In [2]:
```
from urllib.request import urlretrieve
```

In [3]:
```
import pandas as pd
```

In [4]:
```
italy_covid_url = 'https://gist.githubusercontent.com/aakashns/f6a004fa20c84fec53

urlretrieve(italy_covid_url, 'italy-covid-daywise.csv')
```

Out[4]:
```
('italy-covid-daywise.csv', <http.client.HTTPMessage at 0x1eea7246dc0>)
```

In [5]:
```
import pandas as pd
```

In [6]:
```
covid_df = pd.read_csv('italy-covid-daywise.csv')
```

In [7]:
```
type(covid_df)
```

Out[7]:
```
pandas.core.frame.DataFrame
```

In [8]:
```
covid_df
```

Out[8]:

	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 × 4 columns

In [9]:
```
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
```

In [10]:
```
covid_df.describe()
```

Out[10]:

	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

In [11]:
```
covid_df.columns
```

Out[11]:
```
Index(['date', 'new_cases', 'new_deaths', 'new_tests'], dtype='object')
```

In [12]:
```
covid_df.shape
```

Out[12]:
```
(248, 4)
```

Retrieving data from a data frame

In [13]:
```
# Pandas format is simliar to this
covid_data_dict = {
    'date':       ['2020-08-30', '2020-08-31', '2020-09-01', '2020-09-02', '2020-
    'new_cases':  [1444, 1365, 996, 975, 1326],
    'new_deaths': [1, 4, 6, 8, 6],
    'new_tests': [53541, 42583, 54395, None, None]
}
```

In [14]:
```
# Pandas format is not similar to this
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},
]
```

In [15]:
```
covid_data_dict['new_cases']
```

Out[15]:
```
[1444, 1365, 996, 975, 1326]
```

In [16]:
```
covid_df['new_cases']
```

Out[16]:
```
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
```

In [17]:
```
type(covid_df['new_cases'])
```

Out[17]:
```
pandas.core.series.Series
```

In [18]:
```
covid_df['new_cases'][246]
```

Out[18]:
```
975.0
```

In [19]:
```
covid_df['new_tests'][240]
```

Out[19]:
```
57640.0
```

In [20]:
```
covid_df.at[246, 'new_cases']
```

Out[20]:
```
975.0
```

In [21]:
```
covid_df.at[240, 'new_tests']
```

Out[21]:
```
57640.0
```

In [22]:
```
covid_df.new_cases
```

Out[22]:
```
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
```

In [23]:
```
cases_df = covid_df[['date', 'new_cases']]
cases_df
```

Out[23]:

	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 × 2 columns

In [24]:
```
covid_df_copy = covid_df.copy()
covid_df
```

Out[24]:

	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 × 4 columns

In [25]:
```
covid_df.loc[243]
```

Out[25]:
```
date          2020-08-30
new_cases         1444.0
new_deaths           1.0
new_tests        53541.0
Name: 243, dtype: object
```

In [26]:
```
type(covid_df.loc[243])
```

Out[26]:
```
pandas.core.series.Series
```

In [27]:
```
covid_df.head(5)
```

Out[27]:

	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

In [28]:
```
covid_df.tail(4)
```

Out[28]:

	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

In [29]:
```
covid_df.at[0, 'new_tests']
```

Out[29]:
```
nan
```

In [30]:
```
type(covid_df.at[0, 'new_tests'])
```

Out[30]:
```
numpy.float64
```

In [31]:
```
covid_df.new_tests.first_valid_index()
```

Out[31]:
```
111
```

In [32]:
```
covid_df.loc[108:113]
```

Out[32]:

	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

In [33]:
```
covid_df.sample(10)
```

Out[33]:

	date	new_cases	new_deaths	new_tests
2	2020-01-02	0.0	0.0	NaN
34	2020-02-03	0.0	0.0	NaN
247	2020-09-03	1326.0	6.0	NaN
116	2020-04-25	3021.0	420.0	38676.0
217	2020-08-04	159.0	12.0	23491.0
233	2020-08-20	642.0	7.0	49662.0
220	2020-08-07	401.0	6.0	30392.0
111	2020-04-20	3047.0	433.0	7841.0
79	2020-03-19	4207.0	473.0	NaN
175	2020-06-23	221.0	23.0	23225.0

