Mukul_project_of_data_analysis_

August 11, 2021

[24]:
```
from urllib.request import urlretrieve
```

[25]:
```
italy_covid_url = 'https://gist.githubusercontent.com/aakashns/
 ↪f6a004fa20c84fec53262f9a8bfee775/raw/
 ↪f309558b1cf5103424cef58e2ecb8704dcd4d74c/italy-covid-daywise.csv'
urlretrieve(italy_covid_url, 'italy-covid-daywise.csv')
```

[25]:
```
('italy-covid-daywise.csv', <http.client.HTTPMessage at 0x7f6a64856af0>)
```

[41]:
```
covid_df = pd.read_csv('italy-covid-daywise.csv')
type(covid_df)
covid_df
```

[41]:

	date	new_cases	new_deaths	new_tests
0	2019-12-31	0.0	0.0	NaN
1	2020-01-01	0.0	0.0	NaN
2	2020-01-02	0.0	0.0	NaN
3	2020-01-03	0.0	0.0	NaN
4	2020-01-04	0.0	0.0	NaN
..	…	…	…	…
243	2020-08-30	1444.0	1.0	53541.0
244	2020-08-31	1365.0	4.0	42583.0
245	2020-09-01	996.0	6.0	54395.0
246	2020-09-02	975.0	8.0	NaN
247	2020-09-03	1326.0	6.0	NaN

[248 rows x 4 columns]

[42]:
```
covid_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 248 entries, 0 to 247
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   date        248 non-null    object
 1   new_cases   248 non-null    float64
 2   new_deaths  248 non-null    float64
```

```
 3   new_tests   135 non-null    float64
dtypes: float64(3), object(1)
memory usage: 7.9+ KB
```

[43]:
```
covid_df.describe()
```

[43]:
```
          new_cases  new_deaths     new_tests
count   248.000000  248.000000    135.000000
mean   1094.818548  143.133065  31699.674074
std    1554.508002  227.105538  11622.209757
min    -148.000000  -31.000000   7841.000000
25%     123.000000    3.000000  25259.000000
50%     342.000000   17.000000  29545.000000
75%    1371.750000  175.250000  37711.000000
max    6557.000000  971.000000  95273.000000
```

[44]:
```
covid_df.columns
```

[44]:
```
Index(['date', 'new_cases', 'new_deaths', 'new_tests'], dtype='object')
```

[45]:
```
covid_df.shape
```

[45]:
```
(248, 4)
```

[50]:
```
# Pandas format is simliar to this
covid_data_dict = {
    'date':       ['2020-08-30', '2020-08-31', '2020-09-01', '2020-09-02',␣
 ↪'2020-09-03'],
    'new_cases':  [1444, 1365, 996, 975, 1326],
    'new_deaths': [1, 4, 6, 8, 6],
    'new_tests': [53541, 42583, 54395, None, None]
}
```

[51]:
```
covid_data_list = [
    {'date': '2020-08-30', 'new_cases': 1444, 'new_deaths': 1, 'new_tests':␣
 ↪53541},
    {'date': '2020-08-31', 'new_cases': 1365, 'new_deaths': 4, 'new_tests':␣
 ↪42583},
    {'date': '2020-09-01', 'new_cases': 996, 'new_deaths': 6, 'new_tests':␣
 ↪54395},
    {'date': '2020-09-02', 'new_cases': 975, 'new_deaths': 8 },
    {'date': '2020-09-03', 'new_cases': 1326, 'new_deaths': 6},
]
```

[52]:
```
covid_data_dict['new_cases']
```

[52]:
```
[1444, 1365, 996, 975, 1326]
```

[53]:
```
covid_df['new_cases']
```

[53]:
```
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
        ...
243    1444.0
244    1365.0
245     996.0
246     975.0
247    1326.0
Name: new_cases, Length: 248, dtype: float64
```

[54]:
```
type(covid_df['new_cases'])
```

[54]:
```
pandas.core.series.Series
```

[55]:
```
covid_df['new_cases'][246]
```

[55]:
```
975.0
```

[56]:
```
covid_df.at[240, 'new_tests']
```

[56]:
```
57640.0
```

[59]:
```
covid_df.new_cases
```

[59]:
```
0         0.0
1         0.0
2         0.0
3         0.0
4         0.0
        ...
243    1444.0
244    1365.0
245     996.0
246     975.0
247    1326.0
Name: new_cases, Length: 248, dtype: float64
```

[60]:
```
cases_df = covid_df[['date', 'new_cases']]
cases_df
```

[60]:
```
           date  new_cases
0    2019-12-31        0.0
```

```
1    2020-01-01        0.0
2    2020-01-02        0.0
3    2020-01-03        0.0
4    2020-01-04        0.0
..          …         …
243  2020-08-30     1444.0
244  2020-08-31     1365.0
245  2020-09-01      996.0
246  2020-09-02      975.0
247  2020-09-03     1326.0

[248 rows x 2 columns]
```

[61]:
```
covid_df_copy = covid_df.copy()
```

[62]:
```
covid_df
```

[62]:
```
           date  new_cases  new_deaths  new_tests
0    2019-12-31        0.0         0.0        NaN
1    2020-01-01        0.0         0.0        NaN
2    2020-01-02        0.0         0.0        NaN
3    2020-01-03        0.0         0.0        NaN
4    2020-01-04        0.0         0.0        NaN
..          …         …           …          …
243  2020-08-30     1444.0         1.0    53541.0
244  2020-08-31     1365.0         4.0    42583.0
245  2020-09-01      996.0         6.0    54395.0
246  2020-09-02      975.0         8.0        NaN
247  2020-09-03     1326.0         6.0        NaN

[248 rows x 4 columns]
```

