770 Project

December 24, 2021

0.0.1 Hongge Kang (Nick)

1 How well is restaurant recovering from COVID-19 in 2021?

[1]:
```
import pandas as pd     #load the pandas package and call it pd
import numpy as np      #load the pandas package and call it np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib import gridspec
import datetime as dt               # load datetime and shorten it to dt
import seaborn as sns
```

2 Data cleaning and preping

3 Population Rank

[2]:
```
import sys
import warnings

if not sys.warnoptions:
    warnings.simplefilter("ignore")

pop = pd.read_excel('nst-est2019-02.xlsx',  thousands = "," ,header=4,␣
 ↪skipfooter = 9)
pop.rename(columns = {'Unnamed: 0':'state', 'April 1, 2010 Estimates Base':
 ↪'population' }, inplace=True)
population = pop[['state', 'population']]
population['state'] = population['state'].str.strip('.')
population.sort_values(by = 'population', inplace = True)
c = population['population'][4]
t = population['population'][43]
population.head(3)
```

[2]:
```
                  state  population
50                Wyoming      563775
```

```
8   District of Columbia      601767
45               Vermont      625737
```

4 Restaurant Opening Data

[3]:
```
res_open = pd.read_csv('2020-2021vs2019_Seated_Diner_Data.csv')
res_open.set_index(['Type', 'Name'], inplace=True)
res_open = res_open.transpose()
res_open.columns =res_open.columns.droplevel(level = 0)

res_open.index.name = 'date'
res_open = res_open.drop(['Australia',␣
 ↪'Canada','Germany','Ireland','Mexico','United Kingdom'], axis=1)
res_open = res_open.dropna(axis = 1)

def drop_pct(x):
    return x.str.rstrip('%').astype('float')

col = list(res_open.columns)
res = res_open[col].apply(drop_pct)
res.reset_index(inplace = True)
res['date'] = pd.to_datetime(res['date'])
res.set_index(['date'], inplace=True)
d_c = ['Alberta','Baja California Sur','British Columbia','Jalisco', 'Mexico␣
 ↪City','New South Wales','Nuevo Leon','Ontario','Quebec', 'Queensland',␣
 ↪'Quintana Roo','Victoria',
    'Austin', 'Baltimore','Calgary', 'Charlotte', 'Cincinnati', 'Ciudad de␣
 ↪México', 'Columbus','Dublin', 'Edmonton (CA)', 'Fort Lauderdale', 'Hamburg',␣
 ↪'Honolulu',
    'Indianapolis', 'Las Vegas', 'London', 'Louisville','Miami␣
 ↪Beach','Montréal', 'München', 'Naples (US)', 'Nashville','Phoenix',␣
 ↪'Pittsburgh', 'Portland', 'Raleigh', 'San Antonio', 'San Diego',
    'San Pedro Garza García', 'Scottsdale', 'Tampa', 'Toronto', 'Vancouver']
res.drop(d_c, axis = 1, inplace = True)
res = res.loc['2020/2/18':'2021/10/28', :]

res.head(3)
```

[3]:
```
Name        Global  United States  Alabama  Arizona  California  Colorado  \
date
2020-02-18    -1.0            0.0    -14.0      0.0        -2.0       1.0
2020-02-19     3.0            4.0      7.0      1.0         2.0      -8.0
2020-02-20    -1.0            0.0      1.0      8.0         5.0      -4.0

Name        Connecticut  District of Columbia  Florida  Georgia  ...  \
date                                                             ...
```

```
2020-02-18          8.0                  -5.0      0.0      0.0  ...
2020-02-19         26.0                  55.0     -3.0      4.0  ...
2020-02-20          1.0                   8.0     -4.0     -6.0  ...

Name        Minneapolis  New Orleans  New York  New York  Orlando  \
date
2020-02-18        -14.0         -9.0      -1.0       1.0     -6.0
2020-02-19         27.0          1.0      11.0      11.0     -3.0
2020-02-20        -33.0         14.0       1.0       3.0    -12.0

Name        Philadelphia  San Francisco  Seattle  Washington  Washington
date
2020-02-18          10.0          -15.0      8.0         9.0        -5.0
2020-02-19          55.0          -11.0     11.0         8.0        57.0
2020-02-20          11.0           -4.0      6.0         6.0         9.0

[3 rows x 59 columns]
```

5 Vaccination

[4]:
```
us_state_to_abbrev = {"Alabama": "AL","Alaska": "AK","Arizona": "AZ","Arkansas":
 ↪ "AR","California": "CA","Colorado": "CO","Connecticut": "CT",
"Delaware": "DE","Florida": "FL", "Georgia": "GA", "Hawaii": "HI", "Idaho":␣
 ↪"ID", "Illinois": "IL", "Indiana": "IN", "Iowa": "IA",
"Kansas": "KS", "Kentucky": "KY", "Louisiana": "LA", "Maine": "ME", "Maryland":␣
 ↪"MD", "Massachusetts": "MA", "Michigan": "MI", "Minnesota": "MN",
"Mississippi": "MS","Missouri": "MO","Montana": "MT", "Nebraska": "NE",␣
 ↪"Nevada": "NV", "New Hampshire": "NH", "New Jersey": "NJ",
"New Mexico": "NM","New York": "NY","North Carolina": "NC","North Dakota":␣
 ↪"ND","Ohio": "OH","Oklahoma": "OK","Oregon": "OR","Pennsylvania": "PA",
"Rhode Island": "RI","South Carolina": "SC","South Dakota": "SD","Tennessee":␣
 ↪"TN","Texas": "TX","Utah": "UT","Vermont": "VT","Virginia":␣
 ↪"VA","Washington": "WA","West Virginia": "WV",
"Wisconsin": "WI","Wyoming": "WY","District of Columbia": "DC","American Samoa":
 ↪ "AS","Guam": "GU","Northern Mariana Islands": "MP","Puerto Rico": "PR",
"United States Minor Outlying Islands": "UM", "U.S. Virgin Islands": "VI",␣
 ↪'United States':'US'}

# invert the dictionary
abbrev_to_us_state = dict(map(reversed, us_state_to_abbrev.items()))
```

[5]:
```
vac_r = pd.read_csv('COVID-19_Vaccinations_in_the_United_States_Jurisdiction.
 ↪csv', thousands = ",")
vac = vac_r[['Date', 'Location', 'Series_Complete_Yes']]
vac.set_index(['Date', 'Location'], inplace=True)
vac = vac.stack()
```

```
vac = vac.unstack('Location')
vac.reset_index(inplace = True)
vac['Date'] = pd.to_datetime(vac['Date'])
vac.set_index(['Date'], inplace=True)
vac = vac.sort_index(axis=0)
vac.rename(columns = abbrev_to_us_state, inplace=True)
vac.drop(['level_1','BP2','FM', 'DD2', 'IH2','LTC','MH','RP','VA2'], axis = 1,␣
 ↪inplace = True)

vac.head(3)
```

