

---

```

% Part III: The Effects of Travel

load('COVID_STL.mat');
realDates = dates(3:length(dates));
load('COVIDbyCounty.mat');

% population 1 = St. Louis City + County (given data)
% population 2 = Riverside County, CA (taken from Case Study 1)

% simulate 156 weeks (omit the first two weeks b/c COVIDbyCounty.mat from
% Case Study 1 contains 156 instead of 158 weeks)
cases_STL = cases_STL(3:158);
deaths_STL = deaths_STL(3:158);
POP_RVS = CNTY_CENSUS.POPESTIMATE2021(23);
% calculate raw data from given normalized data
cases_RVS = round(CNTY_COVID(23, :) * POP_RVS / 100000);

d1 = deaths_STL(1) / POP_STL;
r1 = 0;
i1 = cases_STL(1) / POP_STL - d1;
s1 = 1 - i1 - d1;
d2 = 0; % assume no initial deaths since no relevant data is given
r2 = 0;
i2 = cases_RVS(1) / POP_RVS;
s2 = 1 - i2;
x = [s1, i1, r1, d1, s2, i2, r2, d2]';
x1 = x(1:4);
x2 = x(5:8);

activeCases1 = [];
newCases1 = [];
newDeaths1 = [];
% same assumptions for STL as part2.m
for j = 1:length(cases_STL) - 1
    newCases1 = [newCases1, cases_STL(j + 1) - cases_STL(j)];
    if j == 1
        activeCases1 = [activeCases1, cases_STL(j + 1) - cases_STL(j)];
    else
        activeCases1 = [activeCases1, cases_STL(j + 1) - cases_STL(j) ...
            + round(activeCases1(j - 1) * 9 / 10)];
    end
    newDeaths1 = [newDeaths1, deaths_STL(j + 1) - deaths_STL(j)];
end
avgTotalNewCases1 = mean(newCases1) / (POP_STL - mean(cases_STL) + ...
    mean(activeCases1));
totalNewDeaths1 = [];
for j = 1:length(deaths_STL) - 1
    totalNewDeaths1 = [totalNewDeaths1, newDeaths1(j) / activeCases1(j)];
end
avgTotalNewDeaths1 = mean(totalNewDeaths1);

activeCases2 = [];