[63]:
```
covid_df.loc[243]
```

[63]:
```
date          2020-08-30
new_cases         1444.0
new_deaths           1.0
new_tests        53541.0
Name: 243, dtype: object
```

[64]:
```
covid_df.head(5)
```

[64]:
```
         date  new_cases  new_deaths  new_tests
0  2019-12-31        0.0         0.0        NaN
1  2020-01-01        0.0         0.0        NaN
2  2020-01-02        0.0         0.0        NaN
3  2020-01-03        0.0         0.0        NaN
```

```
4  2020-01-04        0.0         0.0        NaN
```

[65]:
```
covid_df.tail(4)
```

[65]:
```
           date  new_cases  new_deaths  new_tests
244  2020-08-31     1365.0         4.0    42583.0
245  2020-09-01      996.0         6.0    54395.0
246  2020-09-02      975.0         8.0        NaN
247  2020-09-03     1326.0         6.0        NaN
```

[66]:
```
covid_df.at[0, 'new_tests']
```

[66]:
```
nan
```

[67]:
```
type(covid_df.at[0, 'new_tests'])
```

[67]:
```
numpy.float64
```

[68]:
```
covid_df.new_tests.first_valid_index()
```

[68]:
```
111
```

[69]:
```
covid_df.loc[108:113]
```

[69]:
```
           date  new_cases  new_deaths  new_tests
108  2020-04-17     3786.0       525.0        NaN
109  2020-04-18     3493.0       575.0        NaN
110  2020-04-19     3491.0       480.0        NaN
111  2020-04-20     3047.0       433.0     7841.0
112  2020-04-21     2256.0       454.0    28095.0
113  2020-04-22     2729.0       534.0    44248.0
```

[70]:
```
covid_df.sample(10)
```

[70]:
```
           date  new_cases  new_deaths  new_tests
147  2020-05-26      300.0        92.0    33944.0
102  2020-04-11     3951.0       570.0        NaN
46   2020-02-15        0.0         0.0        NaN
7    2020-01-07        0.0         0.0        NaN
177  2020-06-25      577.0       -31.0    29421.0
136  2020-05-15      992.0       262.0    39027.0
111  2020-04-20     3047.0       433.0     7841.0
12   2020-01-12        0.0         0.0        NaN
90   2020-03-30     5217.0       758.0        NaN
44   2020-02-13        0.0         0.0        NaN
```

0.1 Analyzing data from data frames

Let's try to answer some questions about our data.

Q1: What are the total number of reported cases and deaths related to Covid-19 in Italy?

[71]:
```
total_cases = covid_df.new_cases.sum()
total_deaths = covid_df.new_deaths.sum()
print('The number of reported cases is {} and the number of reported deaths is␣
 ↪{}.'.format(int(total_cases), int(total_deaths)))
```

```
The number of reported cases is 271515 and the number of reported deaths is
35497.
```

Q2: What is the overall death rate (ratio of reported deaths to reported cases)?

[73]:
```
death_rate = covid_df.new_deaths.sum() / covid_df.new_cases.sum()
print("The overall reported death rate in Italy is {:.2f} %.".
 ↪format(death_rate*100))
```

```
The overall reported death rate in Italy is 13.07 %.
```

Q3: What is the overall number of tests conducted? A total of 935310 tests were conducted before daily test numbers were reported.

[74]:
```
initial_tests = 935310
total_tests = initial_tests + covid_df.new_tests.sum()
total_tests
```

[74]:
```
5214766.0
```

Q4: What fraction of tests returned a positive result?

[75]:
```
positive_rate = total_cases / total_tests
print('{:.2f}% of tests in Italy led to a positive diagnosis.'.
 ↪format(positive_rate*100))
```

```
5.21% of tests in Italy led to a positive diagnosis.
```

0.2 Querying and sorting rows

Let's say we want only want to look at the days which had more than 1000 reported cases. We can use a boolean expression to check which rows satisfy this criterion.

[93]:
```
high_new_cases = covid_df.new_cases > 1000
high_new_cases
```

[93]:
```
0      False
1      False
2      False
3      False
```

```
4      False
       …
243     True
244     True
245    False
246    False
247     True
Name: new_cases, Length: 248, dtype: bool
```

[94]:
```
covid_df[high_new_cases]
```

[94]:
```
           date  new_cases  new_deaths  new_tests location
68   2020-03-08     1247.0        36.0        NaN    Italy
69   2020-03-09     1492.0       133.0        NaN    Italy
70   2020-03-10     1797.0        98.0        NaN    Italy
72   2020-03-12     2313.0       196.0        NaN    Italy
73   2020-03-13     2651.0       189.0        NaN    Italy
..          …          …           …          …       …
241  2020-08-28     1409.0         5.0    65135.0    Italy
242  2020-08-29     1460.0         9.0    64294.0    Italy
243  2020-08-30     1444.0         1.0    53541.0    Italy
244  2020-08-31     1365.0         4.0    42583.0    Italy
247  2020-09-03     1326.0         6.0        NaN    Italy

[72 rows x 5 columns]
```