[5]:
```
Location    Alaska  Alabama  Arkansas  American Samoa  Arizona  California  \
Date
2020-12-13     NaN      NaN       NaN             0.0      NaN         NaN
2020-12-14     0.0      0.0       0.0             0.0      0.0         0.0
2020-12-15     0.0      0.0       0.0             0.0      0.0         0.0

Location    Colorado  Connecticut  District of Columbia  Delaware  ...  Texas  \
Date                                                               ...
2020-12-13       NaN          NaN                   NaN       NaN  ...    NaN
2020-12-14       0.0          0.0                   0.0       0.0  ...    0.0
2020-12-15       0.0          0.0                   0.0       0.0  ...    0.0

Location    United States  Utah  Virginia  U.S. Virgin Islands  Vermont  \
Date
2020-12-13            0.0   NaN       NaN                  0.0      NaN
2020-12-14            0.0   0.0       0.0                  0.0      0.0
2020-12-15            0.0   0.0       0.0                  0.0      0.0

Location    Washington  Wisconsin  West Virginia  Wyoming
Date
2020-12-13         NaN        NaN            NaN      NaN
2020-12-14         0.0        0.0            0.0      0.0
2020-12-15         0.0        0.0            0.0      0.0

[3 rows x 57 columns]
```

[6]:
```
col = list(res.columns)
vac_sum = vac.loc['2021-10-29'].sort_values(axis = 0)
column = list(vac_sum.index)
top_5_vac = column[-7:-1]
top_5_vac.remove('New York')
bottom_5_vac =[]
for state in column:
    if len(bottom_5_vac) != 5:
        if state in col:
            bottom_5_vac.append(state)
```

[7]:
```
# vaccination

col = list(vac.columns)
vac_pct = vac[col].apply('pct_change')*100
vac_pct.fillna(0, inplace= True)
vac_pct.to_excel('vac.xlsx')

vac_c = pd.read_excel('vac_c.xlsx')
vac_c.set_index(['Date'], inplace=True)
vac_c = vac_c.sort_index(axis=0)
vac_c.head(3)
```

[7]:

Date	Alaska	Alabama	Arkansas	American Samoa	Arizona	California	...	Texas	United States	Utah	Virginia	U.S. Virgin Islands	Vermont	Washington	Wisconsin	West Virginia	Wyoming
2020-02-18	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-02-19	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2020-02-20	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

[3 rows x 57 columns]

6 COVID Daily Test

[8]:
```
covid_o = pd.
 ↪read_csv('United_States_COVID-19_Cases_and_Deaths_by_State_over_Time.csv', ␣
 ↪thousands = ",")

#Total case by state
covid_tc = covid_o[['submission_date', 'state', 'tot_cases']]
```

```
covid_tc.set_index(['submission_date', 'state'], inplace=True)
covid_tc = covid_tc.stack()
covid_tc = covid_tc.unstack('state')
covid_tc.reset_index(inplace = True)
covid_tc['submission_date'] = pd.to_datetime(covid_tc['submission_date'])
covid_tc.set_index(['submission_date'], inplace=True)
covid_tc = covid_tc.sort_index(axis=0)
covid_tc.rename(columns = abbrev_to_us_state, inplace=True)
covid_tc.drop(['level_1','FSM', 'PW', 'RMI'], axis = 1, inplace = True)
covid_tc = covid_tc.loc['2020/2/18':'2021/10/28']

# New case by state
covid_nc = covid_o[['submission_date', 'state', 'new_case']]
covid_nc.set_index(['submission_date', 'state'], inplace=True)
covid_nc = covid_nc.stack()
covid_nc = covid_nc.unstack('state')
covid_nc.reset_index(inplace = True)
covid_nc['submission_date'] = pd.to_datetime(covid_nc['submission_date'])
covid_nc.set_index(['submission_date'], inplace=True)
covid_nc = covid_nc.sort_index(axis=0)
covid_nc.rename(columns = abbrev_to_us_state, inplace=True)
covid_nc.drop(['level_1','FSM', 'PW', 'RMI'], axis = 1, inplace = True)
covid_nc = covid_nc.loc['2020/2/18':'2021/10/28']

# Total death by state
covid_td = covid_o[['submission_date', 'state', 'tot_death']]
covid_td.set_index(['submission_date', 'state'], inplace=True)
covid_td = covid_td.stack()
covid_td = covid_td.unstack('state')
covid_td.reset_index(inplace = True)
covid_td['submission_date'] = pd.to_datetime(covid_td['submission_date'])
covid_td.set_index(['submission_date'], inplace=True)
covid_td = covid_td.sort_index(axis=0)
covid_td.rename(columns = abbrev_to_us_state, inplace=True)
covid_td.drop(['level_1','FSM', 'PW', 'RMI'], axis = 1, inplace = True)
covid_td = covid_td.loc['2020/2/18':'2021/10/28']

# New death by state
covid_nd = covid_o[['submission_date', 'state', 'new_death']]
covid_nd.set_index(['submission_date', 'state'], inplace=True)
covid_nd = covid_nd.stack()
covid_nd = covid_nd.unstack('state')
covid_nd.reset_index(inplace = True)
covid_nd['submission_date'] = pd.to_datetime(covid_nd['submission_date'])
covid_nd.set_index(['submission_date'], inplace=True)
covid_nd = covid_nd.sort_index(axis=0)
covid_nd.rename(columns = abbrev_to_us_state, inplace=True)
```

```
covid_nd.drop(['level_1','FSM', 'PW', 'RMI'], axis = 1, inplace = True)
covid_nd = covid_nd.loc['2020/2/18':'2021/10/28']

covid_tc.head(3)
```

[8]:
```
state            Alaska  Alabama  Arkansas  American Samoa  Arizona  \
submission_date
2020-02-18            0       23         0               0        1
2020-02-19            0       23         0               0        1
2020-02-20            0       24         0               0        1

state            California  Colorado  Connecticut  District of Columbia  \
submission_date
2020-02-18              590         0            0                     0
2020-02-19              605         0            0                     0
2020-02-20              617         0            0                     0

state            Delaware  ...  Tennessee  Texas  Utah  Virginia  \
submission_date           ...
2020-02-18              0  ...          0      0     0         0
2020-02-19              0  ...          0      0     0         0
2020-02-20              0  ...          0      0     0         0

state            U.S. Virgin Islands  Vermont  Washington  Wisconsin  \
submission_date
2020-02-18                         0        0           1          0
2020-02-19                         0        0           1          0
2020-02-20                         0        0           1          0

state            West Virginia  Wyoming
submission_date
2020-02-18                   0        0
2020-02-19                   0        0
2020-02-20                   0        0

[3 rows x 57 columns]
```