```

---

---

```

newCases2 = cases_RVS(1:length(cases_RVS) - 1);
for j = 1:length(cases_RVS) - 1
    if j == 1
        activeCases2 = [activeCases2, newCases2(j)];
    else
        activeCases2 = [activeCases2, newCases2(j) + ...
            round(activeCases2(j - 1) * 9 / 10)];
    end
end
cumulativeCases2 = [];
total = 0;
for j = 1:length(cases_RVS)
    total = total + cases_RVS(j);
    cumulativeCases2 = [cumulativeCases2, total];
end
avgTotalNewCases2 = mean(newCases2) / (POP_RVS - ...
    mean(cumulativeCases2) + mean(activeCases2));

% STL matrix
A = [1 - avgTotalNewCases1,      0.09,      0, 0;
     avgTotalNewCases1,          0.9,        0, 0;
     0,                          0.01 - avgTotalNewDeaths1, 1, 0;
     0,                          avgTotalNewDeaths1,    0, 1];

% RVS to STL matrix
B = [0.95,      0.03,      0, 0;
     0.05,      0.95,      0, 0;
     0,          0.01,      1, 0;
     0,          0.01,      0, 1];

% STL to RVS matrix
C = [0.97,      0.01,      0, 0;
     0.03,      0.98,      0, 0;
     0,          0.005,     1, 0;
     0,          0.005,     0, 1];

% RVS matrix
D = [1 - avgTotalNewCases2,      0.1,      0, 0;
     avgTotalNewCases2,          0.8,      0, 0;
     0,                          0.095,     1, 0;
     0,                          0.005,     0, 1];

E = [A, B; C, D];

Y = x;
Y1 = x1;
Y2 = x2;
for j = 2:length(dates) - 1
    x = (E * x) / 2; % w/ traveling
    Y = [Y, x];
    x1 = A * x1; % STL w/o traveling
    Y1 = [Y1, x1];
    x2 = D * x2; % RVS w/o traveling
    Y2 = [Y2, x2];

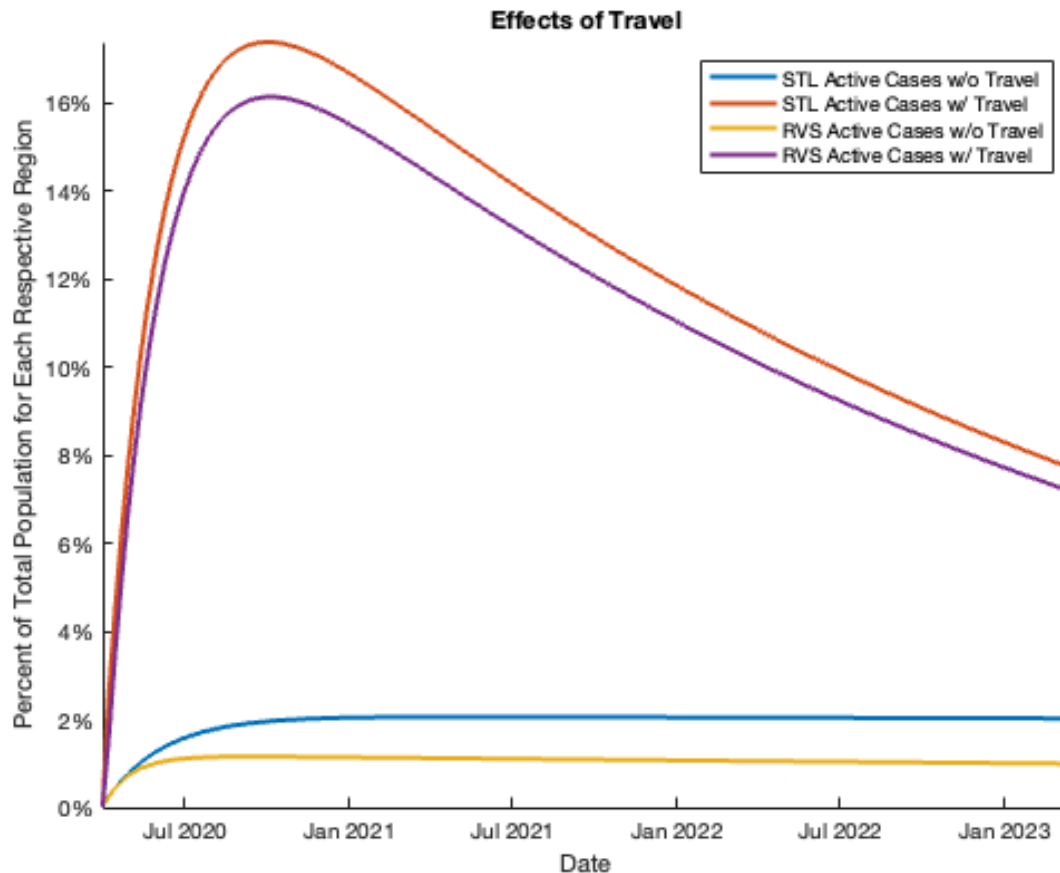
```

