DATA 311 - Fall 2020

Final Project - due Tuesday, Nov 24 by midnight

J.Mo Yang

In [59]:
```
import pandas as pd
import sqlite3
```

In [60]:
```
conn= sqlite3.connect('Final.db')
curs = conn.cursor()
curs.execute("PRAGMA foreign_keys=ON;")
```

Out[60]:
```
<sqlite3.Cursor at 0x7f6102e7e180>
```

---

1) Which state(s) were the first to issue a state of emergency, and how many positive test cases had been reported in those state(s) at that time?

Return three columns:

- State
- Number of positive tests
- Date of emergency declaration

In [61]:
```
curs.execute("DROP VIEW IF EXISTS vFirstStEM")
curs.execute("""CREATE VIEW vFirstStEm AS
                    SELECT State, StateEmergency
                    FROM tEmergency
                    ORDER BY StateEmergency ASC
                    LIMIT 1;""")
```

Out[61]:
```
<sqlite3.Cursor at 0x7f6102e7e180>
```

In [62]:
```
pd.read_sql("""SELECT State, Positive, StateEmergency AS Date_of_Emergency
                FROM vFirstStEm
                JOIN tCovidDaily USING (State)
                WHERE Date IS '2020-02-29'
                GROUP BY State;""",conn)
```

Out[62]:

	State	Positive	Date_of_Emergency
0	Washington	18	2020-02-29

---

2) Of states which did declare a state of emergency, which were the last, how many DEATHS had been reported in those state(s) at that time, and, if they did issue a statewide stay at home order, when?

Return 4 columns:

- State
- Number of deaths at the time state of emergency was declared
- Date the state of emergency was declared
- Date stay-at-home order issued (if it exists)

In [63]:
```
curs.execute("DROP VIEW IF EXISTS vLastStEM")
curs.execute("""CREATE VIEW vLastStEm AS
                    SELECT State, StateEmergency
                    FROM tEmergency
                    ORDER BY StateEmergency DESC
                    LIMIT 1;""")
```

Out[63]:
```
<sqlite3.Cursor at 0x7f6102e7e180>
```

In [64]:
```
pd.read_sql("""SELECT State, Death, StateEmergency AS Date_of_Emergency,
                    DateAnnounce AS StayHomeAnnounced
                FROM vLastStEm
                JOIN tCovidDaily USING (State)
                JOIN tStayAtHome USING (State)
                WHERE date IS '2020-03-16'
                GROUP BY State""",conn)
```

Out[64]:

	State	death	Date_of_Emergency	StayHomeAnnounced
0	West Virginia	0	2020-03-16	2020-03-23

---

3) According to the data provided, which state(s) did not issue a stay-at-home order, and how many total deaths have been reported in those state(s)?

Return two columns:

- State
- Number of deaths (as of Nov 15)

In [65]:
```
pd.read_sql("""SELECT State, Death
                FROM tState
                JOIN tCovidDaily USING (State)
                LEFT JOIN tStayAtHome USING (state)
                WHERE DateAnnounce IS NULL
                AND date IS '2020-11-15'""",conn)
```

Out[65]:

	State	death
0	Arkansas	2183
1	Iowa	1985
2	Nebraska	779
3	North Dakota	570
4	Oklahoma	1528
5	South Carolina	4112
6	South Dakota	644

	State	death
7	Utah	718
8	Wyoming	144

---

4) Repeat the previous question, but this time look at states that did issue a stay-at-home order

Return three columns:

- State
- Number of deaths (as of Nov 15)
- Date stay-at-home order announced

In [66]:
```
pd.read_sql("""SELECT State, Death, DateAnnounce
                FROM tStayAtHome
                JOIN tCovidDaily USING (State)
                WHERE date IS '2020-11-15'
                ORDER BY DateAnnounce ASC;""",conn)
```

Out[66]:

	State	death	DateAnnounce
0	California	18253	2020-03-19
1	Connecticut	4737	2020-03-20
2	Illinois	11162	2020-03-20
3	New Jersey	16566	2020-03-20
4	New York	26133	2020-03-20
5	Delaware	736	2020-03-22
6	Kentucky	1661	2020-03-22
7	Louisiana	6132	2020-03-22
8	Ohio	5722	2020-03-22
9	Hawaii	222	2020-03-23
10	Indiana	4910	2020-03-23
11	Massachusetts	10329	2020-03-23
12	Michigan	8376	2020-03-23
13	New Mexico	1208	2020-03-23
14	Oregon	761	2020-03-23
15	Pennsylvania	9312	2020-03-23
16	Washington	2519	2020-03-23
17	West Virginia	582	2020-03-23
18	Vermont	59	2020-03-24
19	Wisconsin	2751	2020-03-24

