

---

```
% Anthony Galczak - Tristin Glunt
% CS375 - HW8
clc
close all

% Q1b
A = [1 -3 3; 2 -2 2; 2 0 0];
a_eig = eig(A);
fprintf("Q1b\nEigenvalues of A\n");
fprintf("%g\n",a_eig);

% Q2b
[eval,vec] = power_method(A, [1;1;1], 10^(-3));
fprintf("\nQ2b\nPower method with [1;1;1]\n");
fprintf("Eigenvalue: %f\nEigenvector:\n", eval);
fprintf("%g\n", vec');

[eval,vec] = power_method(A, [1;-1;1], 10^(-3));
fprintf("\nPower method with [1;-1;1]\n");
fprintf("Eigenvalue: %f\nEigenvector:\n", eval);
fprintf("%g\n", vec);

%Q2d
lam=3;
n=20;
B = spdiags(ones(n,1)*[-1 2-lam -1],[-1, 0, 1], n,n);
A = spdiags(ones(n,1)*[-1 2 -1],[-1, 0, 1], n,n);

[eval_A,vec_A] = power_method(A, ones(n,1), 10^(-15));
[eval_B,vec_B] = power_method(B, ones(n,1), 10^(-15));

fprintf("Q2d\n");
fprintf("Eigenvalue of A: %d\n", eval_A);
fprintf("Eigenvalue of B: %d\n", eval_B);

figure
plot(1:20, vec_A);
hold on
plot(1:20, vec_B);

title("2d.) Eigenvectors of A and B");
xlabel("x-axis");
ylabel("y-axis");
legend(["A","B"]);
hold off

%Q3b
data = csvread('polls.csv');
days = data(:,1);
clinton = data(:,2);
sanders = data(:,3);
diff = clinton - sanders;
```

---

```

figure
plot(days, diff);
hold on

% Flipping these as it's better to look at it as a time-series.
set(gca, 'xdir', 'reverse');

title("3b.) Clinton net lead over Sanders");
xlabel("Days from Jan. 3");
ylabel("Net lead (+Clinton)");
hold off

%Q3d
n_s = [1,3,5,11,15];

for n = n_s

    figure
    % Plot the real values first
    scatter(days, diff);

    hold on
    % The coefficients needed to be flipped.
    coeffs = poll_projection(days,diff,n);
    y_vals = polyval(flipud(coeffs), days);
    plot(days, y_vals);

    title(['Q3d.) Poll projection (estimate of +Clinton) for poly n='
    num2str(n)]);

    xlabel("Days until Jan. 3, 2016");
    ylabel("+Clinton over Sanders");

    hold off
end

%Q2a
function [eval, evec] = power_method(A, x, tol)

    accuracy = 100;
    eval = 0;

    while accuracy > tol
        y = A * x;
        x_prev = x;

        x = y / norm(y, inf);

        % Making sure signs are the same for next iter
        if sign(x_prev(1)) ~= sign(x(1))
            x = x * -1;
        end
    end

```

---

---

```

        eval = phi(y) / phi(x);

        evec = x;
        accuracy = norm(x - x_prev, 2);
    end

end

% Function that returns the first non-zero element of a vector.