---

---

end

```
hold on;
plot(realDates(1:length(dates) - 1), Y1(2, :) * 100, 'LineWidth', 2);
plot(realDates(1:length(dates) - 1), Y(2, :) * 100, 'LineWidth', 2);
plot(realDates(1:length(dates) - 1), Y2(2, :) * 100, 'LineWidth', 2);
plot(realDates(1:length(dates) - 1), Y(6, :) * 100, 'LineWidth', 2);
hold off;
axis tight;
ylim([0 inf]);
title('Effects of Travel');
legend('STL Active Cases w/o Travel', 'STL Active Cases w/ Travel', ...
       'RVS Active Cases w/o Travel', 'RVS Active Cases w/ Travel');
xlabel('Date');
ylabel('Percent of Total Population for Each Respective Region');
ytickformat('percentage');
exportgraphics(gca, 'travel.png');
```



what-if policy on travel restrictions implemented during a surge of new cases in the initial phase (week 36) and continue throughout the end of the Omicron phase (week 105)

```
% two regions: same as above (STL & RVS)
% two phases: same as part2.m (Delta & Omicron)
```

---

```

% pre-policy phase: 3/18/2020 ~ 11/17/2020

avgPreNewCases1 = mean(newCases1(1:35)) / (POP_STL - ...
    mean(cases_STL(1:35)) + mean(activeCases1(1:35)));
preNewDeaths1 = [];
for j = 1:35
    preNewDeaths1 = [preNewDeaths1, newDeaths1(j) / activeCases1(j)];
end
avgPreNewDeaths1 = mean(preNewDeaths1);

preActiveCases2 = [];
preNewCases2 = cases_RVS(1:35);
for j = 1:35
    if j == 1
        preActiveCases2 = [preActiveCases2, preNewCases2(j)];
    else
        preActiveCases2 = [preActiveCases2, preNewCases2(j) + ...
            round(preActiveCases2(j - 1) * 9 / 10)];
    end
end
avgPreNewCases2 = mean(preNewCases2) / (POP_RVS - ...
    mean(cumulativeCases2(1:35)) + mean(preActiveCases2));

x3 = [s1, i1, r1, d1, s2, i2, r2, d2]';

A = [1 - avgPreNewCases1,      0.09,      0, 0;
     avgPreNewCases1,         0.9,        0, 0;
     0,                       0.01 - avgPreNewDeaths1, 1, 0;
     0,                       avgPreNewDeaths1,    0, 1];

D = [1 - avgPreNewCases2,      0.1,      0, 0;
     avgPreNewCases2,         0.8,        0, 0;
     0,                       0.095,      1, 0;
     0,                       0.005,      0, 1];

E = [A, B; C, D];

Y3 = x3;
for j = 2:35
    x3 = (E * x3) / 2;
    Y3 = [Y3, x3];
end

figure;
hold on;
plot(realDates(1:35), Y3(2, :) * 100, 'LineWidth', 2);
plot(realDates(1:35), Y3(6, :) * 100, 'LineWidth', 2);
hold off;
axis tight;
ylim([0 inf]);
title('Prior to Travel Restrictions');
legend('STL Active Cases', 'RVS Active Cases');
xlabel('Date');
ylabel('Percent of Total Population for Each Respective Region');

