Table of Contents

lab-9: Regression

```
% Regression establishes a function f that describes the trend
 (relationship)
% among a set of input x and output y data points, i.e., y = f(x).

% we consider the least-squares (LS) approach to minimizing
% the sum of squared errors, where the error is described by
% the vertical distance to the curve from the data points.

% we evaluate:
% 1. Linear least square regression using straight line fit
% 2. Polynomial curve fit
```

Polynomial regression degrees

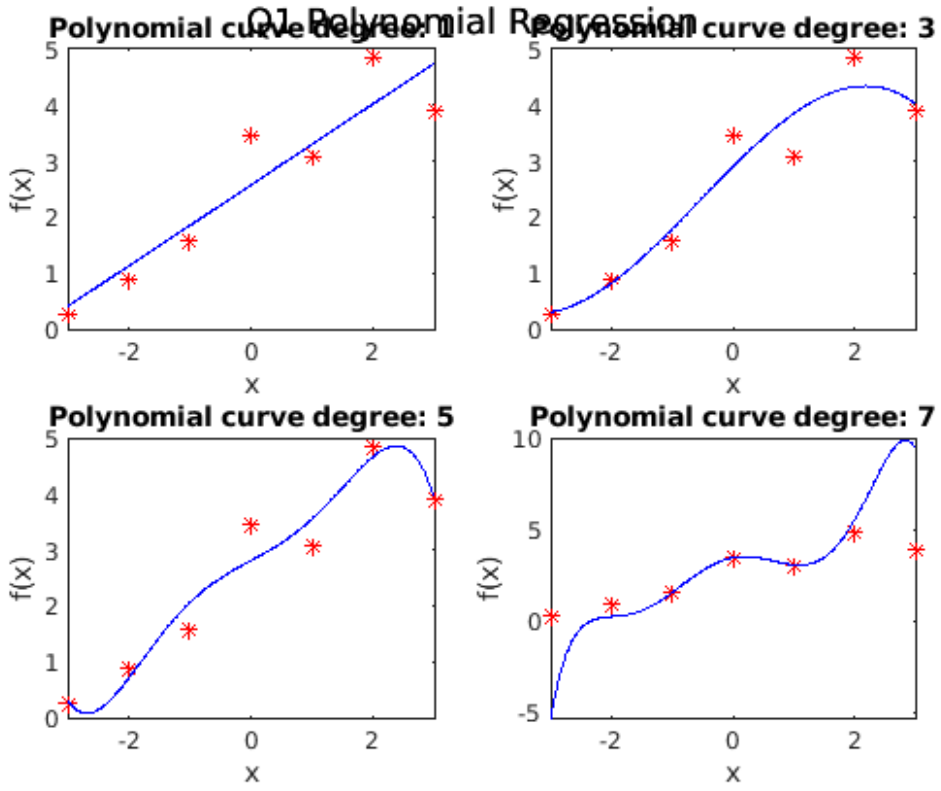
```
polyFit = [1 3 5 7];
len = length(polyFit);
```