[95]:
```
high_cases_df = covid_df[covid_df.new_cases > 1000]
high_cases_df
```

[95]:
```
           date  new_cases  new_deaths  new_tests location
68   2020-03-08     1247.0        36.0        NaN    Italy
69   2020-03-09     1492.0       133.0        NaN    Italy
70   2020-03-10     1797.0        98.0        NaN    Italy
72   2020-03-12     2313.0       196.0        NaN    Italy
73   2020-03-13     2651.0       189.0        NaN    Italy
..          …          …           …          …       …
241  2020-08-28     1409.0         5.0    65135.0    Italy
242  2020-08-29     1460.0         9.0    64294.0    Italy
243  2020-08-30     1444.0         1.0    53541.0    Italy
244  2020-08-31     1365.0         4.0    42583.0    Italy
247  2020-09-03     1326.0         6.0        NaN    Italy

[72 rows x 5 columns]
```

The data frame contains 72 rows, but only the first & last five rows are displayed by default with Jupyter for brevity. We can change some display options to view all the rows.

from IPython.display import display

[96]:
```
from IPython.display import display
with pd.option_context('display.max_rows', 100):
    display(covid_df[covid_df.new_cases > 1000])
```

	date	new_cases	new_deaths	new_tests	location
68	2020-03-08	1247.0	36.0	NaN	Italy
69	2020-03-09	1492.0	133.0	NaN	Italy
70	2020-03-10	1797.0	98.0	NaN	Italy
72	2020-03-12	2313.0	196.0	NaN	Italy
73	2020-03-13	2651.0	189.0	NaN	Italy
74	2020-03-14	2547.0	252.0	NaN	Italy
75	2020-03-15	3497.0	173.0	NaN	Italy
76	2020-03-16	2823.0	370.0	NaN	Italy
77	2020-03-17	4000.0	347.0	NaN	Italy
78	2020-03-18	3526.0	347.0	NaN	Italy
79	2020-03-19	4207.0	473.0	NaN	Italy
80	2020-03-20	5322.0	429.0	NaN	Italy
81	2020-03-21	5986.0	625.0	NaN	Italy
82	2020-03-22	6557.0	795.0	NaN	Italy
83	2020-03-23	5560.0	649.0	NaN	Italy
84	2020-03-24	4789.0	601.0	NaN	Italy
85	2020-03-25	5249.0	743.0	NaN	Italy
86	2020-03-26	5210.0	685.0	NaN	Italy
87	2020-03-27	6153.0	660.0	NaN	Italy
88	2020-03-28	5959.0	971.0	NaN	Italy
89	2020-03-29	5974.0	887.0	NaN	Italy
90	2020-03-30	5217.0	758.0	NaN	Italy
91	2020-03-31	4050.0	810.0	NaN	Italy
92	2020-04-01	4053.0	839.0	NaN	Italy
93	2020-04-02	4782.0	727.0	NaN	Italy
94	2020-04-03	4668.0	760.0	NaN	Italy
95	2020-04-04	4585.0	764.0	NaN	Italy
96	2020-04-05	4805.0	681.0	NaN	Italy
97	2020-04-06	4316.0	527.0	NaN	Italy
98	2020-04-07	3599.0	636.0	NaN	Italy
99	2020-04-08	3039.0	604.0	NaN	Italy
100	2020-04-09	3836.0	540.0	NaN	Italy
101	2020-04-10	4204.0	612.0	NaN	Italy
102	2020-04-11	3951.0	570.0	NaN	Italy
103	2020-04-12	4694.0	619.0	NaN	Italy
104	2020-04-13	4092.0	431.0	NaN	Italy
105	2020-04-14	3153.0	564.0	NaN	Italy
106	2020-04-15	2972.0	604.0	NaN	Italy
107	2020-04-16	2667.0	578.0	NaN	Italy
108	2020-04-17	3786.0	525.0	NaN	Italy
109	2020-04-18	3493.0	575.0	NaN	Italy
110	2020-04-19	3491.0	480.0	NaN	Italy

111	2020-04-20	3047.0	433.0	7841.0	Italy
112	2020-04-21	2256.0	454.0	28095.0	Italy
113	2020-04-22	2729.0	534.0	44248.0	Italy
114	2020-04-23	3370.0	437.0	37083.0	Italy
115	2020-04-24	2646.0	464.0	95273.0	Italy
116	2020-04-25	3021.0	420.0	38676.0	Italy
117	2020-04-26	2357.0	415.0	24113.0	Italy
118	2020-04-27	2324.0	260.0	26678.0	Italy
119	2020-04-28	1739.0	333.0	37554.0	Italy
120	2020-04-29	2091.0	382.0	38589.0	Italy
121	2020-04-30	2086.0	323.0	41441.0	Italy
122	2020-05-01	1872.0	285.0	43732.0	Italy
123	2020-05-02	1965.0	269.0	31231.0	Italy
124	2020-05-03	1900.0	474.0	27047.0	Italy
125	2020-05-04	1389.0	174.0	22999.0	Italy
126	2020-05-05	1221.0	195.0	32211.0	Italy
127	2020-05-06	1075.0	236.0	37771.0	Italy
128	2020-05-07	1444.0	369.0	13665.0	Italy
129	2020-05-08	1401.0	274.0	45428.0	Italy
130	2020-05-09	1327.0	243.0	36091.0	Italy
131	2020-05-10	1083.0	194.0	31384.0	Italy
134	2020-05-13	1402.0	172.0	37049.0	Italy
236	2020-08-23	1071.0	3.0	47463.0	Italy
237	2020-08-24	1209.0	7.0	33358.0	Italy
240	2020-08-27	1366.0	13.0	57640.0	Italy
241	2020-08-28	1409.0	5.0	65135.0	Italy
242	2020-08-29	1460.0	9.0	64294.0	Italy
243	2020-08-30	1444.0	1.0	53541.0	Italy
244	2020-08-31	1365.0	4.0	42583.0	Italy
247	2020-09-03	1326.0	6.0	NaN	Italy

