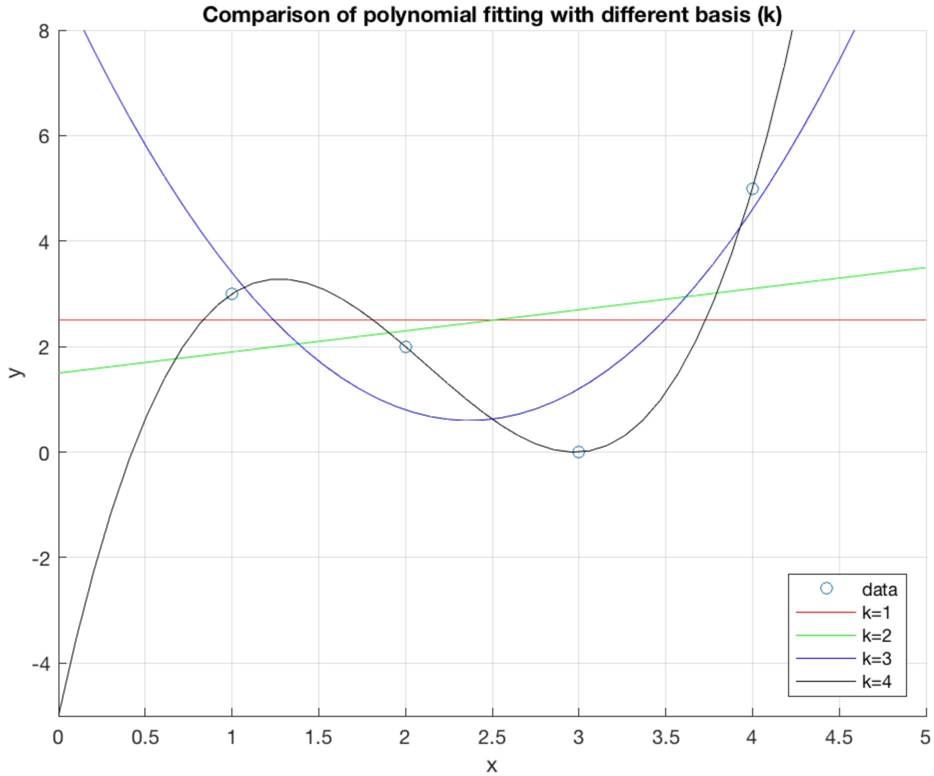
```
%GI07 Homework 1 Question 2 Part 1
%Klaudia Ludwisiak

%=====LINEAR=REGRESSION========
%1st Setup the problem
close all;
clear all;

x = [1; 2; 3; 4];
y = [3; 2; 0; 5];
dummy=linspace(0,5,50);

%2nd fit polynomials to data in order to find best estimate of w
%define a function to MYpolynom.m fit polynomials easily. This takes
 as
%input vector x and the degree of the polynomial k and outputs the
 f(x)
%equivalent to the feature map.
fx1 = MYpolynom(x,1);
fx2 = MYpolynom(x,2);
fx3 = MYpolynom(x,3);
fx4 = MYpolynom(x,4);

%find paremeter w
w1 = fx1\y;
w2 = fx2\y;
w3 = fx3\y;
w4 = fx4\y;

%need larger resolution for plotting hence increase size of x
dummy=linspace(0,5,50)';
xfit = dummy;
x1fit = MYpolynom(xfit,1);
x2fit = MYpolynom(xfit,2);
x3fit = MYpolynom(xfit,3);
x4fit = MYpolynom(xfit,4);

%=========PART==A=========
figure;
hold on
title('Comparison of polynomial fitting with different basis (k)');
axis([0,5,-5,8]);
xlabel('x');
ylabel('y');
grid on;
plot(x,y,'o');
plot(dummy,x1fit*w1,'r'); %plot augmented resolution but keep w
 parameter from fitting to original set of four points
plot(dummy,x2fit*w2,'g');
plot(dummy,x3fit*w3,'b');
plot(dummy,x4fit*w4,'k');
legend('data','k=1','k=2','k=3','k=4','Location','southeast');
```

```
hold off

%=========PART==B=========
%make use of matlab function polyfit to obtain coefficients of the
%polynomial of interest.
    %function specification:
    %P = polyfit(X,Y,N) finds the coefficients of a polynomial P(X) of
    %degree N that fits the data Y best in a least-squares sense.

P1=polyfit(dummy,x1fit*w1,1);
P2=polyfit(dummy,x2fit*w2,2);
P3=polyfit(dummy,x3fit*w3,3);
P4=polyfit(dummy,x4fit*w4,4);
%P4 checks out to solution provided,
    %note: a small term also exists which is aproximate to zero

fprintf('Polynomial order 1: P1=%.2f\n',P1(1,2))
fprintf('Polynomial order 2: P2=%.2f + %.2f x\n ',P2(1,3),P2(1,2))
fprintf('Polynomial order 3: P3=%.2f + %.2f x + %.2f
 x^2\n',P3(1,4),P3(1,3),P3(1,2))
fprintf('Polynomial order 4: P4=%.2f + %.2f x + %.2f x^2+ %.2f x^3\n
\n',P4(1,5),P4(1,4),P4(1,3),P4(1,2))

%=========PART==C==========
%estimating mean squared error using the property: MSE=SSE/m where m
 is the
%amount of rows in the system
MSE1=(fx1*w1-y)'*(fx1*w1-y)/size(y,1);
MSE2=(fx2*w2-y)'*(fx2*w2-y)/size(y,1);
MSE3=(fx3*w3-y)'*(fx3*w3-y)/size(y,1);
MSE4=(fx4*w4-y)'*(fx4*w4-y)/size(y,1);

fprintf('MSE for k=1 is %.3f\n',MSE1)
fprintf('MSE for k=2 is %.3f\n',MSE2)
fprintf('MSE for k=3 is %.3f\n',MSE3)
fprintf('MSE for k=4 is %f\n',MSE4)

%lowest for k=4. Fitting polynomials of higher degree will increase
 the
%MSE, this is to be expected as polynomials of higher order than the
 amount
%of original training points are likely an overfit.
%having tested this for k=6 and k=9 and indeed MSE is minimum at k=4
 and
%increases thereafter, however it error remains low.
```

```
Polynomial order 1: P1=2.50
Polynomial order 2: P2=1.50 + 0.40 x
 Polynomial order 3: P3=9.00 + -7.10 x + 1.50 x^2
Polynomial order 4: P4=-5.00 + 15.17 x + -8.50 x^2+ 1.33 x^3

MSE for k=1 is 3.250
MSE for k=2 is 3.050
MSE for k=3 is 0.800
```

```
MSE for k=4 is 0.000000
```



Published with MATLAB® R2016b