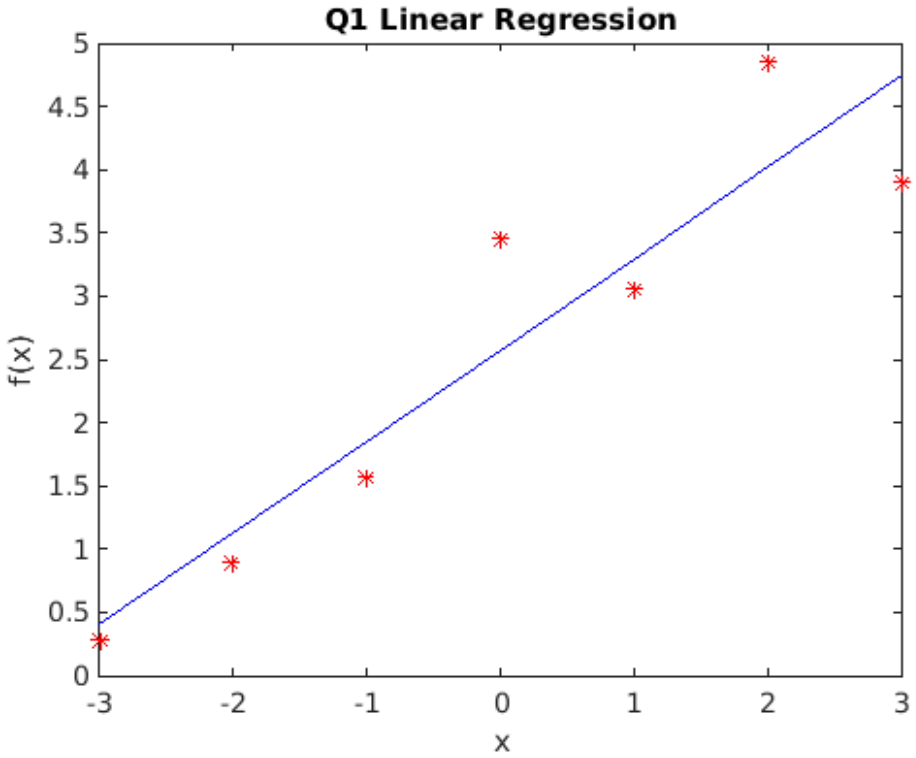
Question 1

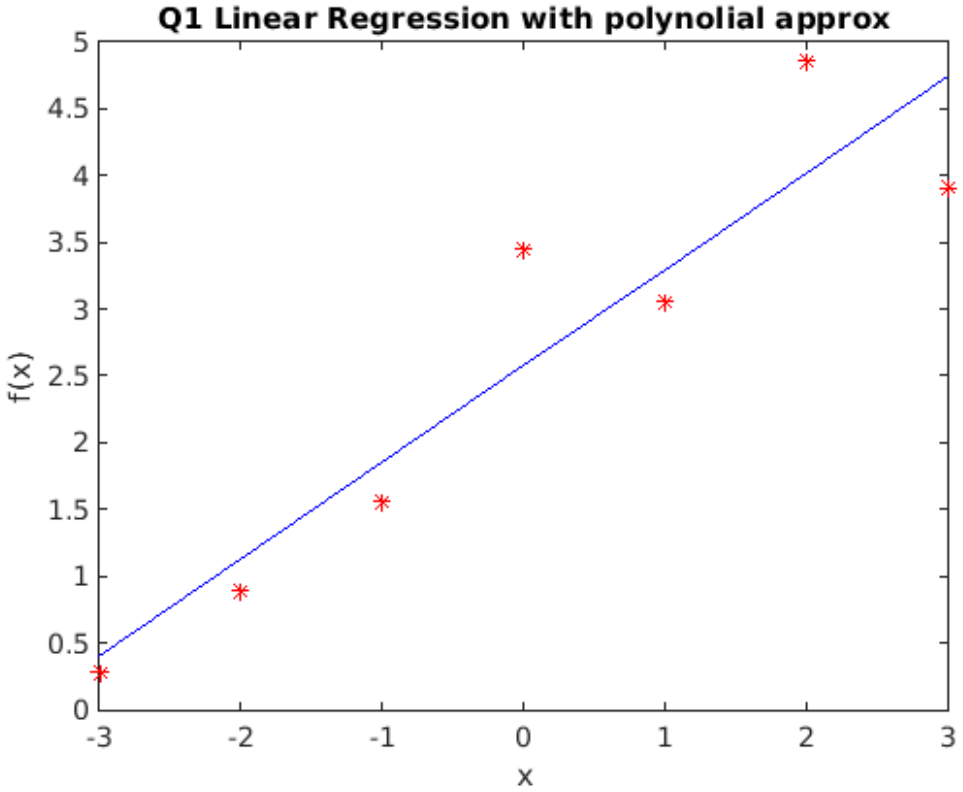
Initialisation

```
x1 = [-3 -2 -1 0 1 2 3];
y1 = [0.2774 0.8958 1.5651 3.4565 3.0601 4.8568 3.8982];

figure(1);

for i = 1 : len
    P = polyreg(x1, y1, polyFit(i));
    xValues = linspace(-3, 3, 500);
    yValues = polyval(P, xValues);
    subplot(2, 2, i)
    plot(x1, y1, 'r*', xValues, yValues, 'b-');
    title("Polynomial curve degree: " + polyFit(i));
    xlabel('x');
    ylabel('f(x)');
end

sgtitle('Q1 Polynomial Regression');
```

```
figure(2);

[b, y] = linreg(x1,y1);
plot(x1, y1, 'r*', x1, y, 'b-');
title("Q1 Linear Regression");
xlabel('x');
ylabel('f(x)');
disp ("Q1 theta = ");
disp(b);

figure(3);

[b, y] = linregpolyapp(x1,y1);
plot(x1, y1, 'r*', x1, y, 'b-');
title("Q1 Linear Regression with polynolial approx");
xlabel('x');
ylabel('f(x)');
disp ("Q1 theta = ");
disp(b);
```