[104]:
```
positive_rate
```

[104]: 0.05206657403227681

[98]:
```
high_ratio_df = covid_df[covid_df.new_cases / covid_df.new_tests >␣
↪positive_rate]
```

[99]:
```
high_ratio_df
```

[99]:

	date	new_cases	new_deaths	new_tests	location
111	2020-04-20	3047.0	433.0	7841.0	Italy
112	2020-04-21	2256.0	454.0	28095.0	Italy
113	2020-04-22	2729.0	534.0	44248.0	Italy
114	2020-04-23	3370.0	437.0	37083.0	Italy
116	2020-04-25	3021.0	420.0	38676.0	Italy
117	2020-04-26	2357.0	415.0	24113.0	Italy

```
118  2020-04-27     2324.0       260.0    26678.0    Italy
120  2020-04-29     2091.0       382.0    38589.0    Italy
123  2020-05-02     1965.0       269.0    31231.0    Italy
124  2020-05-03     1900.0       474.0    27047.0    Italy
125  2020-05-04     1389.0       174.0    22999.0    Italy
128  2020-05-07     1444.0       369.0    13665.0    Italy
```

The result of performing an operation on two columns is a new series.

[100]:
```
covid_df.new_cases / covid_df.new_tests
```

[100]:
```
0           NaN
1           NaN
2           NaN
3           NaN
4           NaN
          …
243    0.026970
244    0.032055
245    0.018311
246         NaN
247         NaN
Length: 248, dtype: float64
```

[101]:
```
covid_df['positive_rate'] = covid_df.new_cases / covid_df.new_tests
covid_df
```

[101]:
```
            date  new_cases  new_deaths  new_tests location  positive_rate
0     2019-12-31        0.0         0.0        NaN    Italy            NaN
1     2020-01-01        0.0         0.0        NaN    Italy            NaN
2     2020-01-02        0.0         0.0        NaN    Italy            NaN
3     2020-01-03        0.0         0.0        NaN    Italy            NaN
4     2020-01-04        0.0         0.0        NaN    Italy            NaN
..           …        …          …       …        …      …              …
243   2020-08-30     1444.0         1.0    53541.0    Italy       0.026970
244   2020-08-31     1365.0         4.0    42583.0    Italy       0.032055
245   2020-09-01      996.0         6.0    54395.0    Italy       0.018311
246   2020-09-02      975.0         8.0        NaN    Italy            NaN
247   2020-09-03     1326.0         6.0        NaN    Italy            NaN

[248 rows x 6 columns]
```

For now, let's remove the positive_rate column using the drop method.

Sorting rows using column values

[108]:
```
covid_df.sort_values('new_cases', ascending=False).head(10)
```

[108]:

	date	new_cases	new_deaths	new_tests	location
82	2020-03-22	6557.0	795.0	NaN	Italy
87	2020-03-27	6153.0	660.0	NaN	Italy
81	2020-03-21	5986.0	625.0	NaN	Italy
89	2020-03-29	5974.0	887.0	NaN	Italy
88	2020-03-28	5959.0	971.0	NaN	Italy
83	2020-03-23	5560.0	649.0	NaN	Italy
80	2020-03-20	5322.0	429.0	NaN	Italy
85	2020-03-25	5249.0	743.0	NaN	Italy
90	2020-03-30	5217.0	758.0	NaN	Italy
86	2020-03-26	5210.0	685.0	NaN	Italy

It looks like the last two weeks of March had the highest number of daily cases. Let's compare this to the days where the highest number of deaths were recorded.

covid_df.sort_values('new_deaths', ascending=Fa

[109]:
```
covid_df.sort_values('new_deaths', ascending=False).head(10)
```

[109]:

	date	new_cases	new_deaths	new_tests	location
88	2020-03-28	5959.0	971.0	NaN	Italy
89	2020-03-29	5974.0	887.0	NaN	Italy
92	2020-04-01	4053.0	839.0	NaN	Italy
91	2020-03-31	4050.0	810.0	NaN	Italy
82	2020-03-22	6557.0	795.0	NaN	Italy
95	2020-04-04	4585.0	764.0	NaN	Italy
94	2020-04-03	4668.0	760.0	NaN	Italy
90	2020-03-30	5217.0	758.0	NaN	Italy
85	2020-03-25	5249.0	743.0	NaN	Italy
93	2020-04-02	4782.0	727.0	NaN	Italy

[110]:
```
covid_df.sort_values('new_cases').head(10)
```

[110]:

	date	new_cases	new_deaths	new_tests	location
172	2020-06-20	-148.0	47.0	29875.0	Italy
0	2019-12-31	0.0	0.0	NaN	Italy
29	2020-01-29	0.0	0.0	NaN	Italy
30	2020-01-30	0.0	0.0	NaN	Italy
32	2020-02-01	0.0	0.0	NaN	Italy
33	2020-02-02	0.0	0.0	NaN	Italy
34	2020-02-03	0.0	0.0	NaN	Italy
36	2020-02-05	0.0	0.0	NaN	Italy
37	2020-02-06	0.0	0.0	NaN	Italy
38	2020-02-07	0.0	0.0	NaN	Italy

You can use the .at method to modify a specific value within the dataframe.