	State	death	DateAnnounce
20	Idaho	759	2020-03-25
21	Minnesota	2905	2020-03-25
22	Colorado	2234	2020-03-26
23	Montana	520	2020-03-26
24	New Hampshire	499	2020-03-26
25	Alaska	98	2020-03-27
26	North Carolina	4806	2020-03-27
27	Kansas	1256	2020-03-28
28	Rhode Island	1254	2020-03-28
29	Arizona	6302	2020-03-30
30	District of Columbia	660	2020-03-30
31	Maryland	4302	2020-03-30
32	Tennessee	3893	2020-03-30
33	Virginia	3800	2020-03-30
34	Maine	165	2020-03-31
35	Mississippi	3543	2020-03-31
36	Texas	19559	2020-03-31
37	Florida	17734	2020-04-01
38	Nevada	1909	2020-04-01
39	Georgia	8957	2020-04-02
40	Alabama	3248	2020-04-03
41	Missouri	3374	2020-04-03

---

5) Return the following statistics for Virginia:

- Total number of postive cases reported
- Total number of deaths
- Total number of deaths per capita
- Mortality rate, estimated by: Number of deaths / number of positive cases

Hint: Beware of data types, integer conversion etc. The answers are probably not zero.

In [67]:
```
pd.read_sql("""SELECT State, Positive, Death, (1.0*Death/Pop19)
                          AS DeathsPerCapita, (1.0*Death/Positive) AS MortRate
               FROM tCovidDaily
               JOIN tState USING (State)
               JOIN tPopDensity USING (State)
```

```
                WHERE State IS 'Virginia'
                AND date IS '2020-11-15';""",conn)
```

Out[67]:

	State	Positive	death	DeathsPerCapita	MortRate
0	Virginia	201960	3800	0.000445	0.018816

---

6) Which state has had the most deaths per capita as of Nov 15?

Return:

- State
- Number of deaths
- Population
- Population per square mile
- Number of deaths per capita
- Mortality rate, estimated as

Hint: I made a view first, which shortened up the SQL here quite a bit

In [68]:

```
pd.read_sql("""SELECT State, Death, Pop19, (1.0*death/Pop19) AS DeathsPerCapita,
                        (1.0*Pop19/LandSqMi) PopPerSqMi, (1.0*death/positive) AS MortRa
                FROM tCovidDaily
                JOIN tPopDensity USING (State)
                WHERE date IS '2020-11-15'
                ORDER BY DeathsPerCapita DESC
                LIMIT 1""",conn)
```

Out[68]:

	State	death	Pop19	DeathsPerCapita	PopPerSqMi	MortRate
0	New Jersey	16566	8882190	0.001865	1207.767785	0.059318

---

7) Repeat the previous question, but this time for the state with the fewest deaths per capita as of Nov 15

In [69]:

```
pd.read_sql("""SELECT State, Death, Pop19, (1.0*death/Pop19) AS DeathsPerCapita,
                        (1.0*Pop19/LandSqMi) PopPerSqMi, (1.0*death/positive) AS MortRa
                FROM tCovidDaily
                JOIN tPopDensity USING (State)
                WHERE Date IS '2020-11-15'
                ORDER BY DeathsPerCapita ASC
                LIMIT 1""",conn)
```

Out[69]:

	State	death	Pop19	DeathsPerCapita	PopPerSqMi	MortRate
0	Vermont	59	623989	0.000095	67.702291	0.020422

---

8) For the entire US (i.e. the sum of all 50 states + Washington DC):

Get the daily number (not the running total) of positive cases, deaths, and tests reported

Return:

- Date
- The number of new positive tests reported per day
- The number of new deaths reported per day
- The number of new tests performed per day

Order the results by date, ascending

In [70]:
```
pd.read_sql("""SELECT Date, SUM(PositiveInc) AS PositivePerDay,
                          SUM(DeathInc) AS DeathsPerDay, SUM(TotalTResultsInc) AS TestPer
                FROM tCovidDaily
                JOIN tState USING (State)
                GROUP BY Date
                ORDER BY Date ASC;""", conn)
```

Out[70]:

	Date	PositivePerDay	DeathsPerDay	TestPerDay
0	2020-01-22	0	0	0
1	2020-01-23	0	0	1
2	2020-01-24	0	0	0
3	2020-01-25	0	0	0
4	2020-01-26	0	0	0
...	...	...	...	...
294	2020-11-11	144134	1553	1380904
295	2020-11-12	149099	1096	1488194
296	2020-11-13	170051	1297	1682170
297	2020-11-14	162755	1314	1654691
298	2020-11-15	144807	657	1473789

299 rows × 4 columns

BONUS: +2 points. The previous results aren't readily intepretable as a long list of numbers.

Make a plot with date on the x-axis, and daily # of deaths on the y-axis.

In [ ]:
```
```

In [71]:
```
conn.commit()
conn.close()
```

In [1]:
```
# Don't forget to close the database!
```