[9]:
```
covid_tc = covid_o[['submission_date', 'state', 'tot_cases']]
covid_tc.set_index(['submission_date', 'state'], inplace=True)
covid_tc = covid_tc.stack()
covid_tc = covid_tc.unstack('state')
covid_tc.reset_index(inplace = True)
covid_tc['submission_date'] = pd.to_datetime(covid_tc['submission_date'])
covid_tc.set_index(['submission_date'], inplace=True)
covid_tc = covid_tc.sort_index(axis=0)
covid_tc.rename(columns = abbrev_to_us_state, inplace=True)
covid_tc.drop(['level_1','FSM', 'PW', 'RMI'], axis = 1, inplace = True)
covid_tc.head(3)
```

[9]:
```
state            Alaska  Alabama  Arkansas  American Samoa  Arizona  \
submission_date
2020-01-22            0        7         0               0        0
2020-01-23            0        7         0               0        0
2020-01-24            0        7         0               0        0

state            California  Colorado  Connecticut  District of Columbia  \
submission_date
2020-01-22                0         0            0                     0
2020-01-23                0         0            0                     0
2020-01-24                0         0            0                     0

state            Delaware  ...  Tennessee  Texas  Utah  Virginia  \
submission_date            ...
2020-01-22              0  ...          0      0     0         0
2020-01-23              0  ...          0      0     0         0
2020-01-24              0  ...          0      0     0         0

state            U.S. Virgin Islands  Vermont  Washington  Wisconsin  \
submission_date
2020-01-22                         0        0           0          0
2020-01-23                         0        0           1          0
2020-01-24                         0        0           1          0

state            West Virginia  Wyoming
submission_date
2020-01-22                   0        0
2020-01-23                   0        0
2020-01-24                   0        0

[3 rows x 57 columns]
```

[10]:
```python
# Top 5 States and bottom 5 States in total case
col = list(res.columns)
covid_sum = covid_tc.loc['2021-10-28'].sort_values(axis = 0)
column = list(covid_sum.index)
top_5_tc = []
for s in reversed(column):
    if len(top_5_tc) != 5:
        if state in col:
            top_5_tc.append(s)

bottom_5_tc =[]
for state in column:
    if len(bottom_5_tc) != 5:
        if state in col:
            bottom_5_tc.append(state)
```

```
p_t5= population[ (population['state'].isin(top_5_tc))]
p_b5 = population[ (population['state'].isin(bottom_5_tc))]
p_t5.set_index('state',inplace = True)
p_b5.set_index('state',inplace = True)
r_t5 = ['Georgia','Illinois','Florida', 'Texas','California' ]
r_b5 = ['New Mexico','Nebraska', 'Hawaii', 'Rhode Island','District of␣
 ↪Columbia']
```

[11]:
```
print(top_5_vac)
print(top_5_tc)
print(bottom_5_vac)
print(bottom_5_tc)
```

```
['Illinois', 'Pennsylvania', 'Florida', 'Texas', 'California']
['California', 'Texas', 'Florida', 'Illinois', 'Georgia']
['District of Columbia', 'Rhode Island', 'Hawaii', 'Nebraska', 'New Mexico']
['District of Columbia', 'Hawaii', 'Rhode Island', 'New Mexico', 'Nebraska']
```

[12]:
```
covid_us = pd.read_excel('us.xlsx', skipfooter=26)
covid_us['Date'] = pd.to_datetime(covid_us['Date'])
covid_us.set_index(['Date'], inplace=True)
covid_us = covid_us.sort_index(axis=0)
peak = covid_us['New Cases'].max()
covid_us.head(3)
```

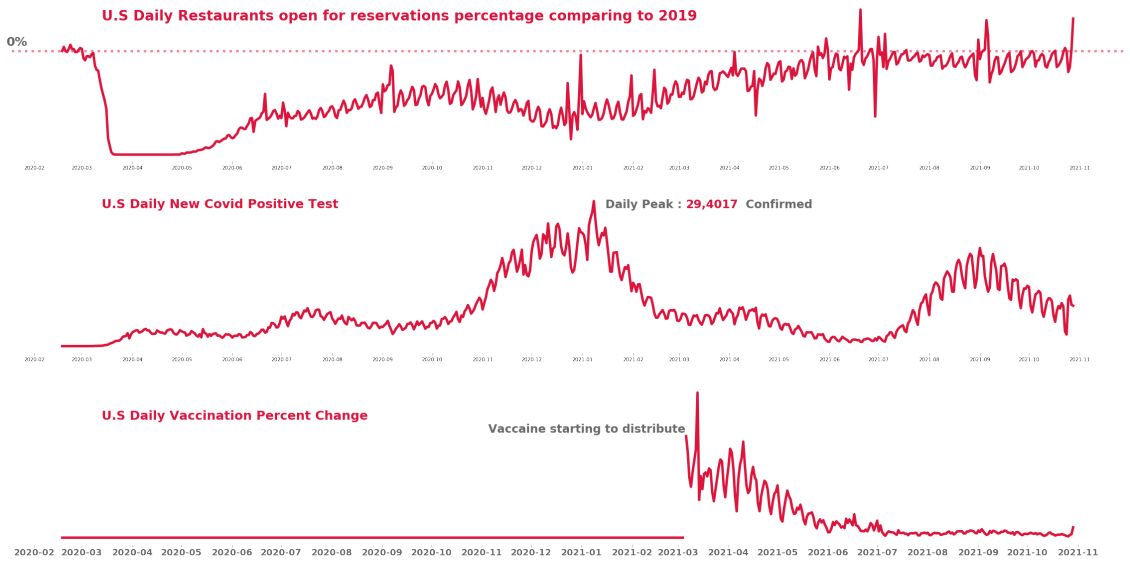
[12]:

Date	State	New Cases	7-Day Moving Avg	Historic Cases
2020-02-18	United States	5	1	0
2020-02-19	United States	3	1	0
2020-02-20	United States	1	1	0

7 Visualization

[13]:
```
fig, ax = plt.subplots(3, 1, figsize=(40,20))

# The fist plot
ax[0].plot(res.index, res['United States'], color= 'crimson', linewidth=5)
ax[0].axhline(y=0, color='crimson', alpha = 0.5, linewidth=5, linestyle = ':')
ax[0].text('2020/3/13', 30 , 'U.S Daily Restaurants open for reservations␣
 ↪percentage comparing to 2019',
           fontsize=28,color= 'crimson',  weight='bold')
ax[0].text('2020/1/15', 5 , '0%', fontsize=25 ,color= 'dimgrey', weight='bold')
plt.xticks(fontsize= 15)
# The second plot

ax[1].plot(covid_us.index, covid_us['New Cases'], color= 'crimson',linewidth=5)
```

```
ax[1].text('2020/3/13', 280000 , 'U.S Daily New Covid Positive Test',
           fontsize=25,color= 'crimson',  weight='bold')
ax[1].text('2021/1/15', 280000 , 'Daily Peak :                Confirmed',
           fontsize=23,color= 'dimgrey',   weight='bold')
ax[1].text('2021/3/5', 280000 , '29,4017',
           fontsize=23,color= 'crimson',   weight='bold')
#The third plot

ax[2].plot(vac_c.index, vac_c['United States'], color= 'crimson',linewidth=5)
ax[2].text('2020/3/13', 4.5 , 'U.S Daily Vaccination Percent Change',
           fontsize=25,color= 'crimson',  weight='bold')
ax[2].text('2020/11/5', 4 , 'Vaccaine starting to distribute',
           fontsize=23,color= 'dimgrey',   weight='bold')

for a in ax:

    a.spines['top'].set_visible(False)
    a.spines['right'].set_visible(False)
    a.spines['bottom'].set_visible(False)
    a.spines['left'].set_visible(False)
    a.set_yticks([])
    a.xaxis.set_major_locator(mdates.MonthLocator())


plt.xticks(fontsize=18, weight='bold', color= 'dimgrey')


plt.savefig('fig_1.png')
plt.show()
```