0.3 Working with dates

[112]:
```
covid_df.date
```

[112]:
```
0      2019-12-31
1      2020-01-01
2      2020-01-02
3      2020-01-03
4      2020-01-04
          …
243    2020-08-30
244    2020-08-31
245    2020-09-01
246    2020-09-02
247    2020-09-03
Name: date, Length: 248, dtype: object
```

[113]:
```
covid_df['date'] = pd.to_datetime(covid_df.date)
covid_df['date']
```

[113]:
```
0     2019-12-31
1     2020-01-01
2     2020-01-02
3     2020-01-03
4     2020-01-04
         …
243   2020-08-30
244   2020-08-31
245   2020-09-01
246   2020-09-02
247   2020-09-03
Name: date, Length: 248, dtype: datetime64[ns]
```

[116]:
```
# You can see that it now has the datatype datetime64. We can now extract␣
 ↪different parts of the data into separate columns, using the DatetimeIndex
covid_df['year'] = pd.DatetimeIndex(covid_df.date).year
covid_df['month'] = pd.DatetimeIndex(covid_df.date).month
covid_df['day'] = pd.DatetimeIndex(covid_df.date).day
covid_df['weekday'] = pd.DatetimeIndex(covid_df.date).weekday
covid_df
```

[116]:

	date	new_cases	new_deaths	new_tests	location	year	month	day	\
0	2019-12-31	0.0	0.0	NaN	Italy	2019	12	31	
1	2020-01-01	0.0	0.0	NaN	Italy	2020	1	1	
2	2020-01-02	0.0	0.0	NaN	Italy	2020	1	2	
3	2020-01-03	0.0	0.0	NaN	Italy	2020	1	3	
4	2020-01-04	0.0	0.0	NaN	Italy	2020	1	4	

```
..          …          …          …          …        …    …     …  …
243 2020-08-30     1444.0        1.0    53541.0    Italy  2020     8   30
244 2020-08-31     1365.0        4.0    42583.0    Italy  2020     8   31
245 2020-09-01      996.0        6.0    54395.0    Italy  2020     9    1
246 2020-09-02      975.0        8.0        NaN    Italy  2020     9    2
247 2020-09-03     1326.0        6.0        NaN    Italy  2020     9    3

     weekday
0          1
1          2
2          3
3          4
4          5
..       …
243        6
244        0
245        1
246        2
247        3

[248 rows x 9 columns]
```

[118]:
```
# Query the rows for May
covid_df_may = covid_df[covid_df.month == 5]

# Extract the subset of columns to be aggregated
covid_df_may_metrics = covid_df_may[['new_cases', 'new_deaths', 'new_tests']]

# Get the column-wise sum
covid_may_totals = covid_df_may_metrics.sum()
covid_may_totals
```

[118]:
```
new_cases       29073.0
new_deaths       5658.0
new_tests     1078720.0
dtype: float64
```

[119]:
```
type(covid_may_totals)
```

[119]:
```
pandas.core.series.Series
```

We can also combine the above operations into a single statement.

[120]:
```
covid_df[covid_df.month == 5][['new_cases', 'new_deaths', 'new_tests']].sum()
```

[120]:
```
new_cases       29073.0
new_deaths       5658.0
```

```
new_tests     1078720.0
dtype: float64
```

[122]:
```
# Overall average
covid_df.new_cases.mean()
```

[122]:
```
1247.2571428571428
```

[123]:
```
# Average for Sundays
covid_df[covid_df.weekday == 6].new_cases.mean()
```

[123]:
```
1247.2571428571428
```

[124]:
```
#Grouping and aggregation
#As a next step, we might want to summarize the day-wise data and create a new␣
 ↪dataframe with month-wise data. We can use the groupby function to create a␣
 ↪group for each month,
covid_month_df = covid_df.groupby('month')[['new_cases', 'new_deaths',␣
 ↪'new_tests']].sum()
covid_month_df
```

[124]:

month	new_cases	new_deaths	new_tests
1	3.0	0.0	0.0
2	885.0	21.0	0.0
3	100851.0	11570.0	0.0
4	101852.0	16091.0	419591.0
5	29073.0	5658.0	1078720.0
6	7772.0	1404.0	830354.0
7	6722.0	388.0	797692.0
8	21060.0	345.0	1098704.0
9	3297.0	20.0	54395.0
12	0.0	0.0	0.0

[125]:
```
covid_month_mean_df = covid_df.groupby('month')[['new_cases', 'new_deaths',␣
 ↪'new_tests']].mean()
covid_month_mean_df
```

[125]:

month	new_cases	new_deaths	new_tests
1	0.096774	0.000000	NaN
2	30.517241	0.724138	NaN
3	3253.258065	373.225806	NaN
4	3395.066667	536.366667	38144.636364
5	937.838710	182.516129	34797.419355
6	259.066667	46.800000	27678.466667
7	216.838710	12.516129	25732.000000

```
8       679.354839   11.129032  35442.064516
9      1099.000000    6.666667  54395.000000
12        0.000000    0.000000           NaN
```