```

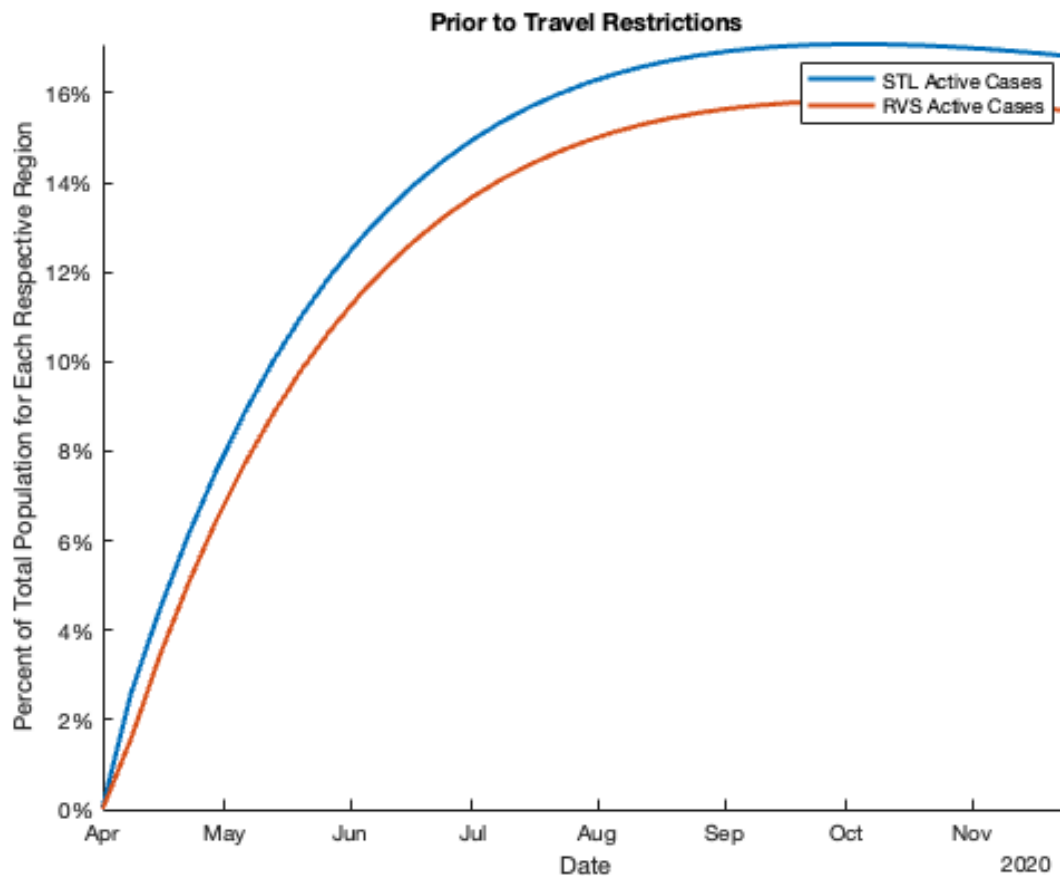
---

---

```

ytickformat('percentage');
exportgraphics(gca, 'no_restrictions.png');

```



first post-policy phase: 11/18/2020 ~ 6/29/2021

```

avgPostNewCases1 = mean(newCases1(36:67)) / (POP_STL - ...
    mean(cases_STL(36:67)) + mean(activeCases1(36:67)));
postNewDeaths1 = [];
for j = 36:67
    postNewDeaths1 = [postNewDeaths1, newDeaths1(j) / activeCases1(j)];
end
avgPostNewDeaths1 = mean(postNewDeaths1);

postActiveCases2 = [];
postNewCases2 = cases_RVS(36:67);
for j = 1:32
    if j == 1
        postActiveCases2 = [postActiveCases2, postNewCases2(j)];
    else
        postActiveCases2 = [postActiveCases2, postNewCases2(j) + ...
            round(postActiveCases2(j - 1) * 9 / 10)];
    end
end
avgPostNewCases2 = mean(postNewCases2) / (POP_RVS - ...

