
```

load COVIDbyCounty.mat;

regions = ["Pacific", "Mountain", "West South Central", "West North
    Central", ...
    "East North Central", "East South Central", "Middle Atlantic", ...
    "South Atlantic", "New England"];

max_indicies = zeros(1,9);

figure()
hold on;

for region_idx = 1:length(regions)
    current_region = regions(region_idx);

    idx_region = (CNTY_CENSUS.DIVNAME == current_region);

    numRows = size(CNTY_CENSUS, 1);
    rowNumber = (1:numRows);

    CNTY_CENSUS.RowNumber = rowNumber';

    region_cntys = CNTY_CENSUS(idx_region, :);
    region_cntys_sorted =
    sortrows(region_cntys, "POPESTIMATE2021", 'descend');
    max = region_cntys_sorted(1, :);
    idx = max.RowNumber;
    max_indicies(region_idx) = idx;
    plot(CNTY_COVID(idx, :));
end

legend("Pacific", "Mountain", "West South Central", "West North Central", ...
    "East North Central", "East South Central", "Middle Atlantic", ...
    "South Atlantic", "New England")
hold off;

%finding angles between vectors
linear_idp_flag = true;

for i = 1:length(max_indicies)
    current_vector = CNTY_COVID(max_indicies(i), :);
    for j = 1:length(max_indicies)
        if i ~= j
            comparison_vector = CNTY_COVID(max_indicies(j), :);

            dot_product = dot(current_vector, comparison_vector);
            norm_current = norm(current_vector);
            norm_comparison = norm(comparison_vector);

            angle = dot_product / (norm_current * norm_comparison);
            if angle == 0

```

```

        disp("No angle in between vectors");
        linear_idp_flag = false;
    end
end
end
end

if linear_idp_flag
    disp("The vectors are linearly independent");
else
    disp("the angles are not linearly independent");
end

d = zeros(9, 156);
for k = 1:9
    curr_max = CNTY_COVID(max_indicies(k), :);
    d(k,:) = curr_max / norm(curr_max);
end

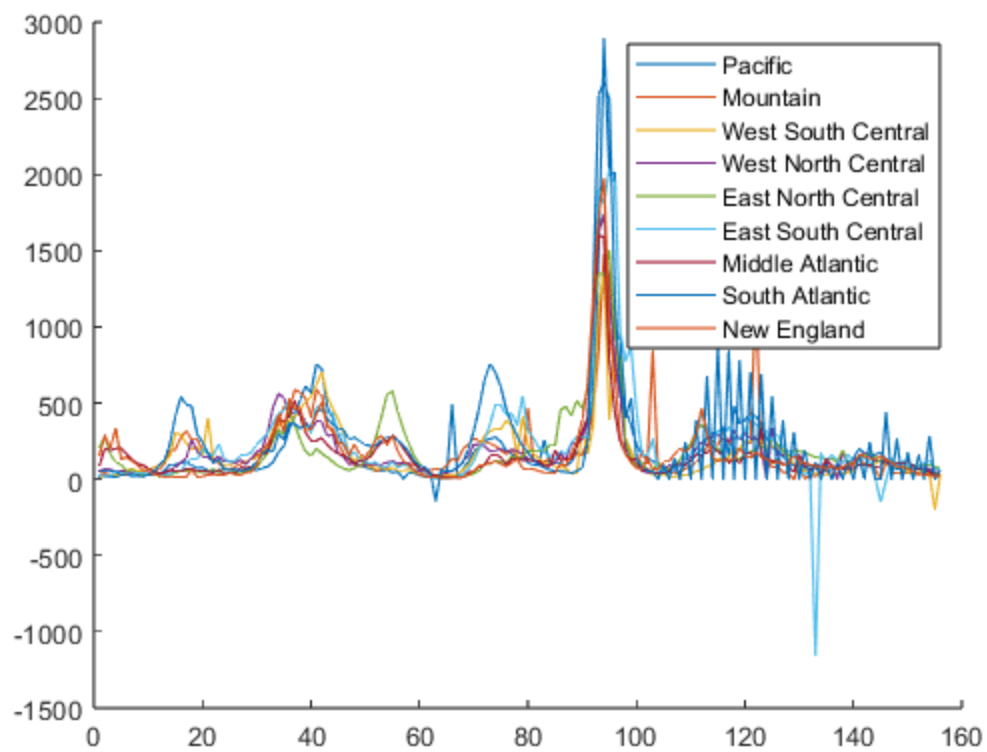
stl_idx = (CNTY_CENSUS.CTYNAME == "St. Louis city");
stl_vector = CNTY_COVID(stl_idx, :);
r = zeros(9, 156);
r_norms = zeros(1, 9);

for m = 1:9
    dot_stl = dot(stl_vector, d(m, :));
    r_i = stl_vector - dot_stl * d(m,:);
    r(m,:) = r_i;
    r_norms(m) = norm(r_i);
end

% r_i gives you information about the difference in case numbers for each
% day between the most populous counties in each region and st. Louis City for
% each day, the norm gives you the total. They might indicate similarity
% between the case rate in St Louis city and the given counties.

```

The vectors are linearly independent



Published with MATLAB® R2023a