[127]:
```
# We can use the cumsum method to compute the cumulative sum of a column as a new series
covid_df['total_cases'] = covid_df.new_cases.cumsum()
covid_df['total_deaths'] = covid_df.new_deaths.cumsum()
covid_df['total_tests'] = covid_df.new_tests.cumsum() + initial_tests
covid_df
```

[127]:
```
           date  new_cases  new_deaths  new_tests location  year  month  day  \
0    2019-12-31        0.0         0.0        NaN    Italy  2019     12   31
1    2020-01-01        0.0         0.0        NaN    Italy  2020      1    1
2    2020-01-02        0.0         0.0        NaN    Italy  2020      1    2
3    2020-01-03        0.0         0.0        NaN    Italy  2020      1    3
4    2020-01-04        0.0         0.0        NaN    Italy  2020      1    4
..          …          …           …          …        …     …    …  …
243  2020-08-30     1444.0         1.0    53541.0    Italy  2020      8   30
244  2020-08-31     1365.0         4.0    42583.0    Italy  2020      8   31
245  2020-09-01      996.0         6.0    54395.0    Italy  2020      9    1
246  2020-09-02      975.0         8.0        NaN    Italy  2020      9    2
247  2020-09-03     1326.0         6.0        NaN    Italy  2020      9    3

     weekday  total_cases  total_deaths  total_tests
0          1          0.0           0.0          NaN
1          2          0.0           0.0          NaN
2          3          0.0           0.0          NaN
3          4          0.0           0.0          NaN
4          5          0.0           0.0          NaN
..       …          …           …          …
243        6     266853.0       35473.0    5117788.0
244        0     268218.0       35477.0    5160371.0
245        1     269214.0       35483.0    5214766.0
246        2     270189.0       35491.0          NaN
247        3     271515.0       35497.0          NaN

[248 rows x 12 columns]
```

[128]:
```
# Merging data from multiple sources
urlretrieve('https://gist.githubusercontent.com/aakashns/8684589ef4f266116cdce023377fc9c8/raw/99ce3826b2a9d1e6d0bde7e9e559fc8b6e9ac88b/locations.csv',
            'locations.csv')
```

[128]:
```
('locations.csv', <http.client.HTTPMessage at 0x7f6a63f27130>)
```

[84]:
```
locations_df = pd.read_csv('locations.csv')
```

[85]:
```
locations_df
```

[85]:
```
          location continent    population  life_expectancy  \
0      Afghanistan      Asia  3.892834e+07            64.83
1          Albania    Europe  2.877800e+06            78.57
2          Algeria    Africa  4.385104e+07            76.88
3          Andorra    Europe  7.726500e+04            83.73
4           Angola    Africa  3.286627e+07            61.15
..             ...       ...           ...              ...
207          Yemen      Asia  2.982597e+07            66.12
208         Zambia    Africa  1.838396e+07            63.89
209       Zimbabwe    Africa  1.486293e+07            61.49
210          World       NaN  7.794799e+09            72.58
211  International       NaN           NaN              NaN

     hospital_beds_per_thousand  gdp_per_capita
0                         0.500        1803.987
1                         2.890       11803.431
2                         1.900       13913.839
3                           NaN             NaN
4                           NaN        5819.495
..                          ...             ...
207                       0.700        1479.147
208                       2.000        3689.251
209                       1.700        1899.775
210                       2.705       15469.207
211                         NaN             NaN

[212 rows x 6 columns]
```

[86]:
```
locations_df[locations_df.location == "Italy"]
```

[86]:
```
   location continent  population  life_expectancy  \
97    Italy    Europe  60461828.0            83.51

    hospital_beds_per_thousand  gdp_per_capita
97                        3.18       35220.084
```

[87]:
```
covid_df['location'] = "Italy"
covid_df
```

[87]:
```
           date  new_cases  new_deaths  new_tests location
0    2019-12-31        0.0         0.0        NaN    Italy
1    2020-01-01        0.0         0.0        NaN    Italy
2    2020-01-02        0.0         0.0        NaN    Italy
```

```
3    2020-01-03        0.0         0.0        NaN    Italy
4    2020-01-04        0.0         0.0        NaN    Italy
..          …         …           …          …      …
243  2020-08-30     1444.0         1.0    53541.0    Italy
244  2020-08-31     1365.0         4.0    42583.0    Italy
245  2020-09-01      996.0         6.0    54395.0    Italy
246  2020-09-02      975.0         8.0        NaN    Italy
247  2020-09-03     1326.0         6.0        NaN    Italy

[248 rows x 5 columns]
```