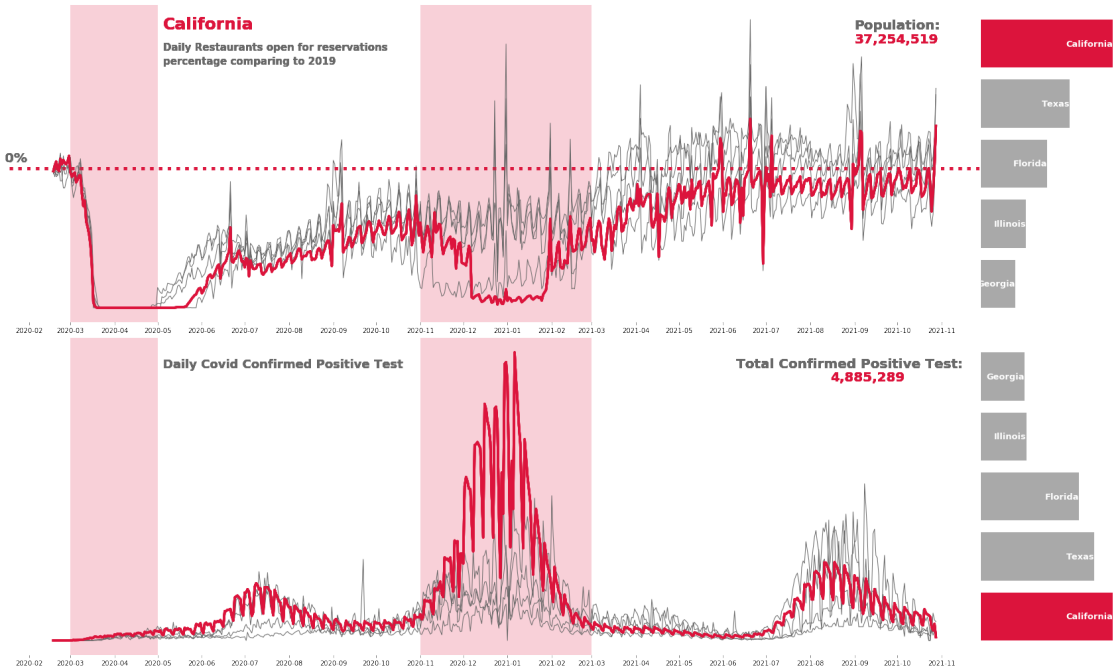


[14]:
```
top_5_tc[0]
```

[14]:
```
'California'
```

[15]:
```
# California


fig = plt.figure(figsize=(30,18))

spec = gridspec.GridSpec(ncols=2, nrows=2, width_ratios=[7,1], wspace=0.
 ↪001,hspace=0.05)
ax0 = fig.add_subplot(spec[0])
ax1 = fig.add_subplot(spec[1])
ax2 = fig.add_subplot(spec[2])
ax3 = fig.add_subplot(spec[3])

y = p_t5['population']/1000000
y1 = [covid_tc.loc['2021/10/28']['California']/1000000,covid_tc.loc['2021/10/
 ↪28']['Texas']/1000000
,covid_tc.loc['2021/10/28']['Florida']/1000000, covid_tc.loc['2021/10/
 ↪28']['Illinois']/1000000
      ,covid_tc.loc['2021/10/28']['Georgia']/1000000]

for state in top_5_tc:
    s = top_5_tc[0]
    ax0.plot(res.index, res[state],color='dimgrey', linewidth=1)
    ax0.plot(res.index, res[s],color='crimson', linewidth=3.5)
    ax0.axhline(y=0, color='crimson', alpha = 0.5, linewidth=5, linestyle = ':')
    ax0.text('2020/5/5', 100 , 'California',
           fontsize=25,color= 'crimson',  weight='bold')
    ax0.text('2020/5/5', 85 , 'Daily Restaurants open for reservations',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/5/5', 75 , 'percentage comparing to 2019',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/1/15', 5 , '0%', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/1', 100 , 'Population:', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/1', 90 , '37,254,519', fontsize=20 ,color= 'crimson',␣
 ↪weight='bold')
    ax1.barh(p_t5.index,p_t5['population']/1000000, color='darkgrey')
    ax1.barh(s,p_t5.loc[s][0]/1000000, color='crimson')
    ax2.plot(covid_nc.index, covid_nc[state],color='dimgrey' , linewidth=1)
    ax2.plot(covid_nc.index, covid_nc[s],color='crimson' , linewidth=3.5)
    ax2.text('2020/5/5', 60000 , 'Daily Covid Confirmed Positive Test',
           fontsize=18,color= 'dimgrey',  weight='bold')
    ax2.text('2021/6/10', 60000 , 'Total Confirmed Positive Test:',
```

```
            fontsize=20,color= 'dimgrey',  weight='bold')
    ax2.text('2021/8/15', 57000 , '4,885,289',
            fontsize=20,color= 'crimson',  weight='bold')
    ax3.barh(state,covid_tc[state]/1000000, color='darkgrey')
    ax3.barh(s,covid_tc[s]/1000000, color='crimson')

for a in [ax0, ax1, ax2, ax3]:

    a.spines['top'].set_visible(False)
    a.spines['right'].set_visible(False)
    a.spines['bottom'].set_visible(False)
    a.spines['left'].set_visible(False)
    a.set_yticks([])
    a.xaxis.set_major_locator(mdates.MonthLocator())

for i, v in enumerate(y):
    ax1.text(v , i, r_t5[i], color='white', fontweight='bold', fontsize=13,␣
 ↪ha='right', va='center')

for i, v in enumerate(y1):
    ax3.text(v , i, top_5_tc[i] , color='white', fontweight='bold',␣
 ↪fontsize=13, ha='right', va='center')

ax0.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor='crimson',␣
 ↪alpha=0.2)
ax0.axvspan(dt.date(2020, 11, 1), dt.date(2021, 3, 1), facecolor='crimson',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor='crimson',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2020, 11, 1), dt.date(2021, 3, 1), facecolor='crimson',␣
 ↪alpha=0.2)

plt.savefig('fig_2.png')
plt.show()
```