```

---

```

    mean(cumulativeCases2(36:67)) + mean(postActiveCases2));

x4 = x3;

A = [1 - avgPostNewCases1,      0.09,      0, 0;
      avgPostNewCases1,        0.9,        0, 0;
      0,                        0.01 - avgPostNewDeaths1, 1, 0;
      0,                        avgPostNewDeaths1, 0, 1];

B = [0.995,      0.01,      0, 0;
      0.005,      0.98,      0, 0;
      0,          0.01,      1, 0;
      0,          0,        0, 1];

C = [0.99,      0.005,      0, 0;
      0.01,      0.99,      0, 0;
      0,          0.005,      1, 0;
      0,          0,        0, 1];

D = [1 - avgPostNewCases2,      0.1,      0, 0;
      avgPostNewCases2,        0.8,      0, 0;
      0,                        0.095,      1, 0;
      0,                        0.005,      0, 1];

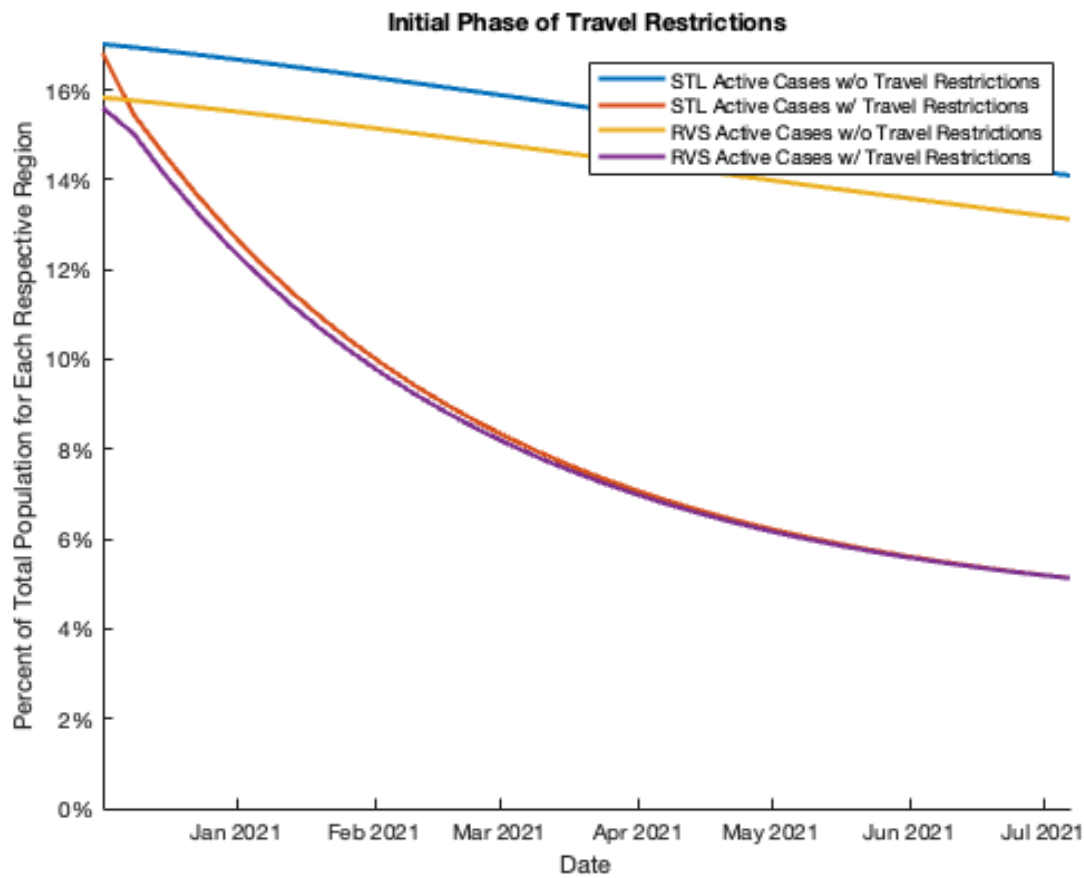
E = [A, B; C, D];

Y4 = x4;
for j = 37:67
    x4 = (E * x4) / 2;
    Y4 = [Y4, x4];
end

figure;
hold on;
plot(realDates(36:67), Y(2, 36:67) * 100, 'LineWidth', 2);
plot(realDates(36:67), Y4(2, :) * 100, 'LineWidth', 2);
plot(realDates(36:67), Y(6, 36:67) * 100, 'LineWidth', 2);
plot(realDates(36:67), Y4(6, :) * 100, 'LineWidth', 2);
hold off;
axis tight;
ylim([0 inf]);
title('Initial Phase of Travel Restrictions');
legend('STL Active Cases w/o Travel Restrictions', ['STL Active Cases ' ...
    'w/ Travel Restrictions'], ['RVS Active Cases w/o Travel ' ...
    'Restrictions'], 'RVS Active Cases w/ Travel Restrictions');
xlabel('Date');
ylabel('Percent of Total Population for Each Respective Region');
ytickformat('percentage');
exportgraphics(gca, 'initial_restrictions.png');