Q1 Linear Regression
f(x)
x
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-3
-2
-1
0
1
2
3


Q1 Linear Regression with polynolial approx
f(x)
x
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-3
-2
-1
0
1
2
3

Question 2

Initialisation

```
x2 = [0.8 1.4 2.7 3.8 4.8 4.9];
y2 = [0.69 1.00 2.00 2.39 2.34 2.83];

figure(4);

for i = 1:len
    P = polyreg(x2, y2, polyFit(i));
    xValues = linspace(0, 6, 500);
    yValues = polyval(P, xValues);
    subplot(2, 2, i)
    plot(x2, y2, 'r*', xValues, yValues, 'b-');
    title("Polynomial curve degree: " + polyFit(i));
    xlabel('x');
    ylabel('f(x)');
end

sgtitle('Q2 Polynomial Regression');

figure(5);

[b, y] = linreg(x2,y2);
plot(x2, y2, 'r*', x2, y, 'b-');
title("Q2 Linear Regression");
xlabel('x');
ylabel('f(x)');
snapnow;
disp ("Q2 theta = ");
disp(b);

figure(6);

[b, y] = linregpolyapp(x2,y2);
plot(x2, y2, 'r*', x2, y, 'b-');
title("Q2 Linear Regression with polynomial approx");
xlabel('x');
ylabel('f(x)');
snapnow;
disp ("Q2 theta = ");
disp(b);
```