[89]:
```
merged_df = covid_df.merge(locations_df, on="location")
merged_df
```

[89]:
```
           date  new_cases  new_deaths  new_tests location continent  \
0    2019-12-31        0.0         0.0        NaN    Italy    Europe
1    2020-01-01        0.0         0.0        NaN    Italy    Europe
2    2020-01-02        0.0         0.0        NaN    Italy    Europe
3    2020-01-03        0.0         0.0        NaN    Italy    Europe
4    2020-01-04        0.0         0.0        NaN    Italy    Europe
..          …         …           …          …      …        …
243  2020-08-30     1444.0         1.0    53541.0    Italy    Europe
244  2020-08-31     1365.0         4.0    42583.0    Italy    Europe
245  2020-09-01      996.0         6.0    54395.0    Italy    Europe
246  2020-09-02      975.0         8.0        NaN    Italy    Europe
247  2020-09-03     1326.0         6.0        NaN    Italy    Europe

     population  life_expectancy  hospital_beds_per_thousand  gdp_per_capita
0    60461828.0            83.51                        3.18       35220.084
1    60461828.0            83.51                        3.18       35220.084
2    60461828.0            83.51                        3.18       35220.084
3    60461828.0            83.51                        3.18       35220.084
4    60461828.0            83.51                        3.18       35220.084
..          …                …                           …              …
243  60461828.0            83.51                        3.18       35220.084
244  60461828.0            83.51                        3.18       35220.084
245  60461828.0            83.51                        3.18       35220.084
246  60461828.0            83.51                        3.18       35220.084
247  60461828.0            83.51                        3.18       35220.084

[248 rows x 10 columns]
```

[131]:
```
locations_df[locations_df.location == "Italy"]
```

[131]:
```
   location continent  population  life_expectancy  \
97    Italy    Europe  60461828.0            83.51
```

```
    hospital_beds_per_thousand  gdp_per_capita
97                        3.18       35220.084
```

[132]:
```
covid_df['location'] = "Italy"
covid_df
```

[132]:
```
           date  new_cases  new_deaths  new_tests location  year  month  day  \
0    2019-12-31        0.0         0.0        NaN    Italy  2019     12   31
1    2020-01-01        0.0         0.0        NaN    Italy  2020      1    1
2    2020-01-02        0.0         0.0        NaN    Italy  2020      1    2
3    2020-01-03        0.0         0.0        NaN    Italy  2020      1    3
4    2020-01-04        0.0         0.0        NaN    Italy  2020      1    4
..          …          …           …          …        …     …     …  …
243  2020-08-30     1444.0         1.0    53541.0    Italy  2020      8   30
244  2020-08-31     1365.0         4.0    42583.0    Italy  2020      8   31
245  2020-09-01      996.0         6.0    54395.0    Italy  2020      9    1
246  2020-09-02      975.0         8.0        NaN    Italy  2020      9    2
247  2020-09-03     1326.0         6.0        NaN    Italy  2020      9    3

     weekday  total_cases  total_deaths  total_tests
0          1          0.0           0.0          NaN
1          2          0.0           0.0          NaN
2          3          0.0           0.0          NaN
3          4          0.0           0.0          NaN
4          5          0.0           0.0          NaN
..       …          …           …          …
243        6     266853.0       35473.0    5117788.0
244        0     268218.0       35477.0    5160371.0
245        1     269214.0       35483.0    5214766.0
246        2     270189.0       35491.0          NaN
247        3     271515.0       35497.0          NaN

[248 rows x 12 columns]
```

[133]:
```
merged_df = covid_df.merge(locations_df, on="location")
merged_df
```

[133]:
```
           date  new_cases  new_deaths  new_tests location  year  month  day  \
0    2019-12-31        0.0         0.0        NaN    Italy  2019     12   31
1    2020-01-01        0.0         0.0        NaN    Italy  2020      1    1
2    2020-01-02        0.0         0.0        NaN    Italy  2020      1    2
3    2020-01-03        0.0         0.0        NaN    Italy  2020      1    3
4    2020-01-04        0.0         0.0        NaN    Italy  2020      1    4
..          …          …           …          …        …     …     …  …
243  2020-08-30     1444.0         1.0    53541.0    Italy  2020      8   30
244  2020-08-31     1365.0         4.0    42583.0    Italy  2020      8   31
245  2020-09-01      996.0         6.0    54395.0    Italy  2020      9    1
```

```
246 2020-09-02      975.0         8.0        NaN    Italy  2020      9    2
247 2020-09-03     1326.0         6.0        NaN    Italy  2020      9    3

     weekday  total_cases  total_deaths  total_tests continent  population  \
0          1          0.0           0.0          NaN    Europe  60461828.0
1          2          0.0           0.0          NaN    Europe  60461828.0
2          3          0.0           0.0          NaN    Europe  60461828.0
3          4          0.0           0.0          NaN    Europe  60461828.0
4          5          0.0           0.0          NaN    Europe  60461828.0
..       …          …             …            …        …         …
243        6     266853.0       35473.0    5117788.0    Europe  60461828.0
244        0     268218.0       35477.0    5160371.0    Europe  60461828.0
245        1     269214.0       35483.0    5214766.0    Europe  60461828.0
246        2     270189.0       35491.0          NaN    Europe  60461828.0
247        3     271515.0       35497.0          NaN    Europe  60461828.0

     life_expectancy  hospital_beds_per_thousand  gdp_per_capita
0              83.51                        3.18       35220.084
1              83.51                        3.18       35220.084
2              83.51                        3.18       35220.084
3              83.51                        3.18       35220.084
4              83.51                        3.18       35220.084
..               …                           …              …
243            83.51                        3.18       35220.084
244            83.51                        3.18       35220.084
245            83.51                        3.18       35220.084
246            83.51                        3.18       35220.084
247            83.51                        3.18       35220.084

[248 rows x 17 columns]
```