Analyzing data from data frames

In [34]:
```
# Q: What are the total number of reported cases and deaths related to Covid-19 i
```

In [35]:
```
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 {}
```

```
The number of reported cases is 271515 and the number of reported deaths is 354
97.
```

In [36]:
```
# Q: What is the overall death rate (ratio of reported deaths to reported cases):
```

In [37]:
```
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*1
```

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

In [38]:
```
# Q: What is the overall number of tests conducted? A total of 935310 tests were
```

In [39]:
```
initial_tests = 935310
total_tests = initial_tests + covid_df.new_tests.sum()
total_tests
```

Out[39]:
```
5214766.0
```

In [40]:
```
# Q: What fraction of tests returned a positive result?
```

In [41]:
```
positive_rate = total_cases / total_tests
print('{:.2f}% of tests in Italy led to a positive diagnosis.'.format(positive_ra
```

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

Querying and sorting rows

In [42]:
```
high_new_cases = covid_df.new_cases > 1000
high_new_cases
```

Out[42]:
```
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
```

In [43]:
```
covid_df[high_new_cases]
```

Out[43]:

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

72 rows × 4 columns

In [44]:
```
high_cases_df = covid_df[covid_df.new_cases > 1000]
high_cases_df
```

Out[44]:

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

72 rows × 4 columns

In [45]:
```
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
68	2020-03-08	1247.0	36.0	NaN
69	2020-03-09	1492.0	133.0	NaN
70	2020-03-10	1797.0	98.0	NaN
72	2020-03-12	2313.0	196.0	NaN
73	2020-03-13	2651.0	189.0	NaN
74	2020-03-14	2547.0	252.0	NaN
75	2020-03-15	3497.0	173.0	NaN
76	2020-03-16	2823.0	370.0	NaN
77	2020-03-17	4000.0	347.0	NaN
78	2020-03-18	3526.0	347.0	NaN
79	2020-03-19	4207.0	473.0	NaN

In [46]:
```
positive_rate
```

Out[46]:
```
0.05206657403227681
```

In [47]:
```
high_ratio_df = covid_df[covid_df.new_cases / covid_df.new_tests > positive_rate]
high_ratio_df
```

Out[47]:

	date	new_cases	new_deaths	new_tests
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
114	2020-04-23	3370.0	437.0	37083.0
116	2020-04-25	3021.0	420.0	38676.0
117	2020-04-26	2357.0	415.0	24113.0
118	2020-04-27	2324.0	260.0	26678.0
120	2020-04-29	2091.0	382.0	38589.0
123	2020-05-02	1965.0	269.0	31231.0
124	2020-05-03	1900.0	474.0	27047.0
125	2020-05-04	1389.0	174.0	22999.0
128	2020-05-07	1444.0	369.0	13665.0

In [48]:
```
covid_df.new_cases / covid_df.new_tests
```

Out[48]:
```
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
```

In [49]:
```
covid_df['positive_rate'] = covid_df.new_cases / covid_df.new_tests
covid_df
```

Out[49]:

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

248 rows × 5 columns

In [50]:
```
covid_df.drop(columns=['positive_rate'], inplace=True)
```

Sorting rows using column value

In [51]:
```
covid_df.sort_values('new_cases', ascending=False).head(10)
```

Out[51]:

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

In [52]:
```
covid_df.sort_values('new_deaths', ascending=False).head(10)
```

Out[52]:

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

In [53]:
```
covid_df.sort_values('new_cases').head(10)
```

Out[53]:

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

In [54]:
```
covid_df.loc[169:175]