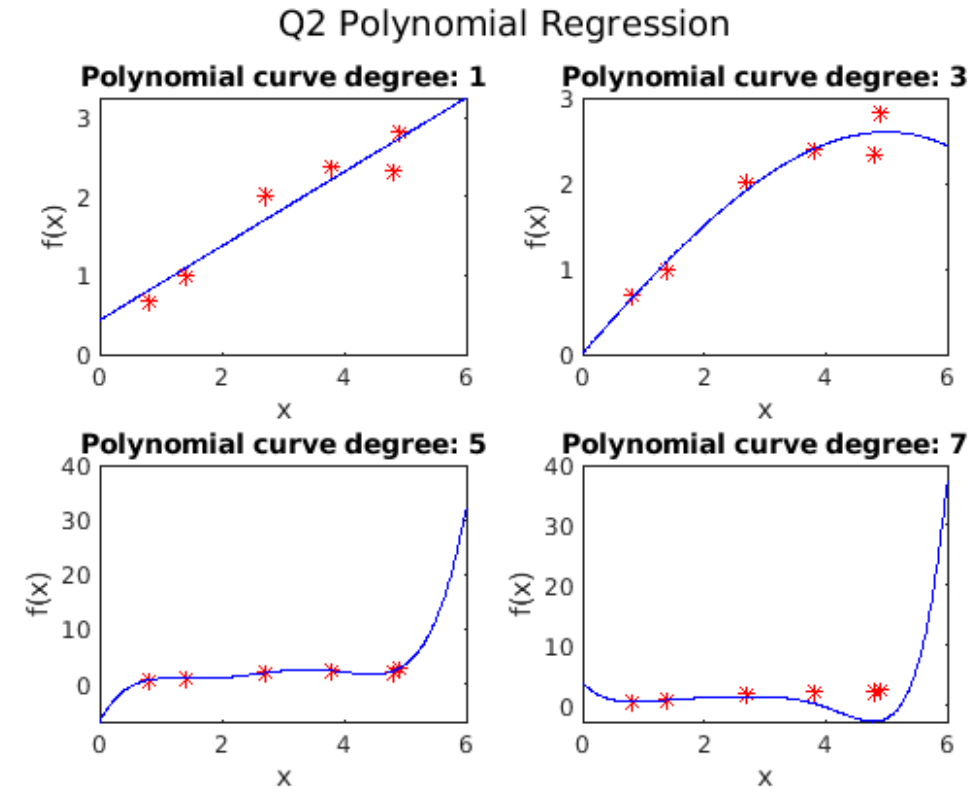
Functions

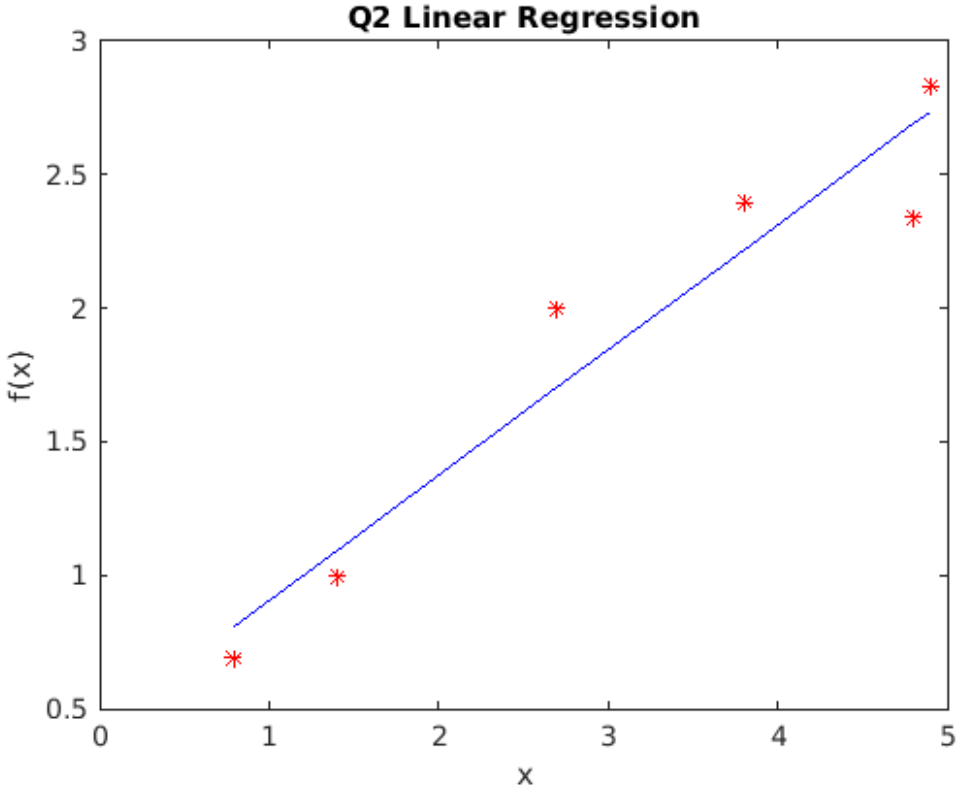
Polynomial Regression

```
function ypred = polyreg(x,y, feature)
    n = length(x);
    A = [x.' ones(n,1)];
    Y = y.';
```

```
    for i = 2:feature
        A = [A(:, 1).*(x.') A];
    end

    ypred = (inv((A.')*A)*(A.'))*Y;
end

% Linear Regression

function [b, ypred] = linreg(x,y)
    y = transpose(y);
    sumX = 0;
    sumX2 = 0;
    sumY = 0;
    sumXY = 0;

    for i = 1:length(x)
        sumX = sumX + x(i);
        sumX2 = sumX2 + x(i)*x(i);
        sumY = sumY + y(i);
        sumXY = sumXY + x(i)*y(i);
    end

    b = (length(x)*sumXY - sumX*sumY)/(length(x)*sumX2 - sumX*sumX);
    a = (sumY - b*sumX)/length(x);

    ypred = a + b*x;
    b = [a; b];
end

% Linear Regression using Polynomical approx of first degree

function [b, ypred] = linregpolyapp(x,y)
    X = [ones(length(x),1) transpose(x)];
    y = transpose(y);
    b = X\y;
    ypred = (X*b).';
end
```