[135]:
```
merged_df['cases_per_million'] = merged_df.total_cases * 1e6 / merged_df.
 ↪population
merged_df['deaths_per_million'] = merged_df.total_deaths * 1e6 / merged_df.
 ↪population
merged_df['tests_per_million'] = merged_df.total_tests * 1e6 / merged_df.
 ↪population
merged_df
```

[135]:
```
          date  new_cases  new_deaths  new_tests location  year  month  day  \
0   2019-12-31        0.0         0.0        NaN    Italy  2019     12   31
1   2020-01-01        0.0         0.0        NaN    Italy  2020      1    1
2   2020-01-02        0.0         0.0        NaN    Italy  2020      1    2
3   2020-01-03        0.0         0.0        NaN    Italy  2020      1    3
4   2020-01-04        0.0         0.0        NaN    Italy  2020      1    4
..         …         …           …          …        …     …      …    …
243 2020-08-30     1444.0         1.0    53541.0    Italy  2020      8   30
```

```
244 2020-08-31     1365.0         4.0    42583.0    Italy  2020      8   31
245 2020-09-01      996.0         6.0    54395.0    Italy  2020      9    1
246 2020-09-02      975.0         8.0        NaN    Italy  2020      9    2
247 2020-09-03     1326.0         6.0        NaN    Italy  2020      9    3

     weekday  total_cases  total_deaths  total_tests continent  population  \
0          1          0.0           0.0          NaN    Europe  60461828.0
1          2          0.0           0.0          NaN    Europe  60461828.0
2          3          0.0           0.0          NaN    Europe  60461828.0
3          4          0.0           0.0          NaN    Europe  60461828.0
4          5          0.0           0.0          NaN    Europe  60461828.0
..       …          …           …          …       …         …
243        6     266853.0       35473.0    5117788.0    Europe  60461828.0
244        0     268218.0       35477.0    5160371.0    Europe  60461828.0
245        1     269214.0       35483.0    5214766.0    Europe  60461828.0
246        2     270189.0       35491.0          NaN    Europe  60461828.0
247        3     271515.0       35497.0          NaN    Europe  60461828.0

     life_expectancy  hospital_beds_per_thousand  gdp_per_capita  \
0              83.51                        3.18       35220.084
1              83.51                        3.18       35220.084
2              83.51                        3.18       35220.084
3              83.51                        3.18       35220.084
4              83.51                        3.18       35220.084
..               …                           …             …
243            83.51                        3.18       35220.084
244            83.51                        3.18       35220.084
245            83.51                        3.18       35220.084
246            83.51                        3.18       35220.084
247            83.51                        3.18       35220.084

     cases_per_million  deaths_per_million  tests_per_million
0             0.000000            0.000000                NaN
1             0.000000            0.000000                NaN
2             0.000000            0.000000                NaN
3             0.000000            0.000000                NaN
4             0.000000            0.000000                NaN
..                 …                   …               …
243        4413.578101          586.700753       84644.943252
244        4436.154329          586.766910       85349.238862
245        4452.627532          586.866146       86248.897403
246        4468.753409          586.998461                NaN
247        4490.684602          587.097697                NaN

[248 rows x 20 columns]
```

[138]:
```
#Writing data back to files
result_df = merged_df[['date',
                       'new_cases',
                       'total_cases',
                       'new_deaths',
                       'total_deaths',
                       'new_tests',
                       'total_tests',
                       'cases_per_million',
                       'deaths_per_million',
                       'tests_per_million']]
```

[139]:
```
result_df
```

[139]:

	date	new_cases	total_cases	new_deaths	total_deaths	new_tests	total_tests	cases_per_million	deaths_per_million	tests_per_million
0	2019-12-31	0.0	0.0	0.0	0.0	NaN	NaN	0.000000	0.000000	NaN
1	2020-01-01	0.0	0.0	0.0	0.0	NaN	NaN	0.000000	0.000000	NaN
2	2020-01-02	0.0	0.0	0.0	0.0	NaN	NaN	0.000000	0.000000	NaN
3	2020-01-03	0.0	0.0	0.0	0.0	NaN	NaN	0.000000	0.000000	NaN
4	2020-01-04	0.0	0.0	0.0	0.0	NaN	NaN	0.000000	0.000000	NaN
..	…	…	…	…	…	…	…	…	…	…
243	2020-08-30	1444.0	266853.0	1.0	35473.0	53541.0	5117788.0	4413.578101	586.700753	84644.943252
244	2020-08-31	1365.0	268218.0	4.0	35477.0	42583.0	5160371.0	4436.154329	586.766910	85349.238862
245	2020-09-01	996.0	269214.0	6.0	35483.0	54395.0	5214766.0	4452.627532	586.866146	86248.897403
246	2020-09-02	975.0	270189.0	8.0	35491.0	NaN	NaN	4468.753409	586.998461	NaN
247	2020-09-03	1326.0	271515.0	6.0	35497.0	NaN	NaN	4490.684602	587.097697	NaN

[248 rows x 10 columns]

[140]:
```
result_df.to_csv('results.csv', index=None)
```

0.4 Bonus: Basic Plotting with Pandas

We generally use a library like matplotlib or seaborn plot graphs within a Jupyter notebook. However, Pandas dataframes & series provide a handy .plot method for quick and easy plotting.

Let's plot a line graph showing how the number of daily cases varies over time.

[141]:
```
result_df.new_cases.plot();
```