```

Out[54]:

	date	new_cases	new_deaths	new_tests
169	2020-06-17	210.0	34.0	33957.0
170	2020-06-18	328.0	43.0	32921.0
171	2020-06-19	331.0	66.0	28570.0
172	2020-06-20	-148.0	47.0	29875.0
173	2020-06-21	264.0	49.0	24581.0
174	2020-06-22	224.0	24.0	16152.0
175	2020-06-23	221.0	23.0	23225.0

In [55]:
```
covid_df.at[172, 'new_cases'] = (covid_df.at[171, 'new_cases'] + covid_df.at[173,
```

Working with date

In [56]:
```
covid_df.date
```

Out[56]:
```
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
```

In [57]:
```
covid_df['date'] = pd.to_datetime(covid_df.date)
covid_df['date']
```

Out[57]:
```
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]
```

In [58]:
```
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
```

Out[58]:

	date	new_cases	new_deaths	new_tests	year	month	day	weekday
0	2019-12-31	0.0	0.0	NaN	2019	12	31	1
1	2020-01-01	0.0	0.0	NaN	2020	1	1	2
2	2020-01-02	0.0	0.0	NaN	2020	1	2	3
3	2020-01-03	0.0	0.0	NaN	2020	1	3	4
4	2020-01-04	0.0	0.0	NaN	2020	1	4	5
...	...	...	...	...	...	...	...	...
243	2020-08-30	1444.0	1.0	53541.0	2020	8	30	6
244	2020-08-31	1365.0	4.0	42583.0	2020	8	31	0
245	2020-09-01	996.0	6.0	54395.0	2020	9	1	1
246	2020-09-02	975.0	8.0	NaN	2020	9	2	2
247	2020-09-03	1326.0	6.0	NaN	2020	9	3	3

248 rows × 8 columns

In [59]:
```
# 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
```

Out[59]:
```
new_cases       29073.0
new_deaths       5658.0
new_tests     1078720.0
dtype: float64
```

In [60]:
```
type(covid_may_totals)
```

Out[60]:
```
pandas.core.series.Series
```

In [61]:
```
covid_df[covid_df.month == 5][['new_cases', 'new_deaths', 'new_tests']].sum()
```

Out[61]:
```
new_cases       29073.0
new_deaths       5658.0
new_tests     1078720.0
dtype: float64
```

In [62]:
```
# Overall average
covid_df.new_cases.mean()
```

Out[62]:
```
1096.6149193548388
```

In [63]:
```
# Average for Sundays
covid_df[covid_df.weekday == 6].new_cases.mean()
```

Out[63]:
```
1247.2571428571428
```

Grouping and aggregation

In [64]:
```
covid_month_df = covid_df.groupby('month')[['new_cases', 'new_deaths', 'new_tests
covid_month_df
```

Out[64]:

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	8217.5	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

In [65]:
```
covid_month_mean_df = covid_df.groupby('month')[['new_cases', 'new_deaths', 'new_
covid_month_mean_df
```

Out[65]:

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	273.916667	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

In [66]:
```
covid_df['total_cases'] = covid_df.new_cases.cumsum()
```

In [67]:
```
covid_df['total_deaths'] = covid_df.new_deaths.cumsum()
```

In [68]:
```
covid_df['total_tests'] = covid_df.new_tests.cumsum() + initial_tests
covid_df
```

Out[68]:

	date	new_cases	new_deaths	new_tests	year	month	day	weekday	total_cases	total_de
0	2019-12-31	0.0	0.0	NaN	2019	12	31	1	0.0	
1	2020-01-01	0.0	0.0	NaN	2020	1	1	2	0.0	
2	2020-01-02	0.0	0.0	NaN	2020	1	2	3	0.0	
3	2020-01-03	0.0	0.0	NaN	2020	1	3	4	0.0	
4	2020-01-04	0.0	0.0	NaN	2020	1	4	5	0.0	
...	...	...	...	...	...	...	...	...	...	
243	2020-08-30	1444.0	1.0	53541.0	2020	8	30	6	267298.5	354
244	2020-08-31	1365.0	4.0	42583.0	2020	8	31	0	268663.5	354
245	2020-09-01	996.0	6.0	54395.0	2020	9	1	1	269659.5	354
246	2020-09-02	975.0	8.0	NaN	2020	9	2	2	270634.5	354
247	2020-09-03	1326.0	6.0	NaN	2020	9	3	3	271960.5	354

248 rows × 11 columns

In [ ]:
```

```