function element = phi(x)

    for i = 1 : size(x,2)
        if x(i) ~= 0
            element = x(i);
            break
        end
    end
end

%Q3c
function coeffs = poll_projection(xvals, yvals, n)

    % A is vandermonde matrix of xvals,
    % n = 2
    % Matrix is columns: xvals^0, xvals^1, xvals^2
    A = fliplr(vander(xvals'));
    A = A(:,1:(n+1));

    [Q,R] = qr(A,0);

    % Getting "[c1;c2]"
    C = transpose(Q) * yvals;
    coeffs = R \ C;

end

Q1b
Eigenvalues of A
1
1.11022e-15
-2

Q2b
Power method with [1;1;1]
Eigenvalue: 1.000000
Eigenvector:
0.5
1
1

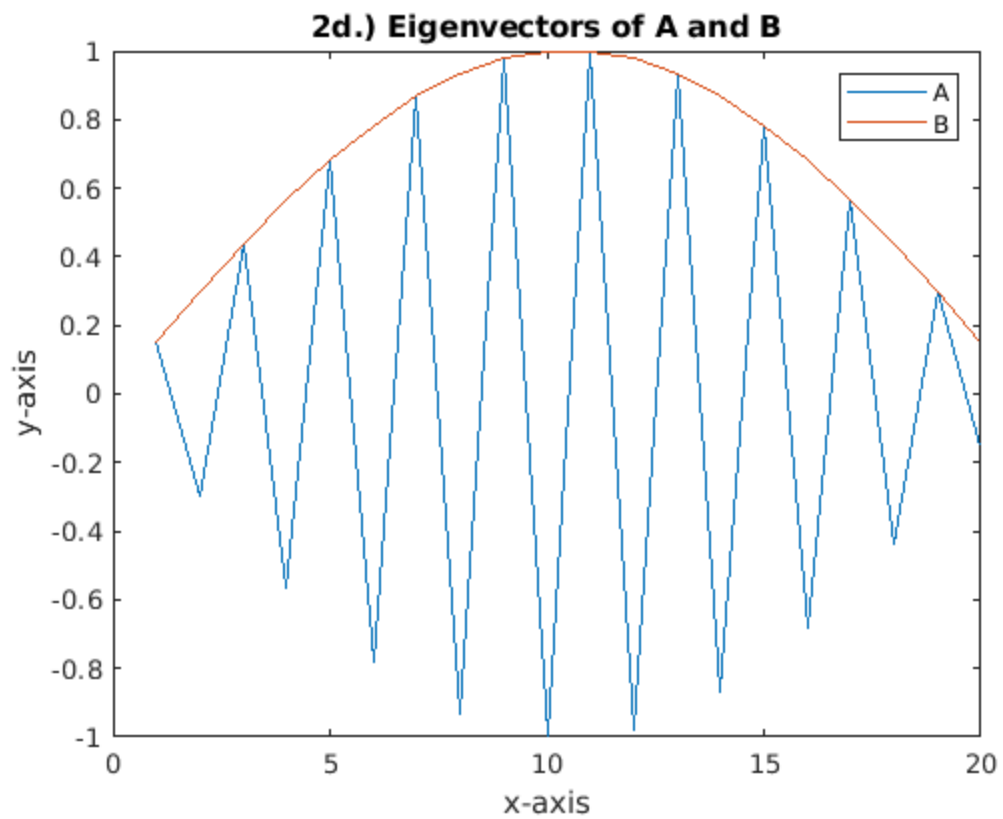
Power method with [1;-1;1]
Eigenvalue: -2.000000
Eigenvector:

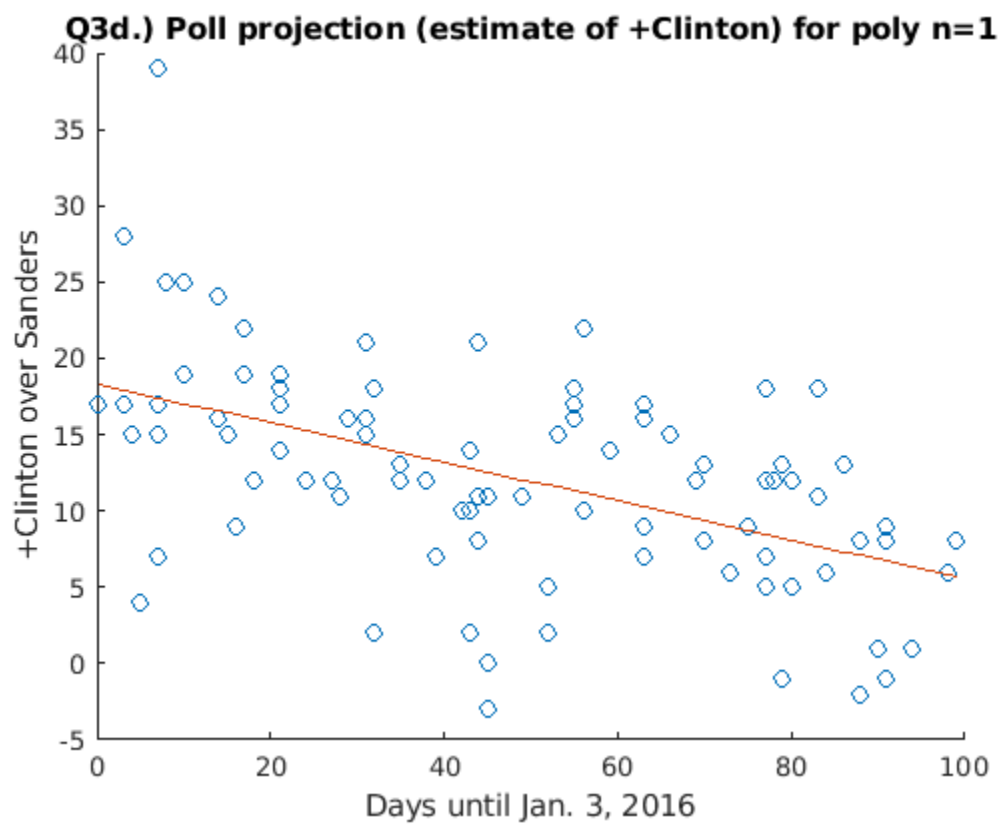