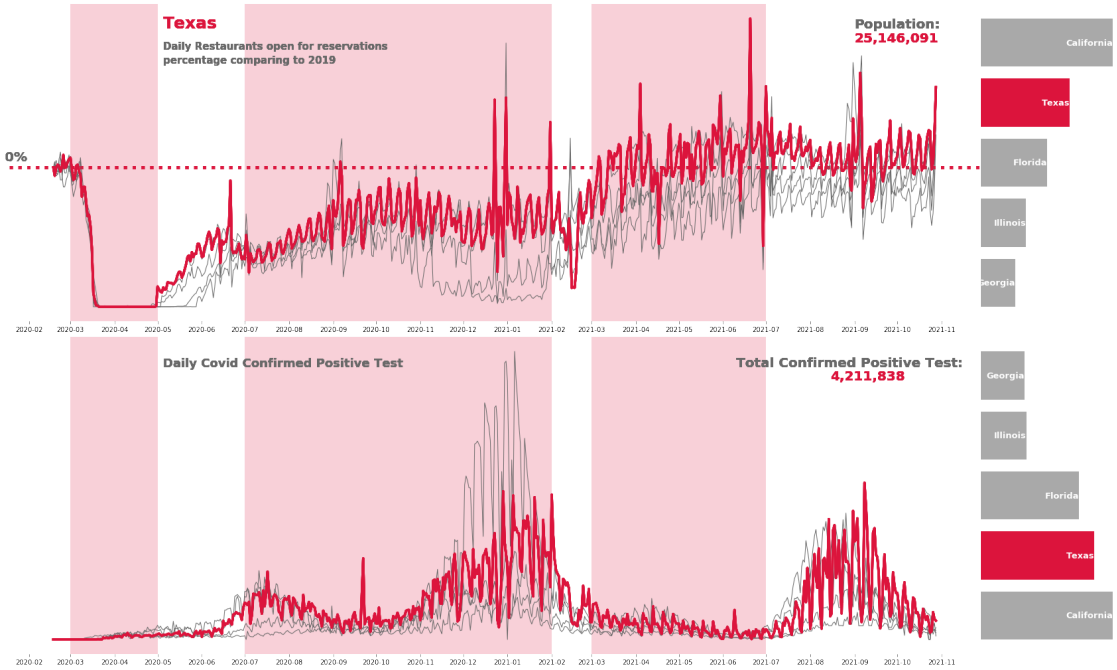
[16]:
```
top_5_tc[1]
```

[16]:
```
'Texas'
```

[17]:
```
# Texas

fig = plt.figure(figsize=(30,18))

spec = gridspec.GridSpec(ncols=2, nrows=2, width_ratios=[7,1], wspace=0.
 ↪001,hspace=0.05)
ax0 = fig.add_subplot(spec[0])
ax1 = fig.add_subplot(spec[1])
ax2 = fig.add_subplot(spec[2])
ax3 = fig.add_subplot(spec[3])

y = p_t5['population']/1000000
y1 = [covid_tc.loc['2021/10/28']['California']/1000000,covid_tc.loc['2021/10/
 ↪28']['Texas']/1000000
,covid_tc.loc['2021/10/28']['Florida']/1000000, covid_tc.loc['2021/10/
 ↪28']['Illinois']/1000000
     ,covid_tc.loc['2021/10/28']['Georgia']/1000000]

for state in top_5_tc:
    s = top_5_tc[1]
    ax0.plot(res.index, res[state],color='dimgrey', linewidth=1)
    ax0.plot(res.index, res[s],color='crimson', linewidth=3.5)
    ax0.axhline(y=0, color='crimson', alpha = 0.5, linewidth=5, linestyle = ':')
```

```
    ax0.text('2020/5/5', 100 , 'Texas',
           fontsize=25,color= 'crimson',  weight='bold')
    ax0.text('2020/5/5', 85 , 'Daily Restaurants open for reservations',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/5/5', 75 , 'percentage comparing to 2019',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/1/15', 5 , '0%', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/1', 100 , 'Population:', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/1', 90 , '25,146,091', fontsize=20 ,color= 'crimson',␣
 ↪weight='bold')
    ax1.barh(p_t5.index,p_t5['population']/1000000, color='darkgrey')
    ax1.barh(s,p_t5.loc[s][0]/1000000, color='crimson')
    ax2.plot(covid_nc.index, covid_nc[state],color='dimgrey' , linewidth=1)
    ax2.plot(covid_nc.index, covid_nc[s],color='crimson' , linewidth=3.5)
    ax2.text('2020/5/5', 60000 , 'Daily Covid Confirmed Positive Test',
           fontsize=18,color= 'dimgrey',  weight='bold')
    ax2.text('2021/6/10', 60000 , 'Total Confirmed Positive Test:',
           fontsize=20,color= 'dimgrey',  weight='bold')
    ax2.text('2021/8/15', 57000 , '4,211,838',
           fontsize=20,color= 'crimson',  weight='bold')
    ax3.barh(state,covid_tc[state]/1000000, color='darkgrey')
    ax3.barh(s,covid_tc[s]/1000000, color='crimson')
for a in [ax0, ax1, ax2, ax3]:

    a.spines['top'].set_visible(False)
    a.spines['right'].set_visible(False)
    a.spines['bottom'].set_visible(False)
    a.spines['left'].set_visible(False)
    a.set_yticks([])
    a.xaxis.set_major_locator(mdates.MonthLocator())

for i, v in enumerate(y):
    ax1.text(v , i, r_t5[i], color='white', fontweight='bold', fontsize=13,␣
 ↪ha='right', va='center')

for i, v in enumerate(y1):
    ax3.text(v , i, top_5_tc[i] , color='white', fontweight='bold',␣
 ↪fontsize=13, ha='right', va='center')

ax0.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor='crimson',␣
 ↪alpha=0.2)
ax0.axvspan(dt.date(2020, 7, 1), dt.date(2021, 2, 1), facecolor='crimson',␣
 ↪alpha=0.2)
```

```
ax0.axvspan(dt.date(2021, 3, 1), dt.date(2021, 7, 1), facecolor='crimson',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor= 'crimson',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2020, 7, 1), dt.date(2021, 2, 1), facecolor='crimson',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2021, 3, 1), dt.date(2021, 7, 1), facecolor='crimson',␣
 ↪alpha=0.2)

plt.savefig('fig_3.png')
plt.show()
```