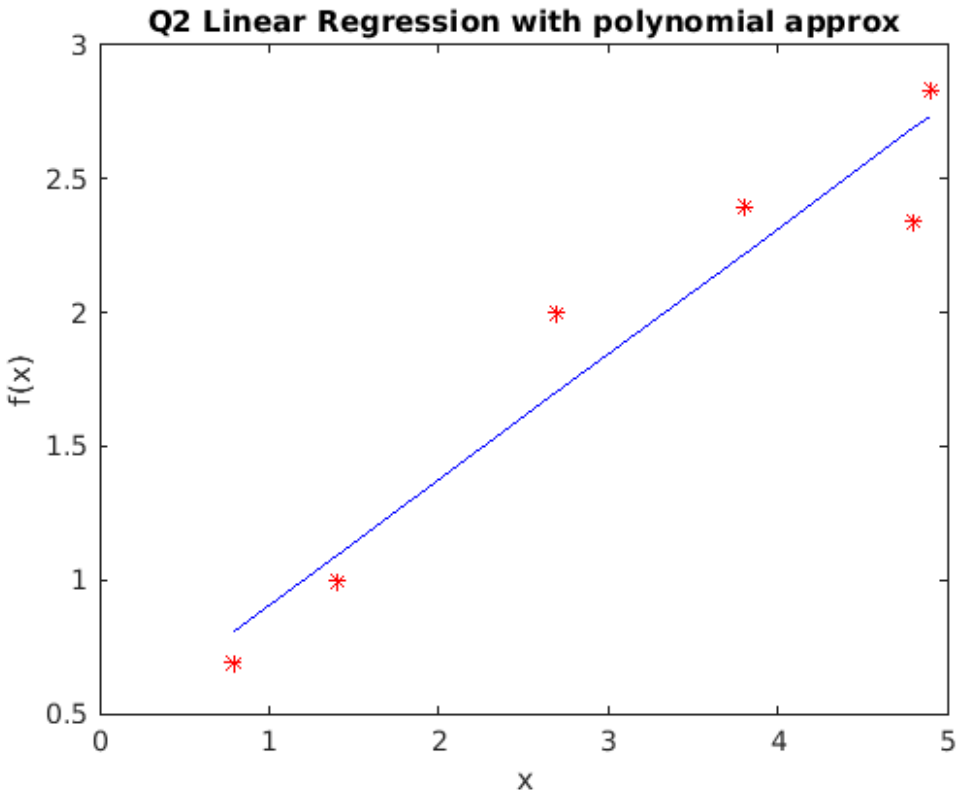
```
Warning: Matrix is close to singular or badly scaled. Results may be
inaccurate. RCOND =  1.215045e-20.
Q1 theta =
    2.5728
    0.7243

Q1 theta =
    2.5728
    0.7243
```


Q2 Polynomial Regression
Polynomial curve degree: 1
Polynomial curve degree: 3
Polynomial curve degree: 5
Polynomial curve degree: 7
f(x)
x


Q2 Linear Regression
f(x)
x

Q2 theta =

```
    0.4327
    0.4703
```




```
Q2 theta =
    0.4327
    0.4703
```

Published with MATLAB® R2021a