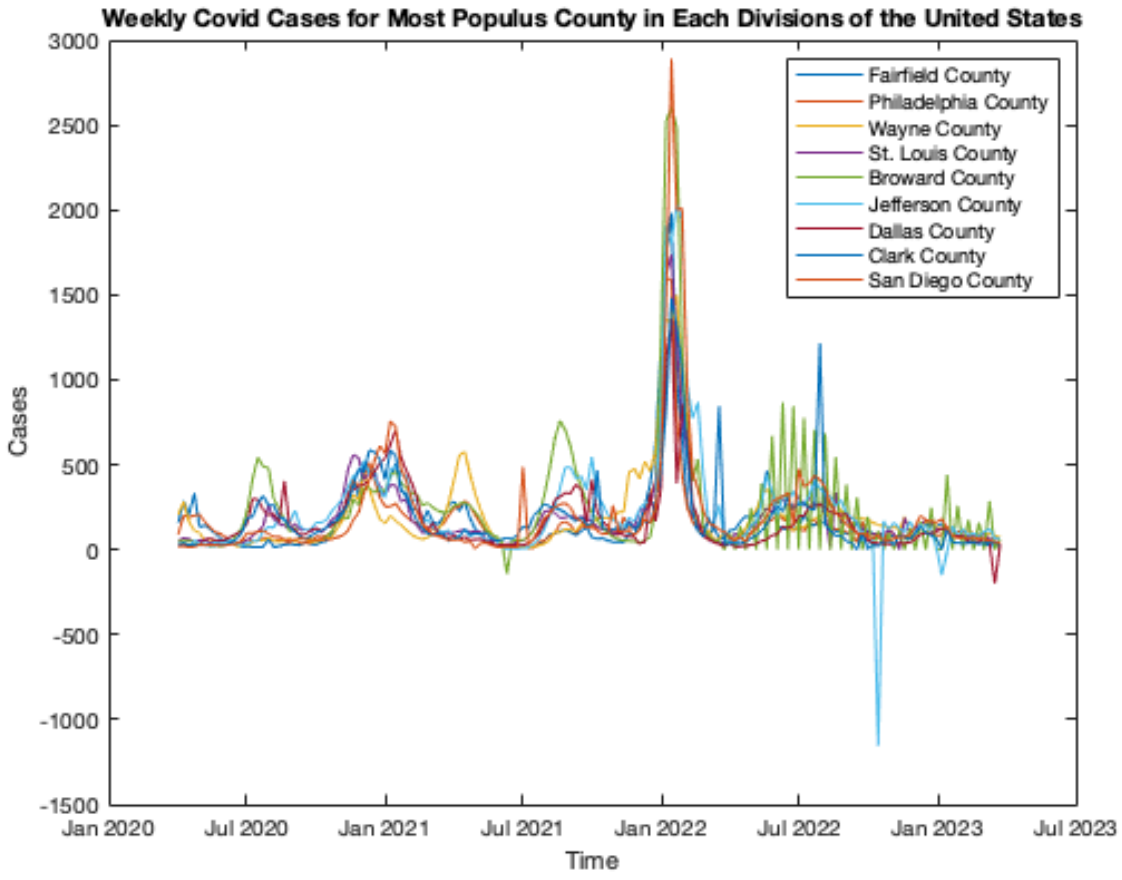
```
% Group: Eric Todd and Toby Sun
% 3.1 Lab Practice
close all
load('COVIDbyCounty.mat'); % Load data

% 3.1.1 Most populus country per division
numDivisions = max(CNTY_CENSUS.DIVISION);

% Initialize arrays
maxPopulationCovidCases = zeros(numDivisions,size(CNTY_COVID, 2));
maxPopulationCntys = cell(numDivisions, 1);

% Fill arrays by using "covidOFMaxPopOfDIvision" function that return the
% most populous city and corresponding case data.
for i = 1:numDivisions
    [maxPopulationCovidCases(i,:), maxPopulationCntys(i)] = ...
        covidOFMaxPopOfDIvision(CNTY_CENSUS, CNTY_COVID, i);
end

%Manually Checking
%sortCNTY = sortrows(CNTY_CENSUS, 6, "descend");

% Plot Data
plot(dates, maxPopulationCovidCases');
title("Weekly Covid Cases for Most Populus County in Each Divisions of the
 United States")
xlabel("Time")
ylabel("Cases")
legend(maxPopulationCntys);

% 3.1.2 Linearly Independent?

% Check every combination of vectors
check = 0;
for i = 1:numDivisions
    for j = 1:numDivisions
        if(i ~= j) % Of course a vector is linearly dependent with itself
            % Use formula cos(angle) = dot(a,b)/(norm(a)*norm(b)
            cases1 = maxPopulationCovidCases(i,:);
            cases2 = maxPopulationCovidCases(j,:)';
            angle = acos(cases1 * cases2) / (norm(cases1) * norm(cases2));
            if(angle == 0)
                % Print message if angle is 0;
                disp("Covid cases of most populus county in each division are
 not linearly independent!")
                check = check + 1;
                break;
            end
        end
    end
end
```

```
if(check == 0)
    disp("Covid cases of most populus county in each division are linearly
 independent!")
end

% 3.1.3 Normalize
d = normalize(maxPopulationCovidCases, 2, "norm");

% 3.1.4 St Louis City Case Data

idx = strcmp(CNTY_CENSUS.CTYNAME, "St. Louis city"); % logical index for "St.
 Louis city"
c = CNTY_COVID(idx,:); % Extract "St. Louis City" COVID cases data

% ri = c # (cTdi)di
r = repmat(c, numDivisions, 1) - d.*(d*c'); % Calculate the orthogonal vectors
rNorm = vecnorm(r,2,2); % Calculate norm of r

% The meaning of r, the formula of projection from c to d_i is
% (c^T*d_i)/((d_i)^T*d_i)* d_i. Because we normalized d_i, (d_i)^T*d_i
% equals to 1. So proj_(di) c = (c^T*d_i)*d_i. Therefore, orthogonal vector
% is c - proj_(di) c, which is r_i. Therefore, r_i measures the difference
% between St. Louis city cases vector and other d_i(representative vectors
% from each divison). r_i norm is the Euclidian distance between St. Louis
% city cases vector and d_i (from each division)

% Function Definitions:

% Find the most populus county in each division and return it's name and
 weekly covid cases.
function [cases, cntyName] = covidOFMaxPopOfDIvision(CNTY_CENSUS, CNTY_COVID,
 div)
    idx = CNTY_CENSUS.DIVISION == div; % Logical index for each division
    divisionRows = CNTY_CENSUS(idx,:); % Extract the rows from same division
    [~,idx] = max(divisionRows.POPESTIMATE2021); % Find county with the
 highest population
    row = divisionRows(idx,:); % Get the row number in each division
    idx = CNTY_CENSUS.fips == row.fips; % Distinguish counties that with same
 name
    cases = CNTY_COVID(idx,:); % Return corresponding cases data from
 CNTY_COVID
    cntyName = CNTY_CENSUS.CTYNAME(idx); % Return city names
end
```

Covid cases of most populus county in each division are linearly independent!



Published with MATLAB® R2023a