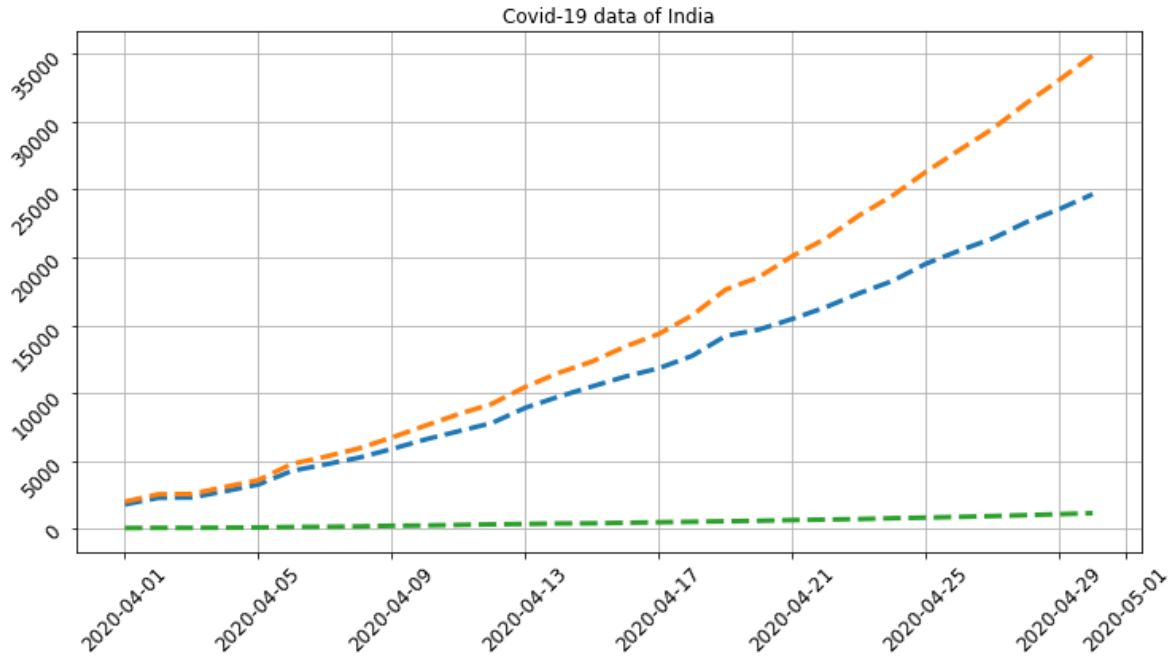
In [33]:

```
1 import pandas as pd
2 import matplotlib.pyplot as plt
```

In [34]:

```
fig=plt.figure()
plt.style
country=input("Enter the country ->")
data=pd.read_json('https://api.covid19api.com/dayone/country/'+country)
data_1=data.set_index('Date')
Date_info=input('Enter the Date -> ')
data_2=data_1[Date_info]
plt.tick_params(labelrotation=45,labelsize=12)
plt.rcParams["figure.figsize"]=(15,8)
plt.grid()
plt.title("Covid-19 data of India")
plt.plot(data_2['Active'],'--',linewidth=3)
plt.plot(data_2['Confirmed'],'--',linewidth=3)
plt.plot(data_2['Deaths'],'--',linewidth=3)
plt.savefig(r'C:\Users\Lenovo\OneDrive\Pictures\saved_figure\covid-19_data_india')
```

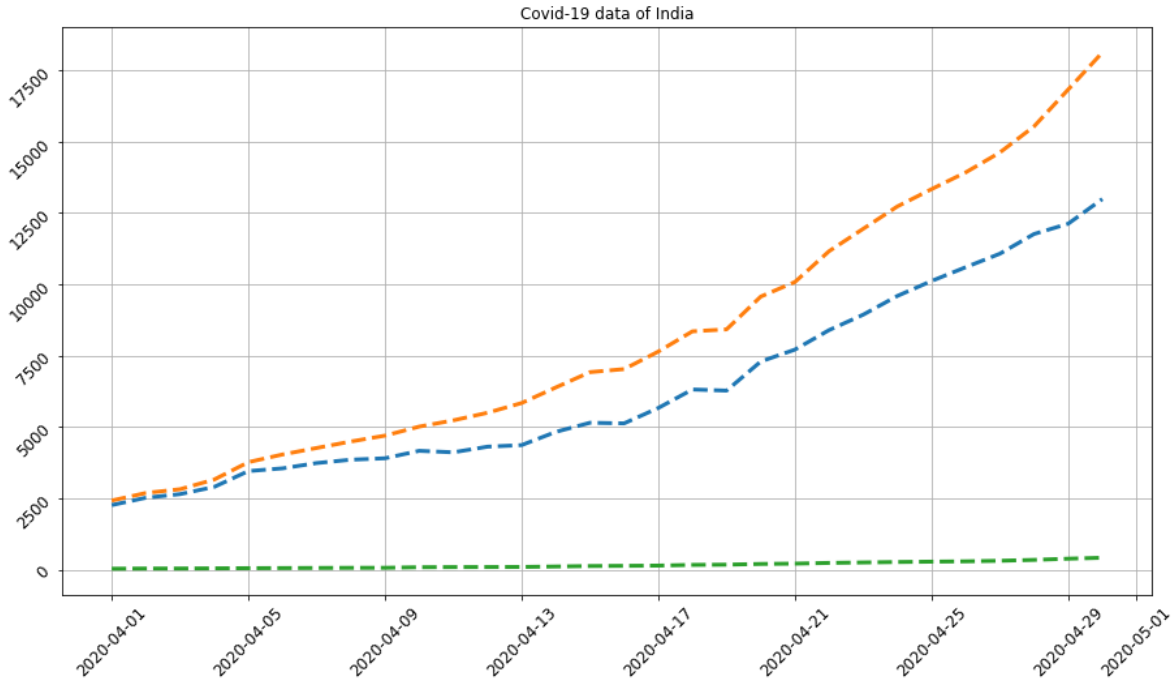
```
Enter the country ->india
Enter the Date -> 2020-04

C:\Users\Lenovo\Anaconda3\lib\site-packages\pandas\plotting\_converter.py:12
9: FutureWarning: Using an implicitly registered datetime converter for a ma
tplotlib plotting method. The converter was registered by pandas on import.
Future versions of pandas will require you to explicitly register matplotlib
converters.

To register the converters:
        >>> from pandas.plotting import register_matplotlib_converters
        >>> register_matplotlib_converters()
  warnings.warn(msg, FutureWarning)
```



In [35]:

```
fig=plt.figure()
plt.style
country=input("Enter the country ->")
data=pd.read_json('https://api.covid19api.com/dayone/country/'+country)
data_1=data.set_index('Date')
Date_info=input('Enter the Date -> ')
data_2=data_1[Date_info]
plt.tick_params(labelrotation=45,labelsize=12)
plt.rcParams["figure.figsize"]=(15,8)
plt.grid()
plt.title("Covid-19 data of India")
plt.plot(data_2['Active'],'--',linewidth=3)
plt.plot(data_2['Confirmed'],'--',linewidth=3)
plt.plot(data_2['Deaths'],'--',linewidth=3)
plt.savefig(r'C:\Users\Lenovo\OneDrive\Pictures\saved_figure\covid-19_data_india')
```

```
Enter the country ->pakistan
Enter the Date -> 2020-04
```



In [ ]:

```
1
```