```

---



delta phase: 6/30/2021 ~ 10/26/2021

```

avgDeltaNewCases1 = mean(newCases1(68:84)) / (POP_STL - ...
    mean(cases_STL(68:84)) + mean(activeCases1(68:84)));
deltaNewDeaths1 = [];
for j = 68:84
    deltaNewDeaths1 = [deltaNewDeaths1, newDeaths1(j) / activeCases1(j)];
end
avgDeltaNewDeaths1 = mean(deltaNewDeaths1);

deltaActiveCases2 = [];
deltaNewCases2 = cases_RVS(68:84);
for j = 1:17
    if j == 1
        deltaActiveCases2 = [deltaActiveCases2, deltaNewCases2(j)];
    else
        deltaActiveCases2 = [deltaActiveCases2, deltaNewCases2(j) + ...
            round(deltaActiveCases2(j - 1) * 9 / 10)];
    end
end
avgDeltaNewCases2 = mean(deltaNewCases2) / (POP_RVS - ...
    mean(cumulativeCases2(68:84)) + mean(deltaActiveCases2));

```

---

```

x5 = x4;

A = [1 - avgDeltaNewCases1,    0.09,    0, 0;
      avgDeltaNewCases1,      0.9,      0, 0;
      0,                      0.01 - avgDeltaNewDeaths1, 1, 0;
      0,                      avgDeltaNewDeaths1,    0, 1];