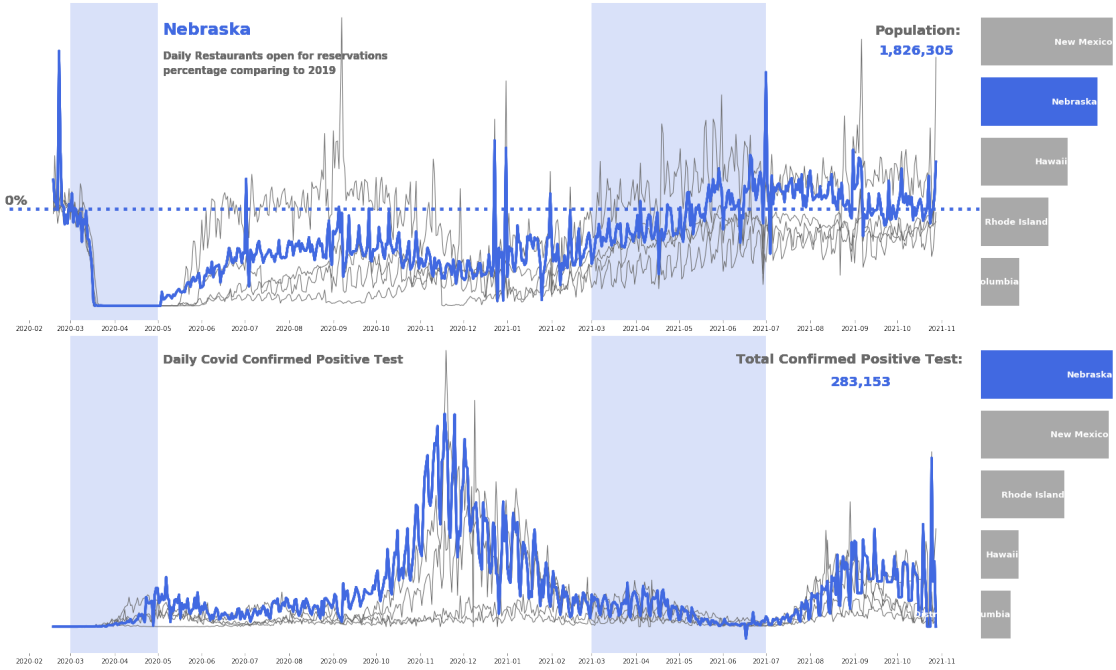
[18]:
```
bottom_5_tc[4]
```

[18]:
```
'Nebraska'
```

[19]:
```
#Nebraska

fig = plt.figure(figsize=(30,18))

spec = gridspec.GridSpec(ncols=2, nrows=2, width_ratios=[7,1], wspace=0.
 ↪001,hspace=0.05)
ax0 = fig.add_subplot(spec[0])
ax1 = fig.add_subplot(spec[1])
ax2 = fig.add_subplot(spec[2])
ax3 = fig.add_subplot(spec[3])
```

```
y = p_b5['population']/100000
y1 = [covid_tc.loc['2021/10/28']['District of Columbia']/10000,covid_tc.
 ↪loc['2021/10/28']['Hawaii']/10000
,covid_tc.loc['2021/10/28']['Rhode Island']/10000, covid_tc.loc['2021/10/
 ↪28']['New Mexico']/10000
      ,covid_tc.loc['2021/10/28']['Nebraska']/10000]
i = 0
for state in bottom_5_tc:
    s = bottom_5_tc[4]
    ax0.plot(res.index, res[state],color='dimgrey', linewidth=1)
    ax0.plot(res.index, res[s],color='royalblue', linewidth=3.5)
    ax0.axhline(y=0, color='royalblue', alpha = 0.5, linewidth=5, linestyle = ':
 ↪')
    ax0.text('2020/5/5', 180 , 'Nebraska',
           fontsize=25,color= 'royalblue',  weight='bold')
    ax0.text('2020/5/5', 155 , 'Daily Restaurants open for reservations',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/5/5', 140 , 'percentage comparing to 2019',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/1/15', 5 , '0%', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/15', 180 , 'Population:', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/18', 160 , '1,826,305', fontsize=20 ,color= 'royalblue',␣
 ↪weight='bold')
    ax1.barh(p_b5.index,y, color='darkgrey')
    ax1.barh(s,y[3], color='royalblue')
    ax2.plot(covid_nc.index, covid_nc[state],color='dimgrey' , linewidth=1)
    ax2.plot(covid_nc.index, covid_nc[s],color='royalblue' , linewidth=3.5)
    ax2.text('2020/5/5', 3500 , 'Daily Covid Confirmed Positive Test',
           fontsize=18,color= 'dimgrey',  weight='bold')
    ax2.text('2021/6/10', 3500 , 'Total Confirmed Positive Test:',
           fontsize=20,color= 'dimgrey',  weight='bold')
    ax2.text('2021/8/15', 3200 , '283,153',
           fontsize=20,color= 'royalblue',  weight='bold')
for i in range(4):
    ax3.barh(bottom_5_tc,y1, color='darkgrey')
    ax3.barh(s,y1[4], color='royalblue')

for a in [ax0, ax1, ax2, ax3]:

    a.spines['top'].set_visible(False)
    a.spines['right'].set_visible(False)
    a.spines['bottom'].set_visible(False)
    a.spines['left'].set_visible(False)
    a.set_yticks([])
```

```
    a.xaxis.set_major_locator(mdates.MonthLocator())

for i, v in enumerate(y):
    ax1.text(v , i,p_b5.index[i], color='white', fontweight='bold',␣
 ↪fontsize=13, ha='right', va='center')

for i, v in enumerate(y1):
    ax3.text(v , i, bottom_5_tc[i] , color='white', fontweight='bold',␣
 ↪fontsize=13, ha='right', va='center')

ax0.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor='royalblue',␣
 ↪alpha=0.2)
ax0.axvspan(dt.date(2021, 3, 1), dt.date(2021, 7, 1), facecolor='royalblue',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor='royalblue',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2021, 3, 1), dt.date(2021, 7, 1), facecolor='royalblue',␣
 ↪alpha=0.2)

plt.savefig('fig_4.png')
plt.show()
```