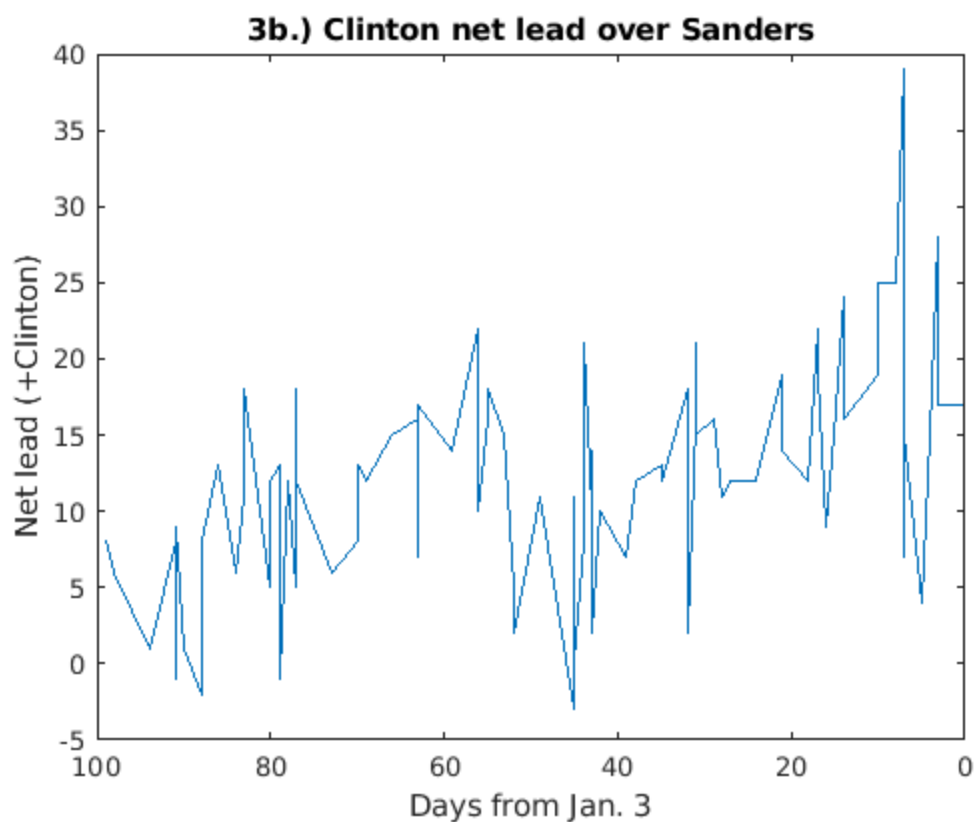
```

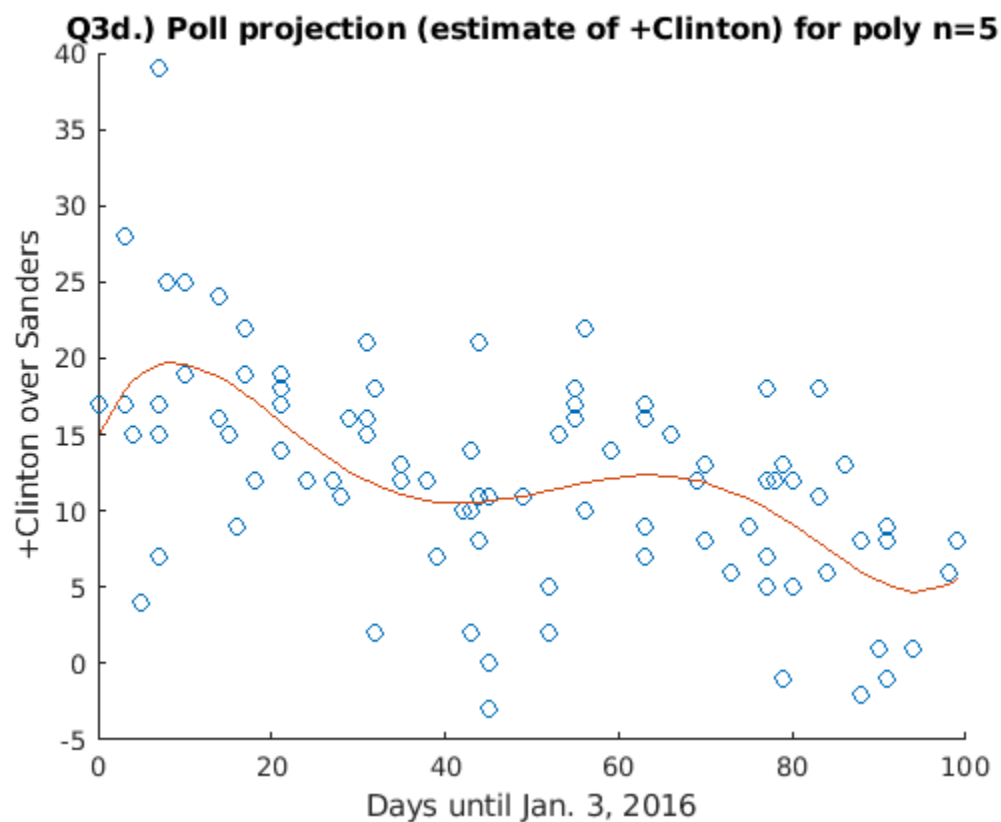
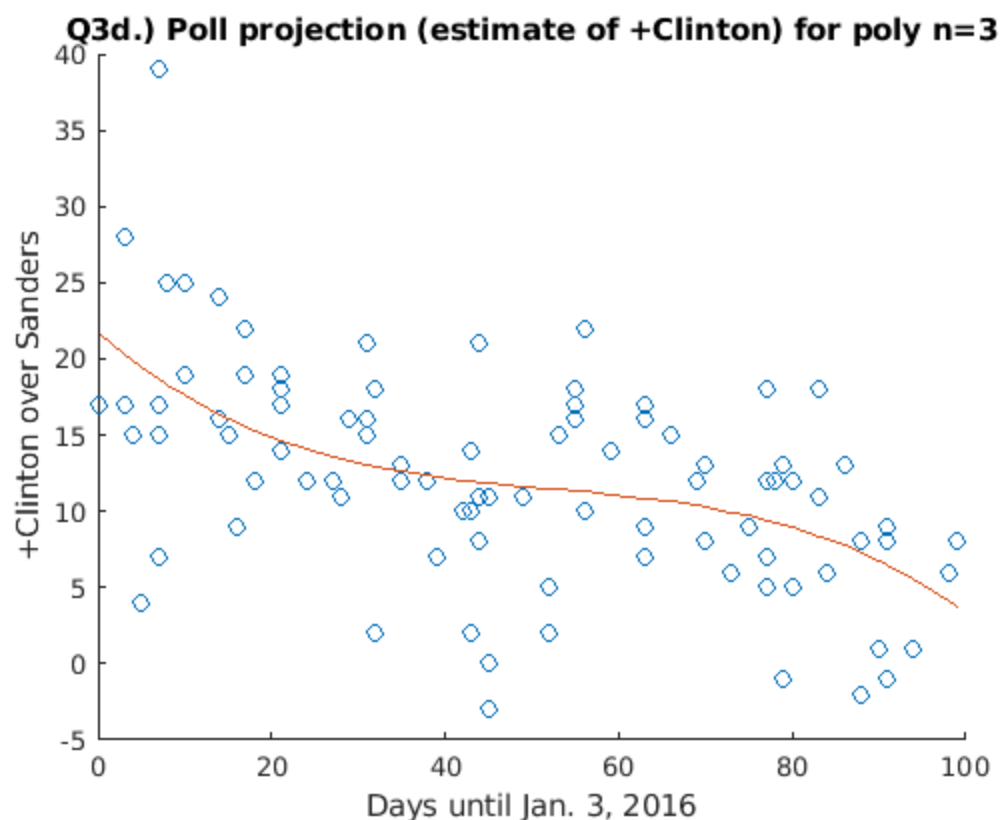
---

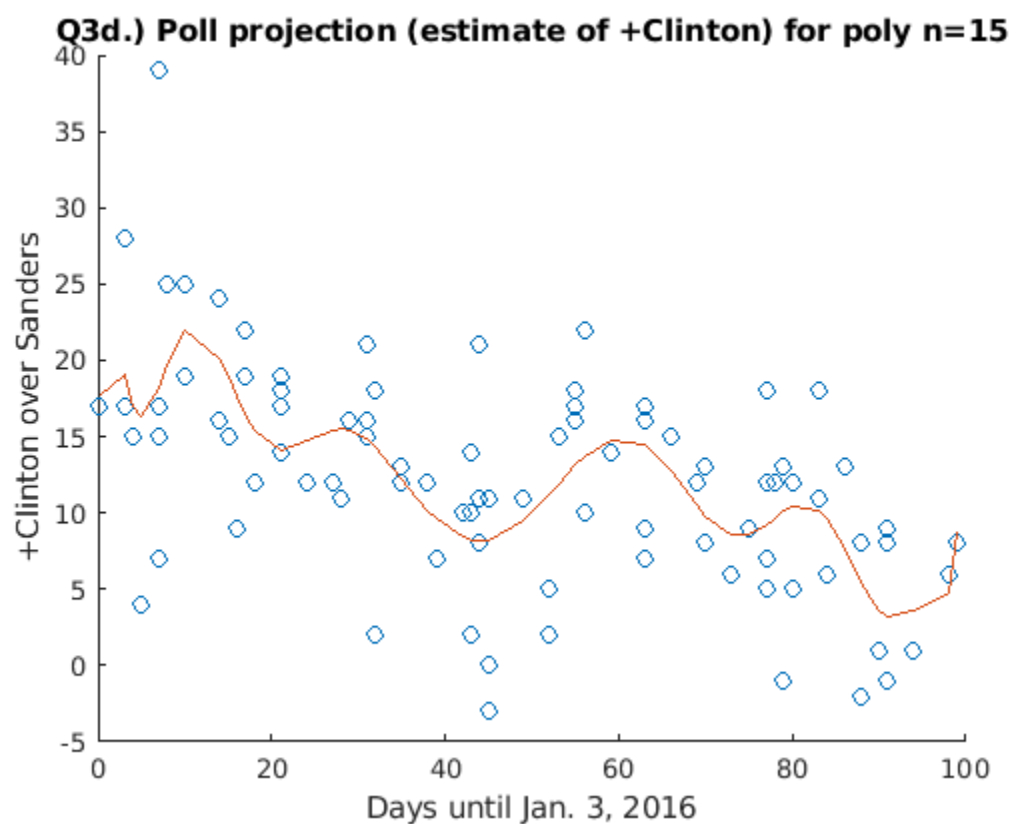
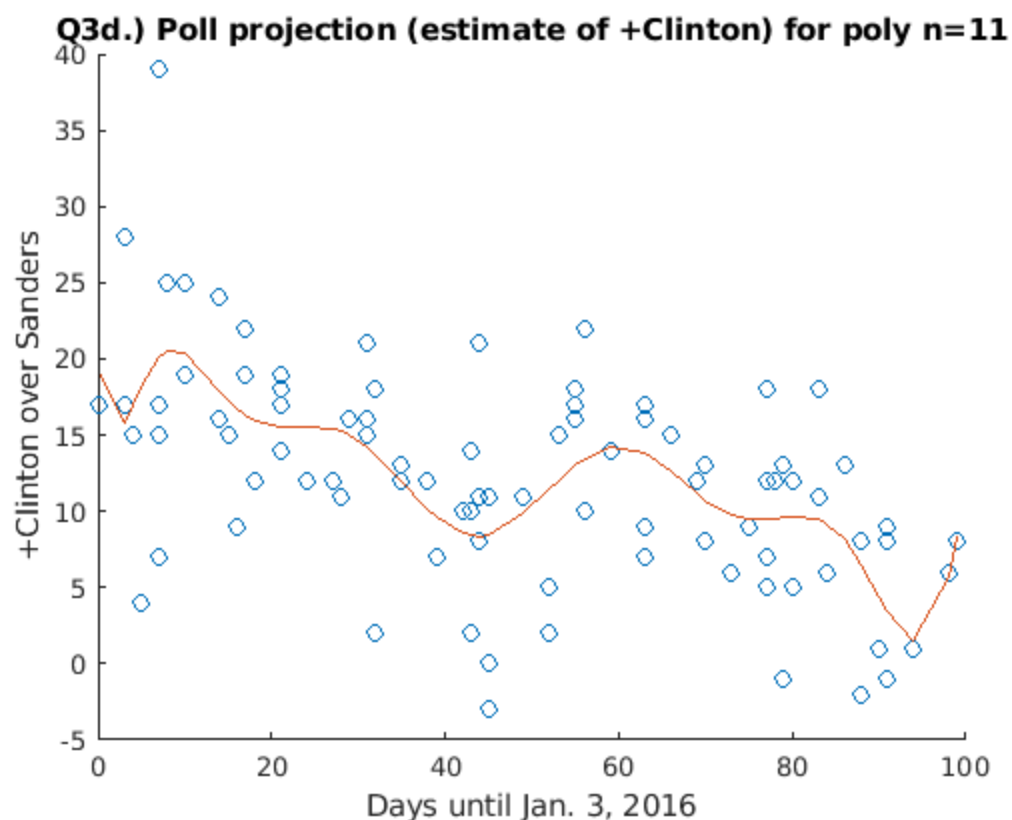
---

0.999725  
-0.000183072  
-1  
Q2d  
Eigenvalue of A: 3.977662e+00  
Eigenvalue of B: -2.977662e+00  
Warning: Matrix is close to singular or badly scaled. Results may be  
inaccurate. RCOND = 1.126543e-21.









---

*Published with MATLAB® R2018a*