D = [1 - avgDeltaNewCases2,    0.1,    0, 0;
      avgDeltaNewCases2,      0.8,      0, 0;
      0,                      0.095,    1, 0;
      0,                      0.005,    0, 1];

E = [A, B; C, D];

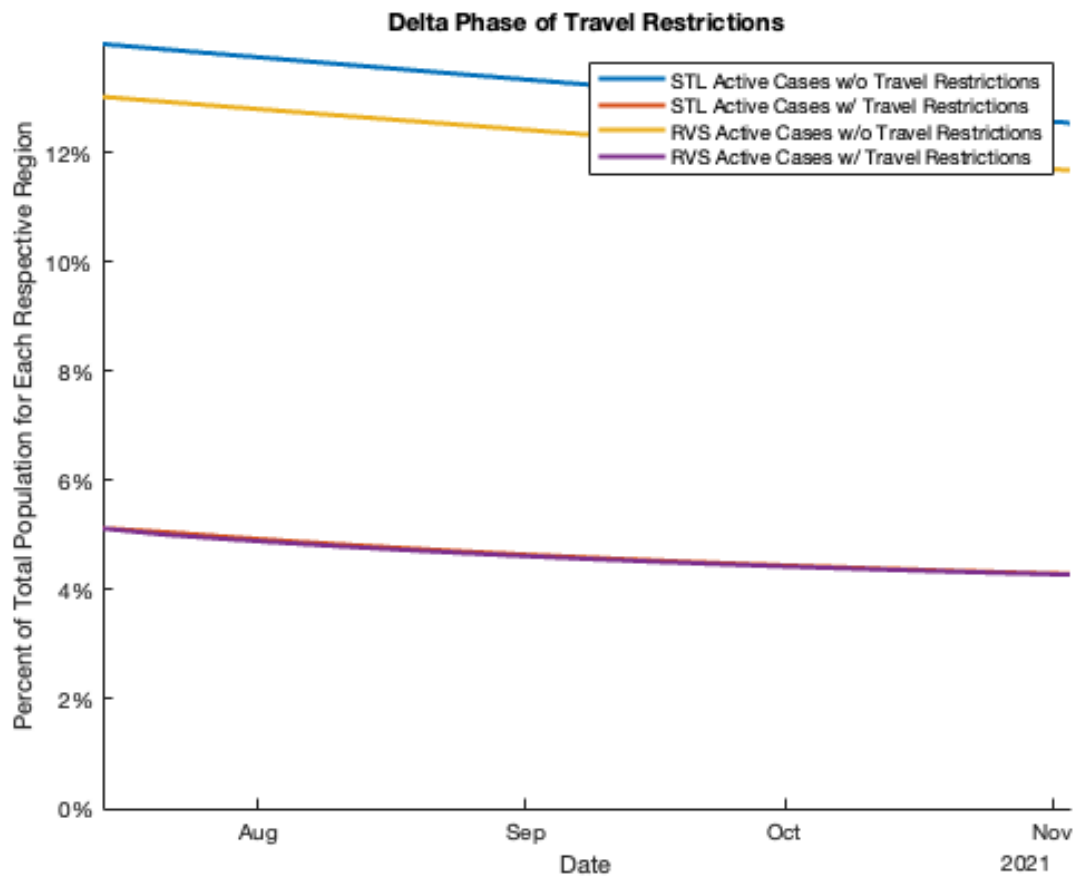
Y5 = x5;
for j = 69:84
    x5 = (E * x5) / 2;
    Y5 = [Y5, x5];
end

figure;
hold on;
plot(realDates(68:84), Y(2, 68:84) * 100, 'LineWidth', 2);
plot(realDates(68:84), Y5(2, :) * 100, 'LineWidth', 2);
plot(realDates(68:84), Y(6, 68:84) * 100, 'LineWidth', 2);
plot(realDates(68:84), Y5(6, :) * 100, 'LineWidth', 2);
hold off;
axis tight;
ylim([0 inf]);
title('Delta Phase of Travel Restrictions');
legend('STL Active Cases w/o Travel Restrictions', ['STL Active Cases ' ...
    'w/ Travel Restrictions'], ['RVS Active Cases w/o Travel ' ...
    'Restrictions'], 'RVS Active Cases w/ Travel Restrictions');
xlabel('Date');
ylabel('Percent of Total Population for Each Respective Region');
ytickformat('percentage');
exportgraphics(gca, 'delta_restrictions.png');