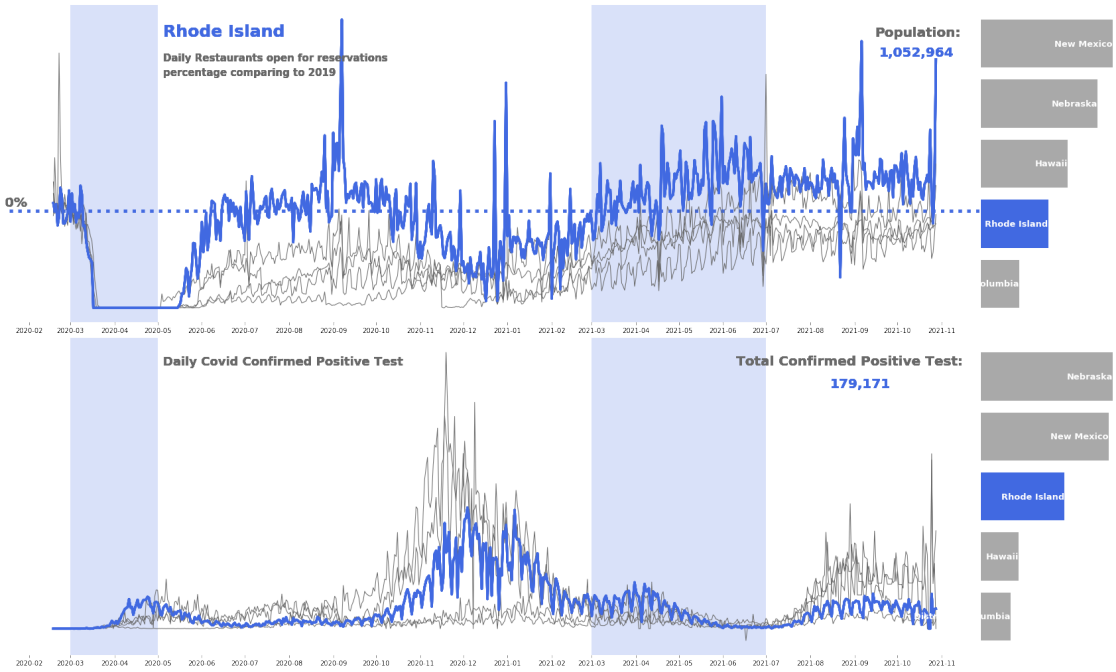


[20]:
```
bottom_5_tc[2]
```

[20]:
```
'Rhode Island'
```

[21]:
```
fig = plt.figure(figsize=(30,18))
spec = gridspec.GridSpec(ncols=2, nrows=2, width_ratios=[7,1], wspace=0.
 ↪001,hspace=0.05)
ax0 = fig.add_subplot(spec[0])
ax1 = fig.add_subplot(spec[1])
ax2 = fig.add_subplot(spec[2])
ax3 = fig.add_subplot(spec[3])

y = p_b5['population']/100000
y1 = [covid_tc.loc['2021/10/28']['District of Columbia']/10000,covid_tc.
 ↪loc['2021/10/28']['Hawaii']/10000
,covid_tc.loc['2021/10/28']['Rhode Island']/10000, covid_tc.loc['2021/10/
 ↪28']['New Mexico']/10000
      ,covid_tc.loc['2021/10/28']['Nebraska']/10000]
i = 0
for state in bottom_5_tc:
    s = bottom_5_tc[2]
    ax0.plot(res.index, res[state],color='dimgrey', linewidth=1)
    ax0.plot(res.index, res[s],color='royalblue', linewidth=3.5)
    ax0.axhline(y=0, color='royalblue', alpha = 0.5, linewidth=5, linestyle = ':
 ↪')
    ax0.text('2020/5/5', 180 , 'Rhode Island',
           fontsize=25,color= 'royalblue',  weight='bold')
    ax0.text('2020/5/5', 155 , 'Daily Restaurants open for reservations',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/5/5', 140 , 'percentage comparing to 2019',
           fontsize=15,color= 'dimgrey',  weight='bold')
    ax0.text('2020/1/15', 5 , '0%', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/15', 180, 'Population:', fontsize=20 ,color= 'dimgrey',␣
 ↪weight='bold')
    ax0.text('2021/9/18', 160 , '1,052,964', fontsize=20 ,color= 'royalblue',␣
 ↪weight='bold')
    ax1.barh(p_b5.index,y, color='darkgrey')
    ax1.barh(s,y[1], color='royalblue')
    ax2.plot(covid_nc.index, covid_nc[state],color='dimgrey' , linewidth=1)
    ax2.plot(covid_nc.index, covid_nc[s],color='royalblue' , linewidth=3.5)
    ax2.text('2020/5/5', 3500 , 'Daily Covid Confirmed Positive Test',
           fontsize=18,color= 'dimgrey',  weight='bold')
    ax2.text('2021/6/10', 3500 , 'Total Confirmed Positive Test:',
           fontsize=20,color= 'dimgrey',  weight='bold')
    ax2.text('2021/8/15', 3200 , '179,171',
           fontsize=20,color= 'royalblue',  weight='bold')
for i in range(4):
    ax3.barh(bottom_5_tc,y1, color='darkgrey')
    ax3.barh(s,y1[2], color='royalblue')
```

```
for a in [ax0, ax1, ax2, ax3]:

    a.spines['top'].set_visible(False)
    a.spines['right'].set_visible(False)
    a.spines['bottom'].set_visible(False)
    a.spines['left'].set_visible(False)
    a.set_yticks([])
    a.xaxis.set_major_locator(mdates.MonthLocator())

for i, v in enumerate(y):
    ax1.text(v , i,p_b5.index[i], color='white', fontweight='bold',␣
 ↪fontsize=13, ha='right', va='center')

for i, v in enumerate(y1):
    ax3.text(v , i, bottom_5_tc[i] , color='white', fontweight='bold',␣
 ↪fontsize=13, ha='right', va='center')

ax0.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor='royalblue',␣
 ↪alpha=0.2)
ax0.axvspan(dt.date(2021, 3, 1), dt.date(2021, 7, 1), facecolor='royalblue',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2020, 3, 1), dt.date(2020, 5, 1), facecolor='royalblue',␣
 ↪alpha=0.2)
ax2.axvspan(dt.date(2021, 3, 1), dt.date(2021, 7, 1), facecolor='royalblue',␣
 ↪alpha=0.2)

plt.savefig('fig_5.png')
plt.show()
```