```

---





omicron phase: 10/27/2021 ~ 3/22/2022

```

avgOmicronNewCases1 = mean(newCases1(85:105)) / (POP_STL - ...
    mean(cases_STL(85:105)) + mean(activeCases1(85:105)));
omicronNewDeaths1 = [];
for j = 85:105
    omicronNewDeaths1 = [omicronNewDeaths1, newDeaths1(j) / ...
        activeCases1(j)];
end
avgOmicronNewDeaths1 = mean(omicronNewDeaths1);

omicronActiveCases2 = [];
omicronNewCases2 = cases_RVS(85:105);
for j = 1:21
    if j == 1
        omicronActiveCases2 = [omicronActiveCases2, omicronNewCases2(j)];
    else
        omicronActiveCases2 = [omicronActiveCases2, ...
            omicronNewCases2(j) + round(omicronActiveCases2(j - 1) ...
                * 9 / 10)];
    end
end
avgOmicronNewCases2 = mean(omicronNewCases2) / (POP_RVS - ...

```

---

```

    mean(cumulativeCases2(85:105)) + mean(omicronActiveCases2));

x6 = x5;

A = [1 - avgOmicronNewCases1,    0.09,                0, 0;
     avgOmicronNewCases1,        0.9,                0, 0;
     0,                          0.01 - avgOmicronNewDeaths1, 1, 0;
     0,                          avgOmicronNewDeaths1,   0, 1];

D = [1 - avgOmicronNewCases2,    0.1,                0, 0;
     avgOmicronNewCases2,        0.8,                0, 0;
     0,                          0.095,              1, 0;
     0,                          0.005,              0, 1];

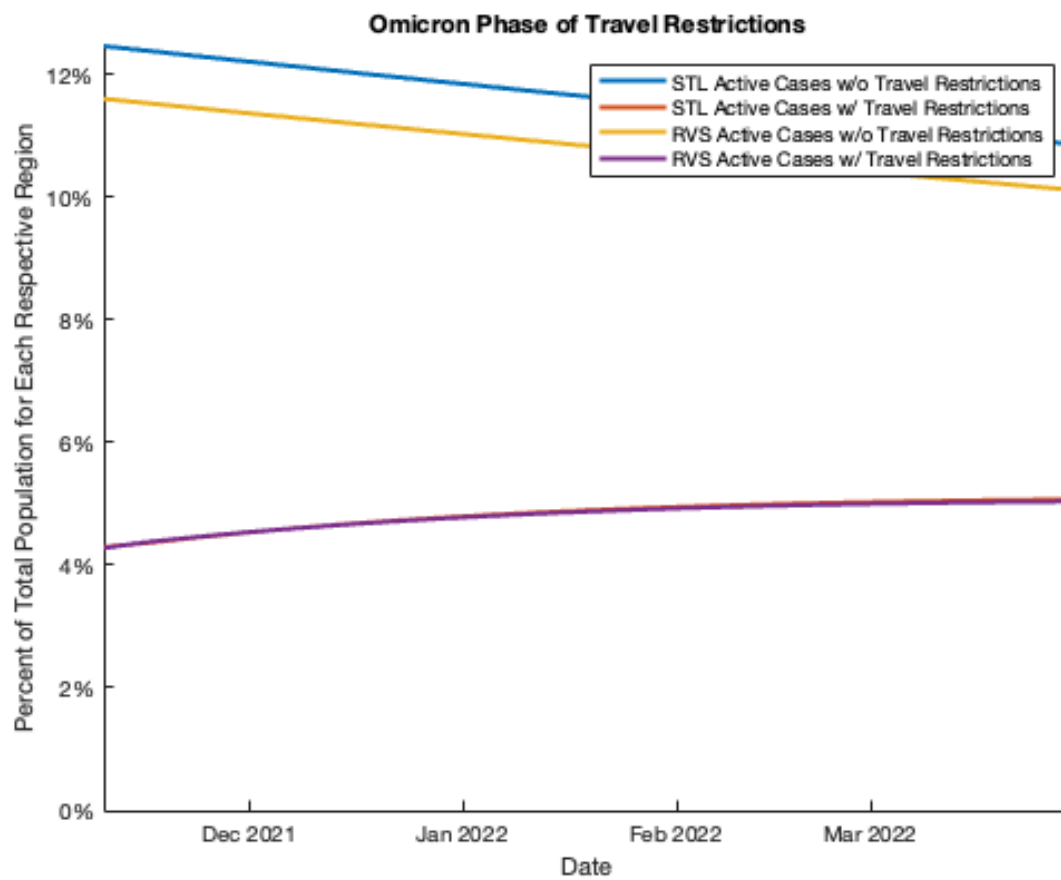
E = [A, B; C, D];

Y6 = x6;
for j = 86:105
    x6 = (E * x6) / 2;
    Y6 = [Y6, x6];
end

figure;
hold on;
plot(realDates(85:105), Y(2, 85:105) * 100, 'LineWidth', 2);
plot(realDates(85:105), Y6(2, :) * 100, 'LineWidth', 2);
plot(realDates(85:105), Y(6, 85:105) * 100, 'LineWidth', 2);
plot(realDates(85:105), Y6(6, :) * 100, 'LineWidth', 2);
hold off;
axis tight;
ylim([0 inf]);
title('Omicron Phase of Travel Restrictions');
legend('STL Active Cases w/o Travel Restrictions', ['STL Active Cases ' ...
    'w/ Travel Restrictions'], ['RVS Active Cases w/o Travel ' ...
    'Restrictions'], 'RVS Active Cases w/ Travel Restrictions');
xlabel('Date');
ylabel('Percent of Total Population for Each Respective Region');
ytickformat('percentage');
exportgraphics(gca, 'omicron_restrictions.png');

```

---



*Published with MATLAB® R2023a*