[22]:
```
fig = plt.figure(figsize=(25,19))


spec = gridspec.GridSpec(ncols=2, nrows=4, width_ratios=[5,1], wspace=0.
 ↪001,hspace=0.05)
ax0 = fig.add_subplot(spec[0])
ax1 = fig.add_subplot(spec[1])
ax2 = fig.add_subplot(spec[2])
ax3 = fig.add_subplot(spec[3])
ax4 = fig.add_subplot(spec[4])
ax5 = fig.add_subplot(spec[5])
ax6 = fig.add_subplot(spec[6])
ax7 = fig.add_subplot(spec[7])


ax0.plot(res.index, res['California'], color= 'seagreen', linewidth=5)
ax0.axhline(y=0, color='seagreen', alpha = 0.8, linewidth=5, linestyle = ':')
ax0.axvline(x= '2021/3/5', color='dimgray', linestyle='-.', linewidth=3.5)
ax0.text('2020/3/13', 30 , 'California',fontsize=20,color= 'seagreen', ␣
 ↪weight='bold')
ax0.text('2020/1/15', 5 , '0%', fontsize=20 ,color= 'dimgrey', weight='bold')
ax0.text('2020/3/13', 15 , 'Mask Order Effective',fontsize=15,color= 'dimgrey',␣
 ↪ weight='bold')
ax0.text('2020/3/13', 50 , 'Daily Restaurants open for reservations percentage␣
 ↪comparing to 2019',
          fontsize=13,color= 'dimgrey',  weight='bold')
```

```
ax0.text('2021/3/20', -80 , 'Vaccination March 5th',    fontsize=15,color=␣
 ↪'seagreen',   weight='bold')
plt.xticks(fontsize= 15)


ax1.plot(vac.index, vac['California'], color= 'seagreen',linewidth=5)
ax1.text('2020/12/13', 45000000 , 'California',fontsize=15,color= 'dimgrey', ␣
 ↪weight='bold')
ax1.text('2020/12/13', 40000000 , 'Fully Vaccinated',fontsize=15,color=␣
 ↪'dimgrey',  weight='bold')
ax1.text('2021/8/1', 40500000 , '64.2%',fontsize=25,color= 'seagreen', ␣
 ↪weight='bold')
ax1.text('2021/3/1', 25000000 , '24,159,427',fontsize=20,color= 'seagreen',  ␣
 ↪weight='bold')



ax2.plot(res.index, res['Texas'], color= 'salmon', linewidth=5)
ax2.axhline(y=0, color='salmon', alpha = 0.8, linewidth=5, linestyle = ':')
ax2.axvline(x= '2021/3/5', color='dimgray', linestyle='-.', linewidth=3.5)
ax2.axvline(x= '2021/3/10', color='dimgrey', linestyle='--', linewidth=3.5)
ax2.text('2020/3/13', 80 , 'Texas',fontsize=20,color= 'salmon',  weight='bold')
ax2.text('2021/3/20', 100 , 'Mask Order Superseded',    fontsize=15,color=␣
 ↪'dimgrey',   weight='bold')
ax2.text('2021/3/20', 80 , 'March 10th',    fontsize=18,color= 'salmon',  ␣
 ↪weight='bold')
ax2.text('2020/1/15', 5 , '0%', fontsize=20 ,color= 'dimgrey', weight='bold')
ax2.text('2021/3/20', -90 , 'Vaccinationn March 5th',    fontsize=15,color=␣
 ↪'seagreen',   weight='bold')
plt.xticks(fontsize= 15)

ax3.plot(vac.index, vac['Texas'], color= 'salmon',linewidth=5)
ax3.text('2020/12/13', 35000000 , 'Texas',fontsize=15,color= 'dimgrey', ␣
 ↪weight='bold')
ax3.text('2020/12/13', 30000000 , 'Fully Vaccinated',fontsize=15,color=␣
 ↪'dimgrey',  weight='bold')
ax3.text('2021/8/1', 30500000 , '55.5%',fontsize=25,color= 'salmon', ␣
 ↪weight='bold')
ax3.text('2021/3/1', 17000000 , '15,452,632',fontsize=20,color= 'salmon',  ␣
 ↪weight='bold')

ax4.plot(res.index, res['Nebraska'], color= 'salmon', linewidth=5)
ax4.axhline(y=0, color='salmon', alpha = 0.8, linewidth=5, linestyle = ':')
ax4.axvline(x= '2021/3/5', color='dimgray', linestyle='-.', linewidth=3.5)
ax4.text('2020/1/15', 8 , '0%', fontsize=20 ,color= 'dimgrey', weight='bold')
```

```
ax4.text('2020/3/13', 135 , 'Nebraska',fontsize=20,color= 'salmon', ␣
 ↪weight='bold')
ax4.text('2020/3/13', 105 , 'No State Mask Order',fontsize=15,color= 'dimgrey',␣
 ↪ weight='bold')
ax4.text('2021/3/20', -90 , 'Vaccination March 5th',    fontsize=15,color=␣
 ↪'seagreen',   weight='bold')
plt.xticks(fontsize= 15)

ax5.plot(vac.index, vac['Nebraska'], color= 'salmon',linewidth=5)
ax5.text('2020/12/13', 2100000 , 'Nebraska',fontsize=15,color= 'dimgrey', ␣
 ↪weight='bold')
ax5.text('2020/12/13', 1805000 , 'Fully Vaccinated',fontsize=15,color=␣
 ↪'dimgrey',  weight='bold')
ax5.text('2021/8/1', 1830000 , '58.3%',fontsize=25,color= 'salmon', ␣
 ↪weight='bold')
ax5.text('2021/3/1', 1200000 , '1,086,941',fontsize=20,color= 'salmon',  ␣
 ↪weight='bold')

ax6.plot(res.index, res['Rhode Island'], color= 'salmon', linewidth=5)
ax6.axhline(y=0, color='salmon', alpha = 0.8, linewidth=5, linestyle = ':')
ax6.axvline(x= '2021/3/5', color='dimgray', linestyle='-.', linewidth=3.5)
ax6.axvline(x= '2021/7/6', color='dimgrey', linestyle='--', linewidth=3.5)
ax6.text('2020/1/15', 5 , '0%', fontsize=20 ,color= 'dimgrey', weight='bold')
ax6.text('2020/3/13', 150 , 'Rhode Island',fontsize=20,color= 'salmon', ␣
 ↪weight='bold')
ax6.text('2021/7/15', 160 , 'Mask Order Superseded',    fontsize=15,color=␣
 ↪'dimgrey',   weight='bold')
ax6.text('2021/7/15', 130 , 'July 6th',    fontsize=18,color= 'salmon',  ␣
 ↪weight='bold')
ax6.text('2021/3/16', -90 , 'Vaccination March 5th',    fontsize=15,color=␣
 ↪'seagreen',   weight='bold')
plt.xticks(fontsize= 15)

ax7.plot(vac.index, vac['Rhode Island'], color= 'salmon',linewidth=5)
ax7.text('2020/12/13', 2100000 , 'Rhode Island',fontsize=15,color= 'dimgrey', ␣
 ↪weight='bold')
ax7.text('2020/12/13', 1850000 , 'Fully Vaccinated',fontsize=15,color=␣
 ↪'dimgrey',  weight='bold')
ax7.text('2021/8/1', 1850000 , '74.1%',fontsize=25,color= 'salmon', ␣
 ↪weight='bold')
ax7.text('2021/3/1', 800000 , '749,548',fontsize=20,color= 'salmon',  ␣
 ↪weight='bold')

for a in [ax0, ax1, ax2, ax3,ax4, ax5, ax6, ax7]:

    a.spines['top'].set_visible(False)
```

```
    a.spines['right'].set_visible(False)
    a.spines['bottom'].set_visible(False)
    a.spines['left'].set_visible(False)
    a.set_yticks([])
    a.set_xticks([])
for a in  [ax1, ax3]:
    a.set_ylim(0, 50000000)
for a in  [ax5, ax7]:
    a.set_ylim(0, 3000000)
ax6.xaxis.set_major_locator(mdates.MonthLocator())




plt.savefig('fig_6.png')
plt.show()
```

[ ]: