

ECE 204 Simulation Assignment 3
Phoebe Luo
University of Waterloo

Output Results

Do the following for “test1.txt” and “test2.txt” and present the results in the report:

- Draw the raw data
- Write the estimated formula on the Figure you drew (examples shown in the Note at the end of the page)
- Draw the estimated function on the Figure.
- Calculate R^2 and present it on the Figure you drew.
- Provide warning for division by 0 and invalid natural logarithm
- Show the command window results during function selection (1, 2, or 3)

- The code is:

```
clc;

% Obtaining Data
A = readmatrix("test1.txt");
x = A(:, 1);
x = x';
y = A(:, 2);
y = y';
n = size(x,2);

% Getting Command
disp("1. Polynomial");
disp("2. Exponential");
disp("3. Saturation");
prompt = "Select the function to fit your data: ";
command = input(prompt);

% 1. Polynomial
if command == 1
    prompt = "Degree of polynomial: ";
    command_p = input(prompt);
    if command_p == 1
        e_1 = x; % matrix of x* = x
        e_2 = y; % matrix of y* = y
        e_3 = x.^2; % matrix of x^2
        e_4 = x.*y; % matrix of xy
        a1 = (n*sum(e_4) - sum(e_1)*sum(e_2))/(n*sum(e_3)-(sum(e_1))^2);
        a0 = sum(e_2)/n - a1*sum(e_1)/n;
        % calculate R^2
        St = 0;
        Sr = 0;
        for c = 1:n
            % for St
            cal_1 = (y(1,c) - (sum(y)/n))^2; % (yi-avg(y))^2
            St = St + cal_1;
            % for Sr
            cal_2 = (y(1,c) - (a0 + a1*x(1,c)))^2; % (yi-y)^2
```

```

Sr = Sr + cal_2;
end
error = (St - Sr)/St;
elseif command_p == 2
e_1 = x.^2; % matrix of x^2
e_2 = x.^3; % matrix of x^3
e_3 = x.^4; % matrix of x^4
e_4 = x.*y; % matrix of xy
e_5 = e_1.*y; % matrix of x^2y
% sum of the data
sum_x = sum(x);
sum_y = sum(y);
sum_e_1 = sum(e_1);
sum_e_2 = sum(e_2);
sum_e_3 = sum(e_3);
    sum_e_4 = sum(e_4);
    sum_e_5 = sum(e_5);

    % do matrix manipulation to get coefficients
A = [n sum_x sum_e_1; sum_x sum_e_1 sum_e_2; sum_e_1 sum_e_2
sum_e_3];
B = [sum_y; sum_e_4; sum_e_5];
C = inv(A)*B;
a0 = C(1,1);
a1 = C(2,1);
a2 = C(3,1);

% calculate R^2
St = 0;
Sr = 0;
for c = 1:n
    % for St
    cal_1 = (y(1,c) - (sum(y)/n))^2; % (yi-avg(y))^2
    St = St + cal_1;
    % for Sr
    cal_2 = (y(1,c) - (a0 + a1*x(1,c) + a2*x(1,c)^2))^2; % (yi-y)^2
    Sr = Sr + cal_2;
end
error = (St - Sr)/St;

elseif command_p == 3
e_1 = x.^2; % matrix of x^2
e_2 = x.^3; % matrix of x^3
e_3 = x.^4; % matrix of x^4
e_4 = x.^5; % matrix of x^5
e_5 = x.^6; % matrix of x^6
e_6 = x.*y; % matrix of xy
e_7 = e_1.*y; % matrix of x^2y
e_8 = e_2.*y; % matrix of x^3y
% sum of the data
sum_x = sum(x);
sum_y = sum(y);
sum_e_1 = sum(e_1);
sum_e_2 = sum(e_2);

```

```

sum_e_3 = sum(e_3);
sum_e_4 = sum(e_4);
sum_e_5 = sum(e_5);
sum_e_6 = sum(e_6);
sum_e_7 = sum(e_7);
sum_e_8 = sum(e_8);

% do matrix manipulation to get coefficients
A = [n sum_x sum_e_1 sum_e_2; sum_x sum_e_1 sum_e_2 sum_e_3; sum_e_1
sum_e_2 sum_e_3 sum_e_4; sum_e_2 sum_e_3 sum_e_4 sum_e_5];
B = [sum_y; sum_e_6; sum_e_7; sum_e_8];
C = inv(A)*B;
a0 = C(1,1);
a1 = C(2,1);
a2 = C(3,1);
a3 = C(4,1);

% calculate R^2
St = 0;
Sr = 0;
for c = 1:n
    % for St
    cal_1 = (y(1,c) - (sum(y)/n))^2; % (yi-avg(y))^2
    St = St + cal_1;
    % for Sr
    cal_2 = (y(1,c) - (a0 + a1*x(1,c) + a2*x(1,c)^2 +
a3*x(1,c)^3))^2; % (yi-y)^2
    Sr = Sr + cal_2;
end
error = (St - Sr)/St;
end
end

% 2. Exponential
if command == 2
    e_1 = x; % matrix of x* = x
    e_2 = log(y); % matrix of y* = ln(y)
    e_3 = x.^2; % matrix of x^2
    e_4 = e_2.*x; % matrix of xy*
    % the sum of the data
    sum_e_1 = 0;
    sum_e_2 = 0;
    sum_e_3 = 0;
    sum_e_4 = 0;

    %check for warning
    for c = 1:n
        if y(1,c) <= 0
            n = n - 1;
            disp("Warning: Division by zero or invalid log");
        else
            sum_e_1 = sum_e_1 + e_1(1,c);
            sum_e_2 = sum_e_2 + e_2(1,c);
            sum_e_3 = sum_e_3 + e_3(1,c);

```

```

        sum_e_4 = sum_e_4 + e_4(1,c);
    end
end

a1 = (n*sum_e_4 - sum_e_1*sum_e_2)/(n*sum_e_3-(sum_e_1)^2);
a0 = sum_e_2/n - a1*sum_e_1/n;
a = exp(a0);
b = a1;

% calculate R^2
St = 0;
Sr = 0;
for c = 1:n
    % for St
    cal_1 = (y(1,c) - (sum(y)/n))^2; % (yi-avg(y))^2
    St = St + cal_1;
    % for Sr
    cal_2 = (y(1,c) - a*exp(b*x(1,c)))^2; % (yi-y)^2
Sr = Sr + cal_2;
end
error = (St - Sr)/St;
end

% 3. Saturation
if command == 3
%check for warning
e_1 = 1./x; % matrix of x* = 1/x
e_2 = 1./y; % matrix of y* = 1/y
e_3 = e_1.^2; % matrix of x*^2
e_4 = e_1.*e_2; % matrix of x*y*
% the sum of the data
sum_e_1 = 0;
sum_e_2 = 0;
sum_e_3 = 0;
sum_e_4 = 0;
%check for warning
for c = 1:n
    if y(1,c) <= 0
        n = n - 1;
        disp("Warning: Division by zero or invalid log");
    else
        sum_e_1 = sum_e_1 + e_1(1,c);
        sum_e_2 = sum_e_2 + e_2(1,c);
        sum_e_3 = sum_e_3 + e_3(1,c);
        sum_e_4 = sum_e_4 + e_4(1,c);
    end
end
a1 = (n*sum_e_4 - sum_e_1*sum_e_2)/(n*sum_e_3-(sum_e_1)^2);
a0 = sum_e_2/n - a1*sum_e_1/n;
a = 1/a0;
b = a1*a;

% calculate R^2
St = 0;
Sr = 0;

```

```

for c = 1:n
% for St
cal_1 = (y(1,c) - (sum(y)/n))^2; % (yi-avg(y))^2
St = St + cal_1;
% for Sr
cal_2 = (y(1,c) - (a*x(1,c))/(b+x(1,c)))^2; % (yi-y)^2
Sr = Sr + cal_2;
end
error = (St - Sr)/St;
end

% Plotting Raw Data
plot(x,y, '.');
xlabel('x');
ylabel('y');
hold on;

if command == 1

if command_p == 1
plot(x, a0+a1*x);
gravstr = 'Polynomial, y = %0.4f + %0.4fx, R^{2} = %0.4f';
gravstr=sprintf(gravstr, a0, a1, error);
legend("Actual Data", gravstr);

elseif command_p == 2
plot(x, a0+a1*x+a2*(x.^2));
gravstr = 'Polynomial, y = %0.4f + %0.4fx + %0.4fx^{2}, R^{2} = %0.4f';
gravstr=sprintf(gravstr, a0, a1, a2, error);
legend("Actual Data", gravstr);

elseif command_p == 3
plot(x, a0+a1*x+a2*(x.^2)+a3*(x.^3));
gravstr = 'Polynomial, y = %0.4f + %0.4fx + %0.4fx^{2} + %0.4fx^{3}, R^{2} = %0.4f';
gravstr=sprintf(gravstr, a0, a1, a2, a3, error);
legend("Actual Data", gravstr);
end

elseif command == 2
plot(x,a*exp(b*x));
gravstr = 'Exponential, y = %0.4fe^{%0.4fx}, R^{2} = %0.4f';
gravstr=sprintf(gravstr, a, b, error);
legend("Actual Data", gravstr);

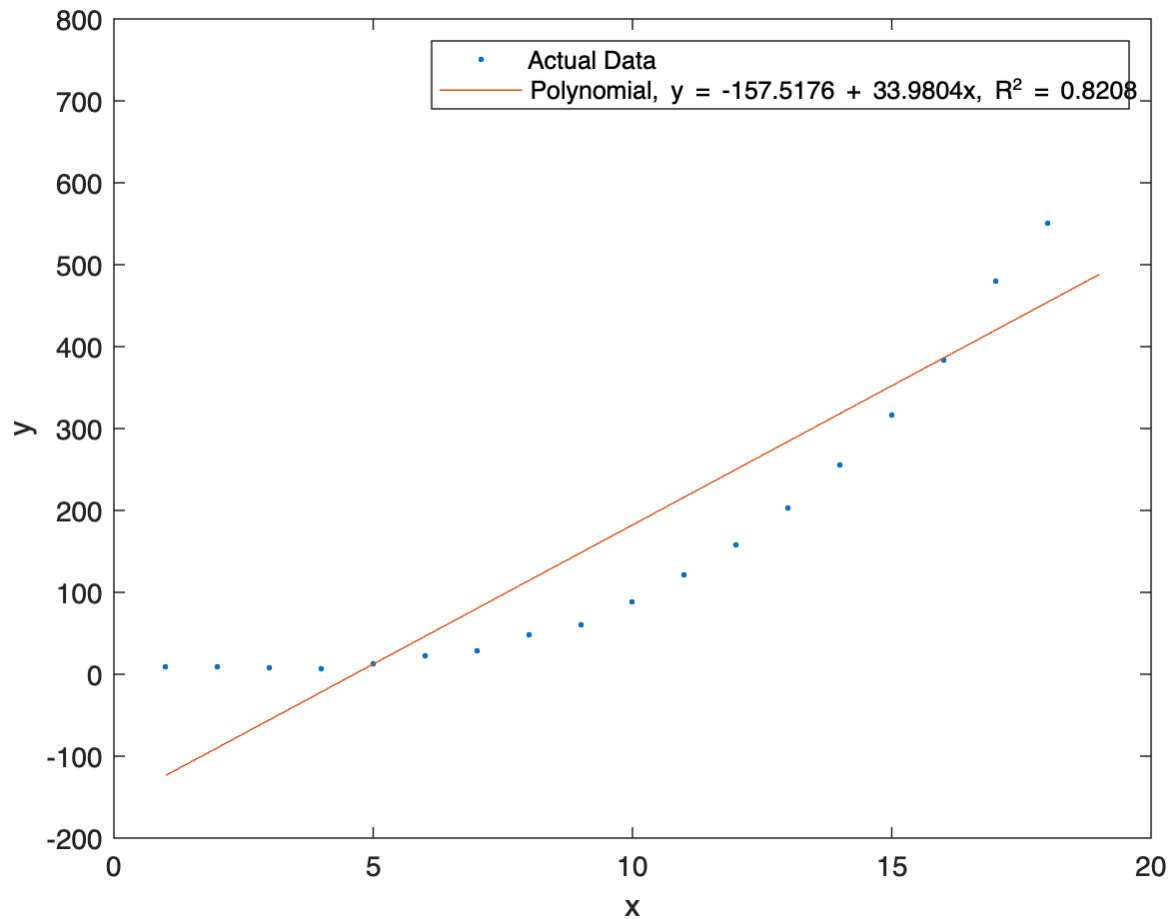
elseif command == 3
plot(x,(a*x)./(b+x));
gravstr = 'Saturation, y = %0.4fx/(%0.4f + x), R^{2} = %0.4f';
gravstr=sprintf(gravstr, a, b, error);
legend("Actual Data", gravstr);
end

```

- The output is:

1. test1.txt

- 1st Order Polynomial



1. Polynomial
2. Exponential
3. Saturation

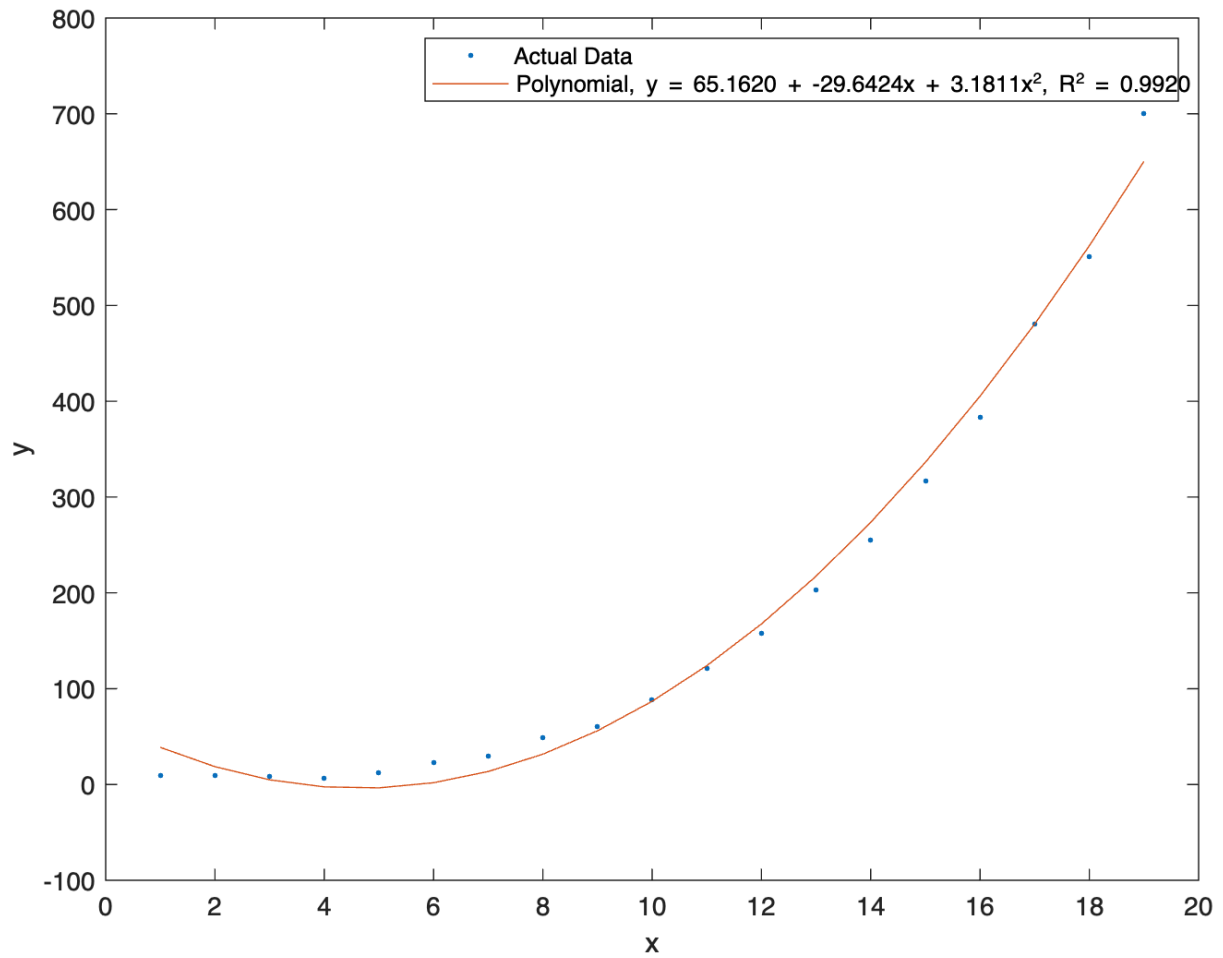
Select the function to fit your data:

1

Degree of polynomial:

1

- 2nd Order Polynomial



1. Polynomial
2. Exponential
3. Saturation

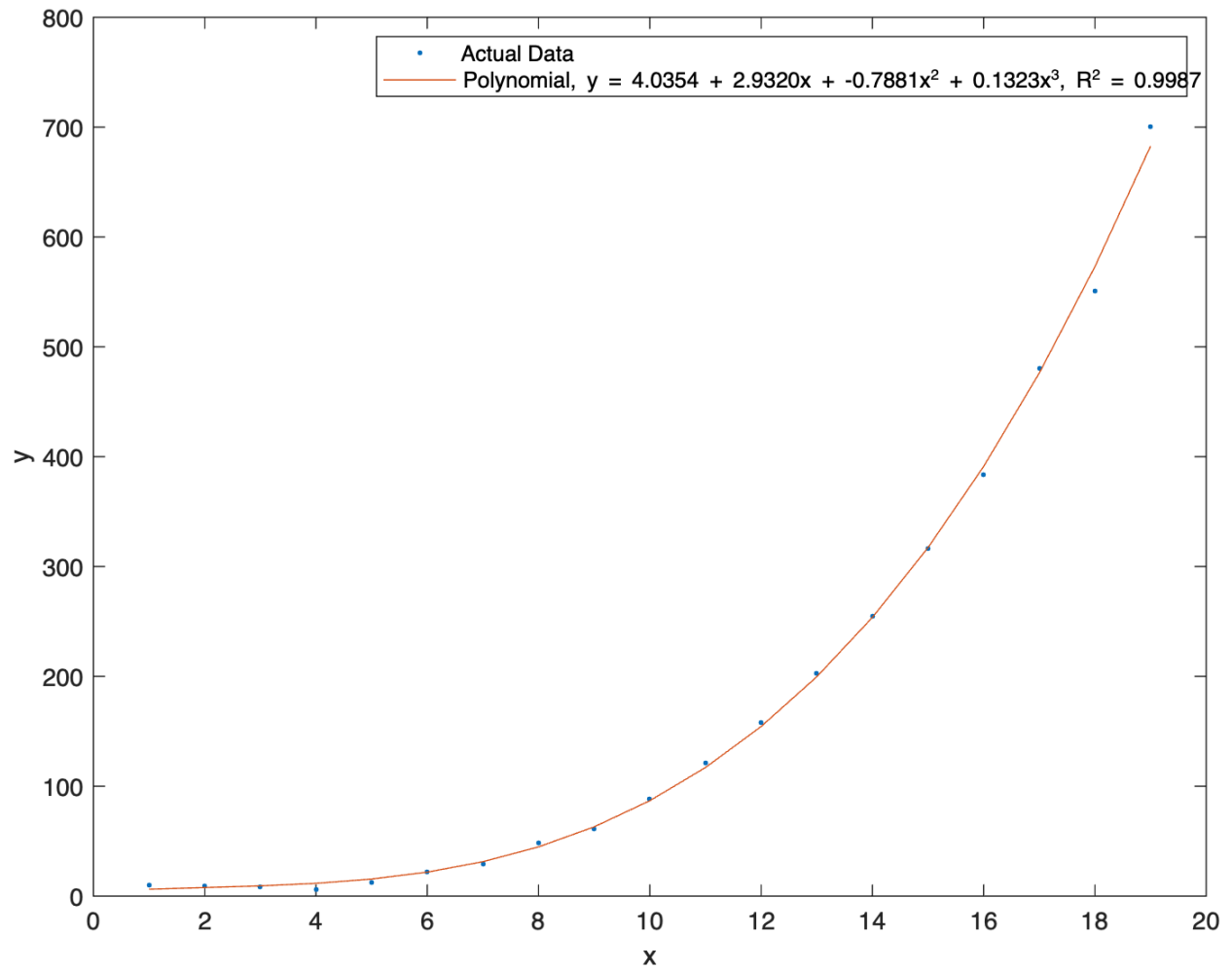
Select the function to fit your data:

1

Degree of polynomial:

2

- 3rd Order Polynomial



1. Polynomial
2. Exponential
3. Saturation

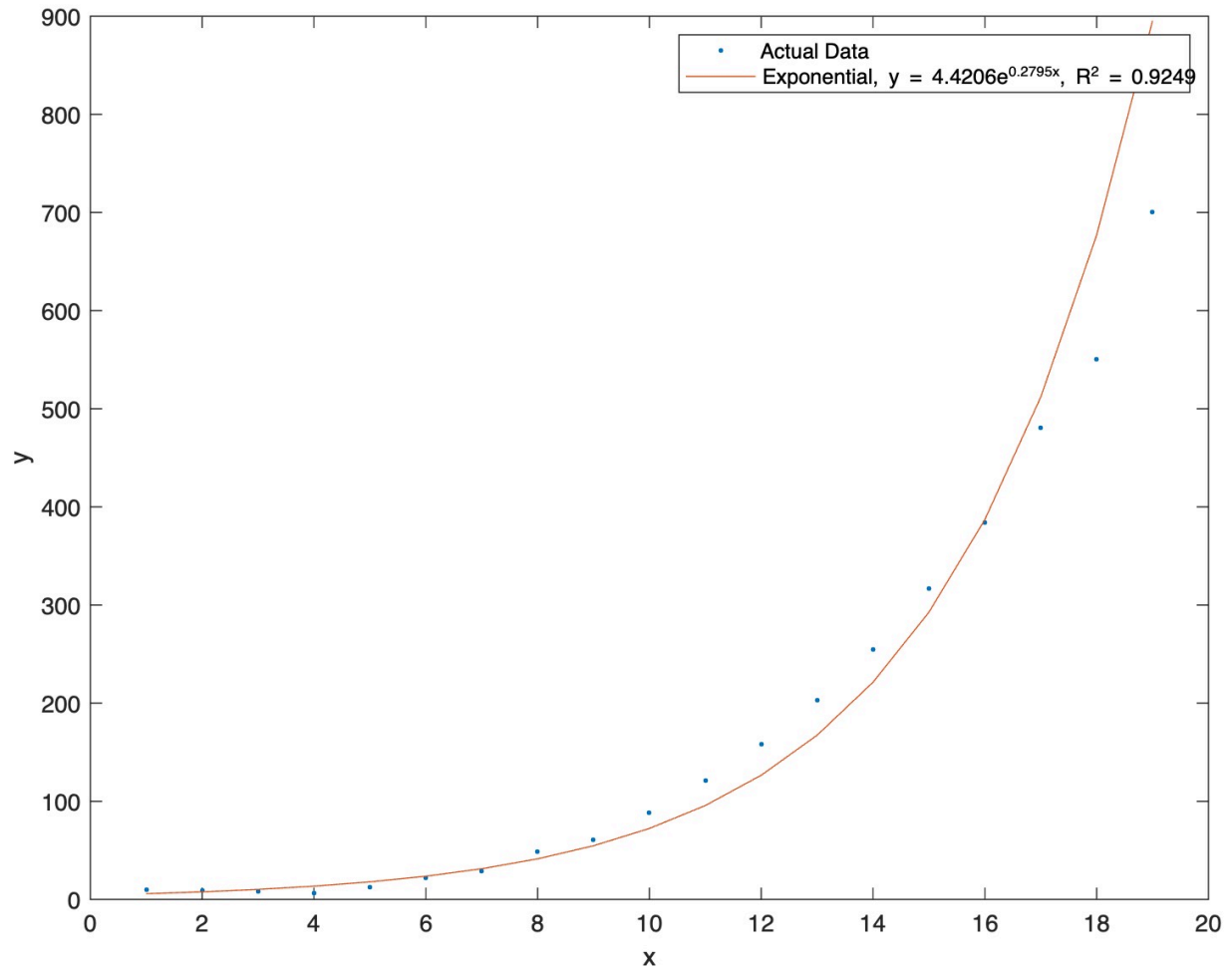
Select the function to fit your data:

1

Degree of polynomial:

3

- Exponential

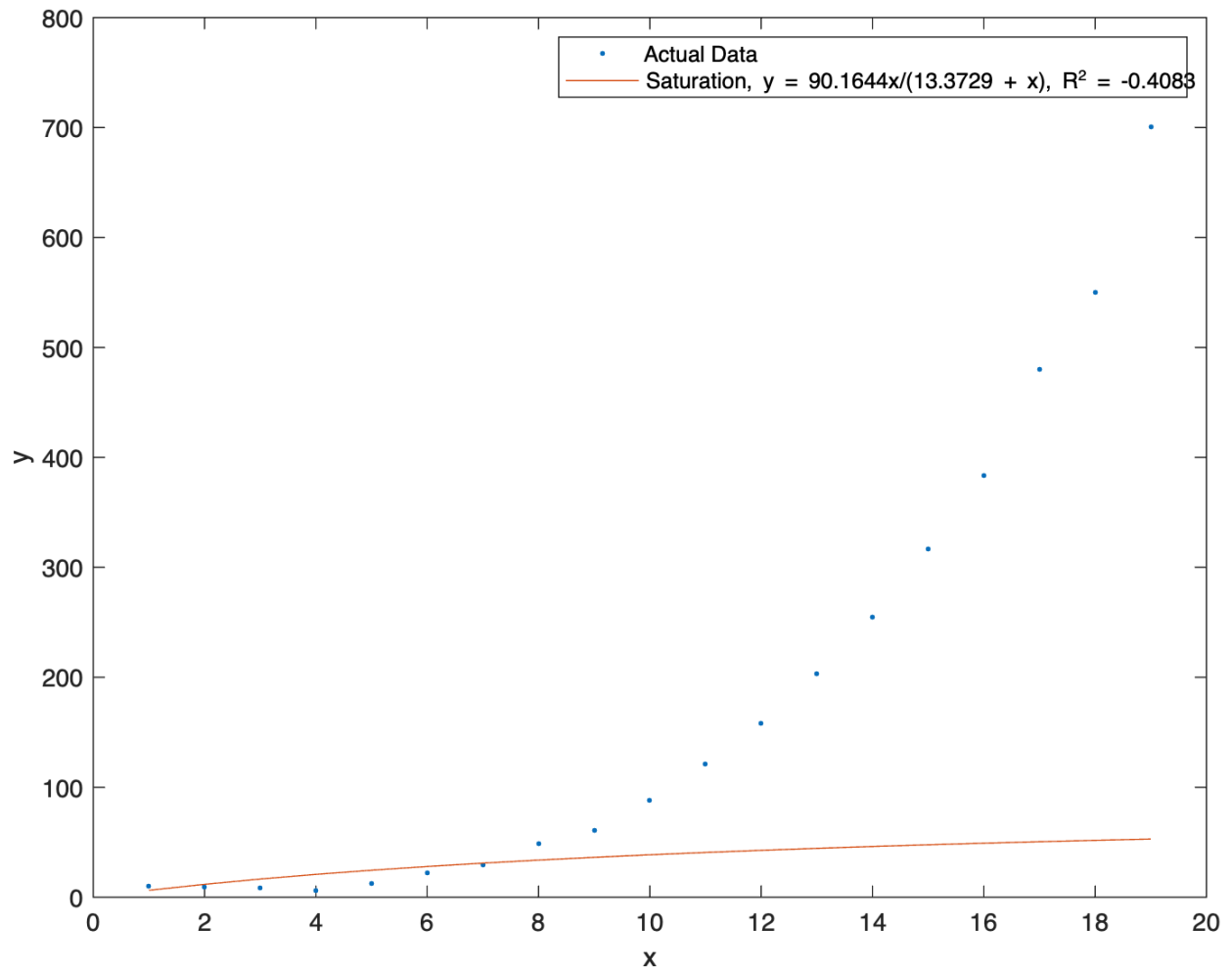


1. Polynomial
2. Exponential
3. Saturation

Select the function to fit your data:

2

- Saturation



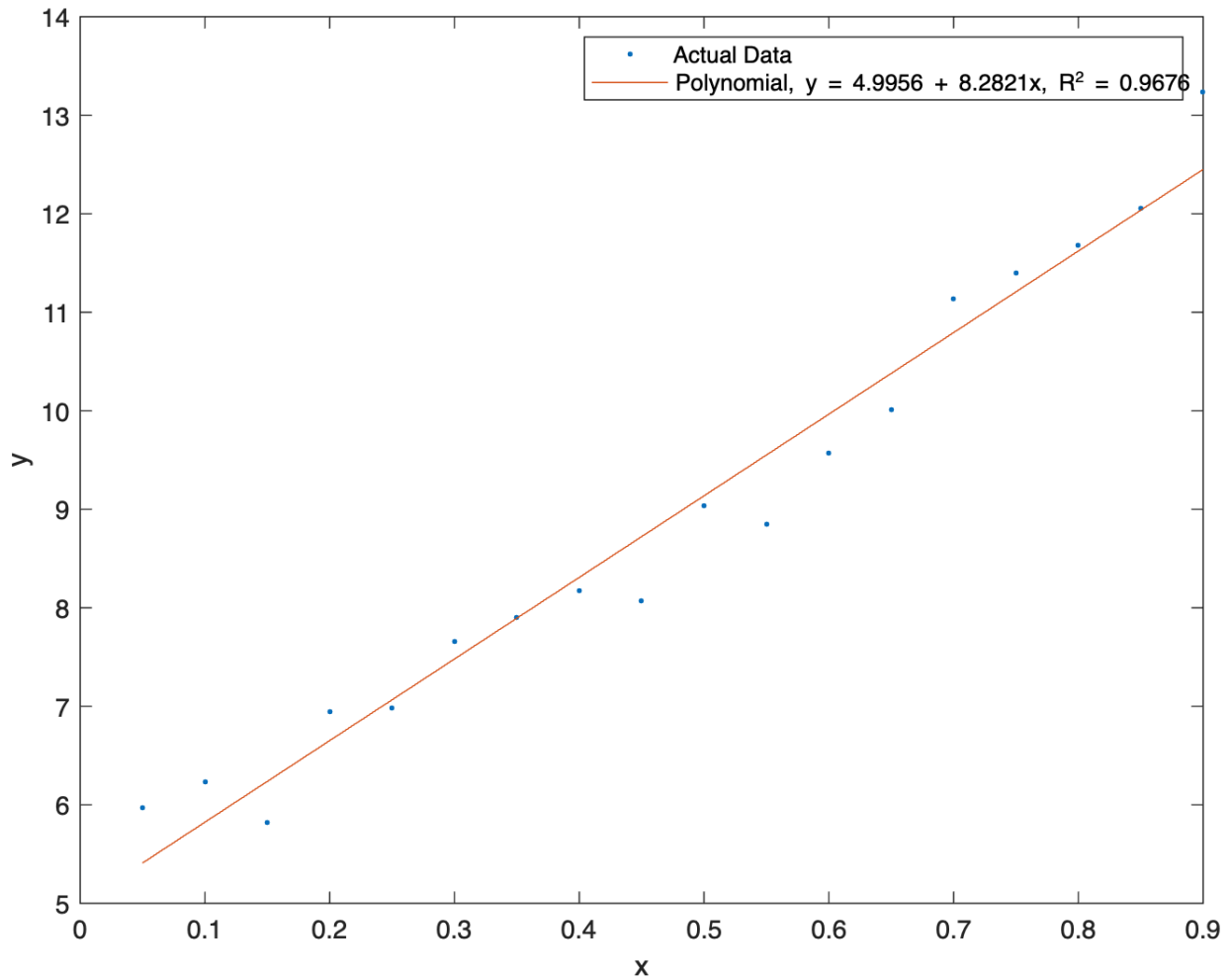
1. Polynomial
2. Exponential
3. Saturation

Select the function to fit your data:

3

2. test2.txt

- 1st Order Polynomial



1. Polynomial
2. Exponential
3. Saturation

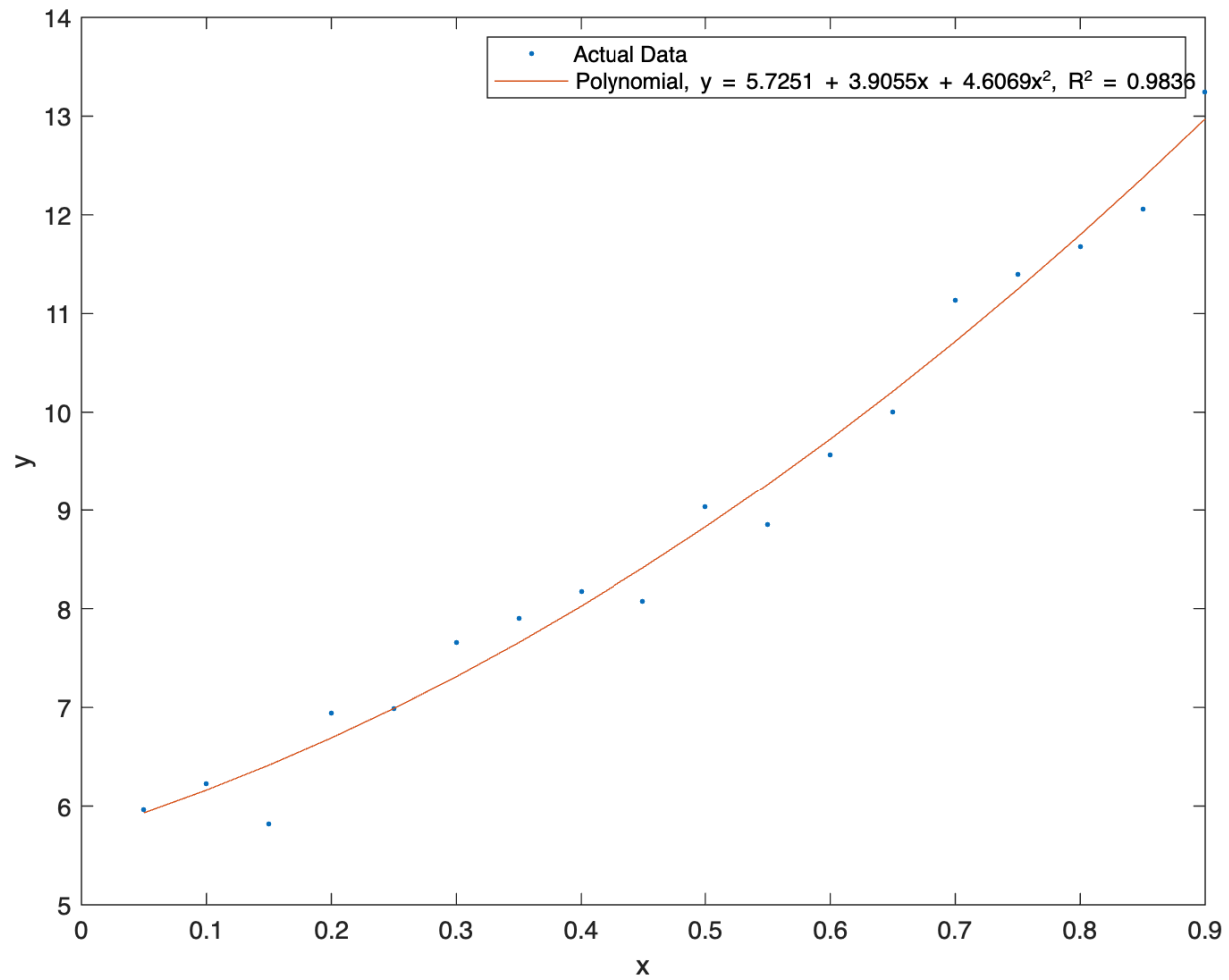
Select the function to fit your data:

1

Degree of polynomial:

1

- 2nd Order Polynomial



1. Polynomial
2. Exponential
3. Saturation

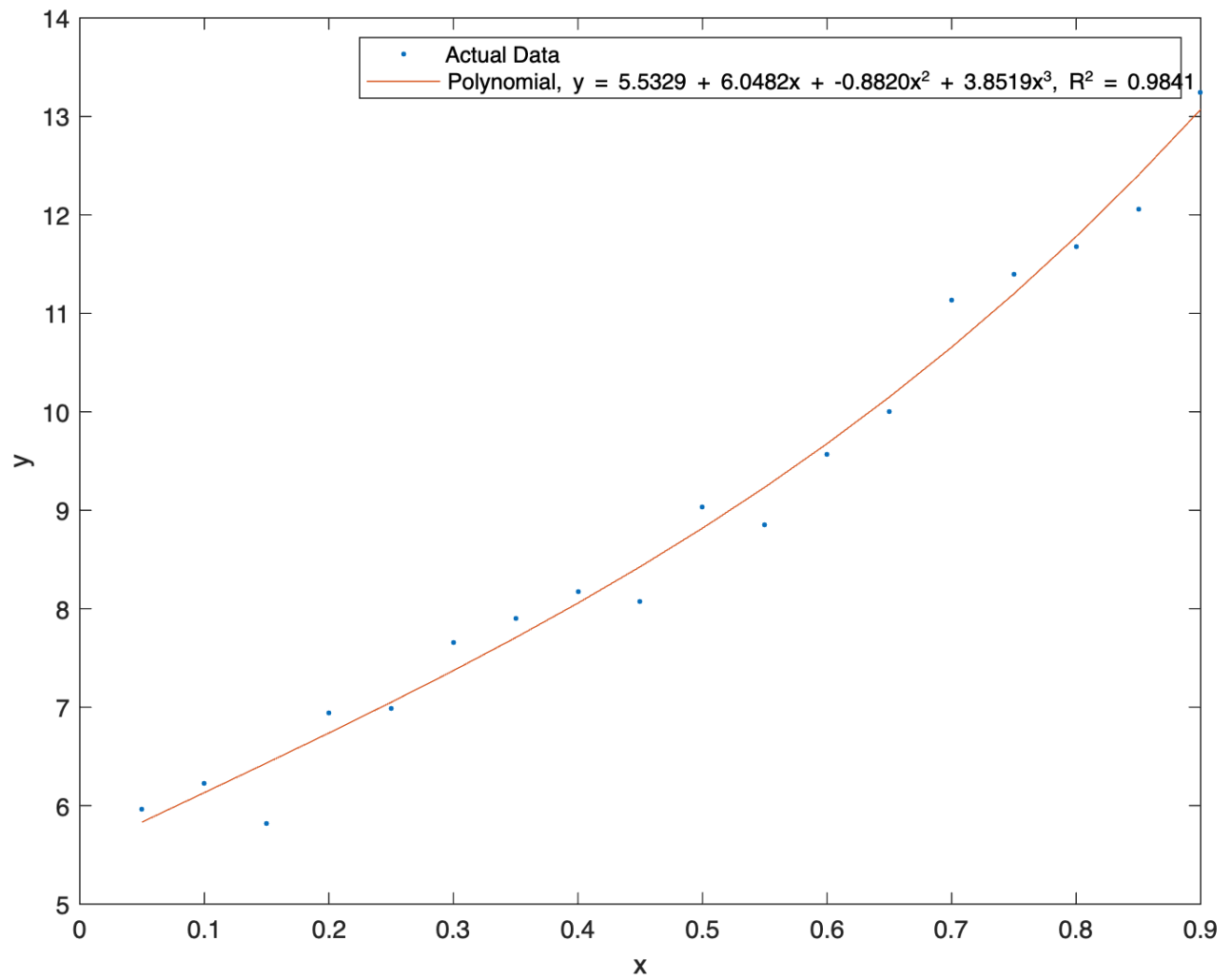
Select the function to fit your data:

1

Degree of polynomial:

2

- 3rd Order Polynomial



1. Polynomial
2. Exponential
3. Saturation

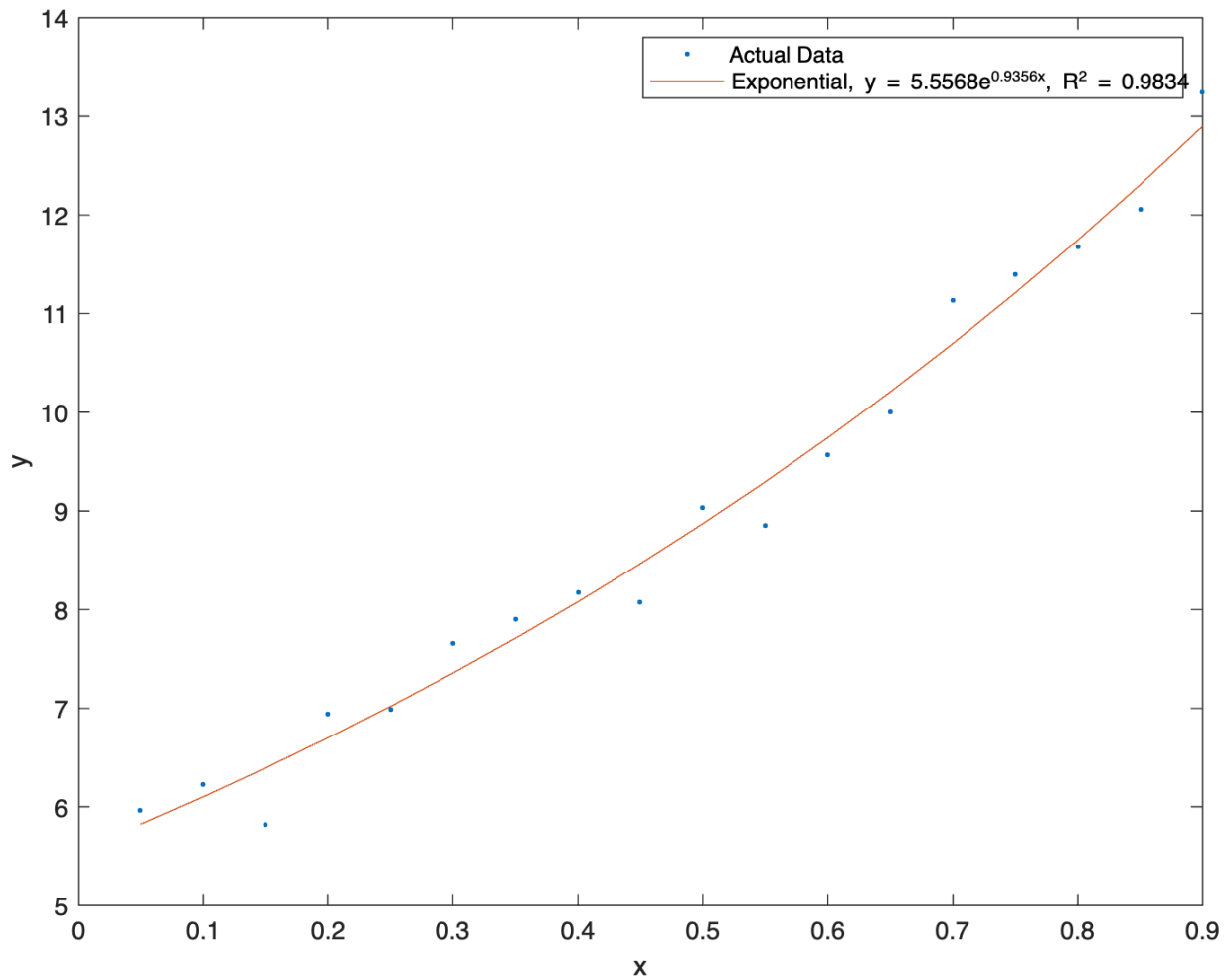
Select the function to fit your data:

1

Degree of polynomial:

3

- Exponential

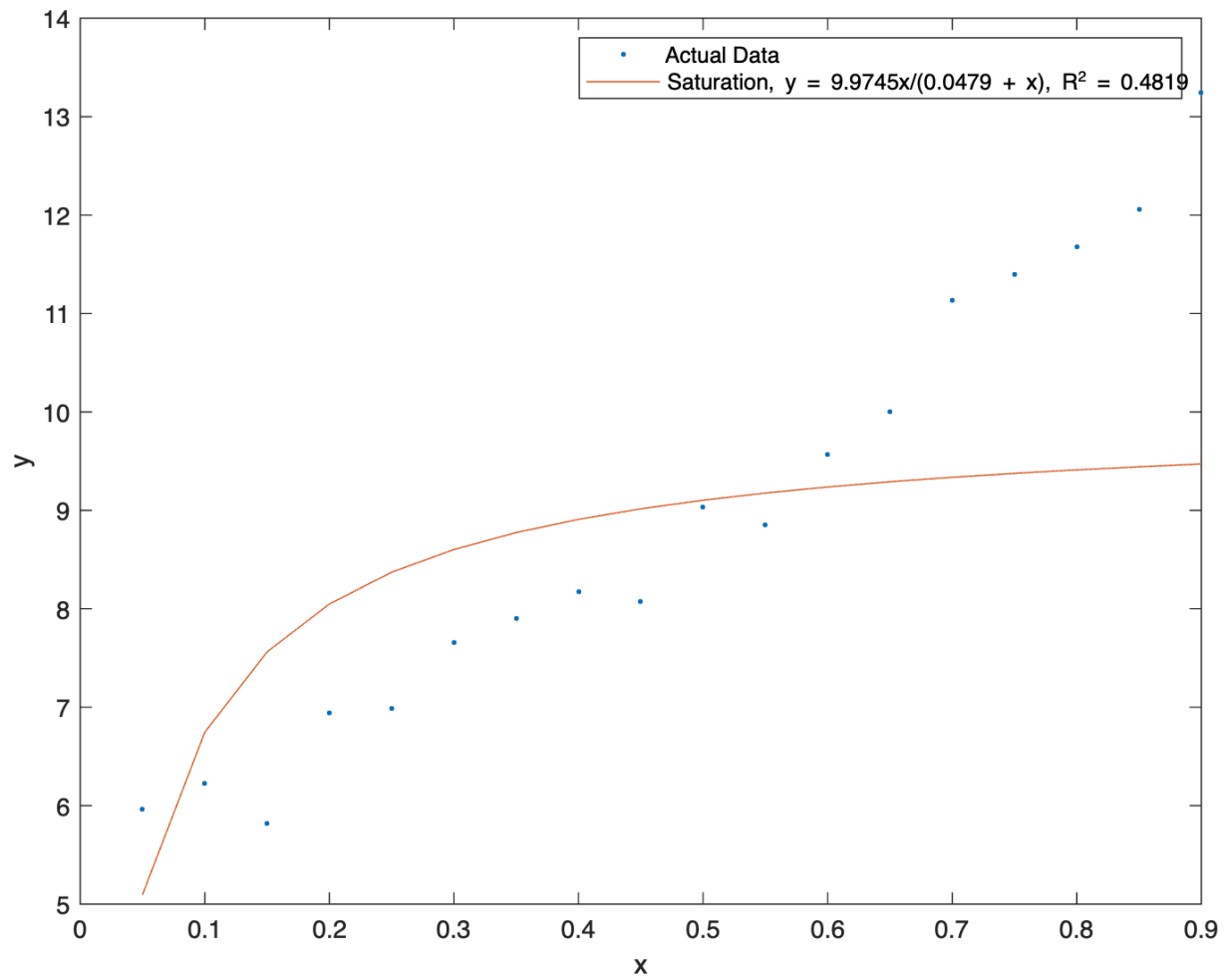


1. Polynomial
2. Exponential
3. Saturation

Select the function to fit your data:

2

- Saturation



1. Polynomial
2. Exponential
3. Saturation

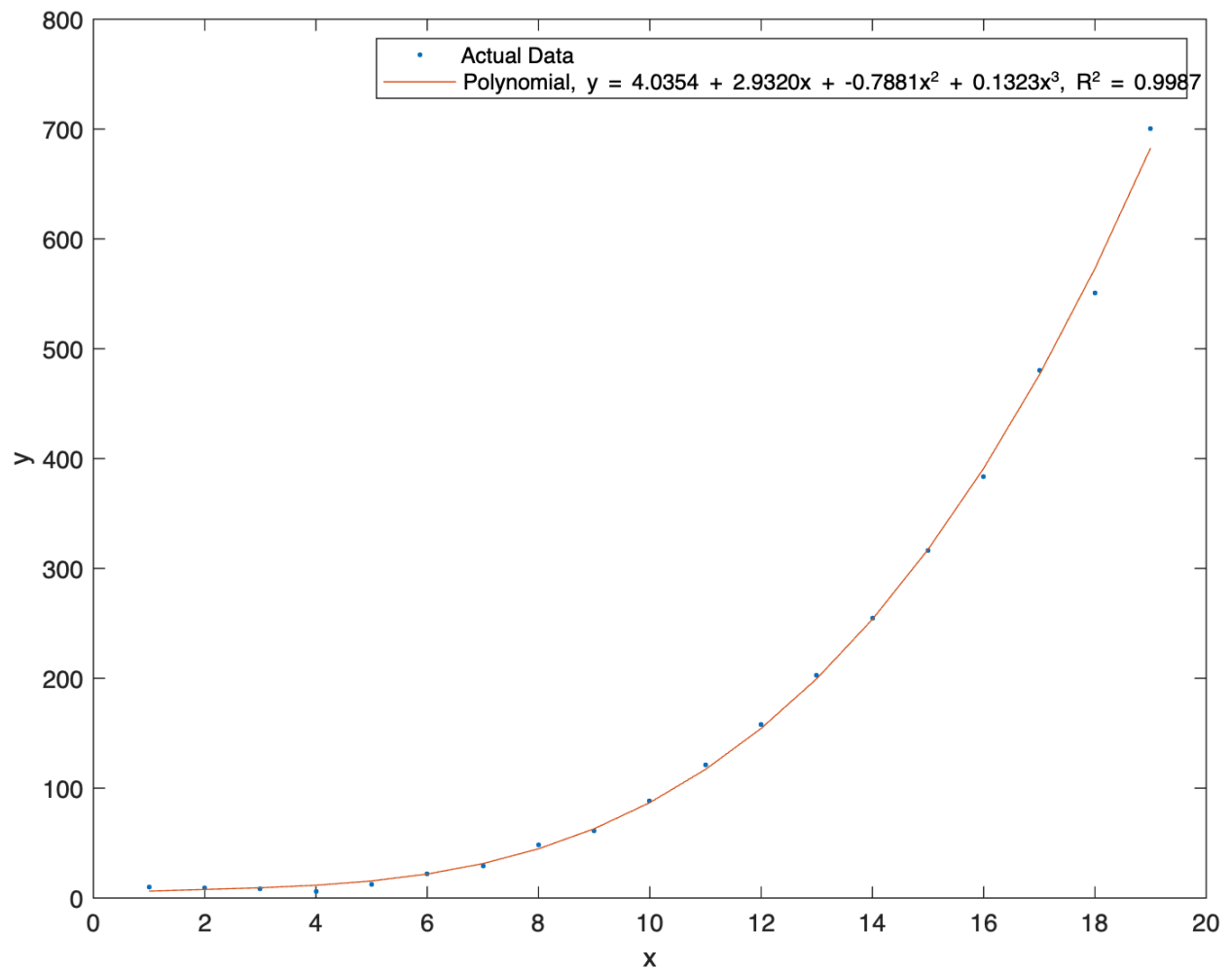
Select the function to fit your data:

3

Analysis Discussion

To find the best fit model for the two set of data, find the model with a R^2 value that is closest to 1 since a higher R^2 represents better improvement of the model. Therefore, the best fir models for “test1.txt” and “test2.txt” are the ones with R^2 closest to 1 (highest R^2).

For “test1.txt”, the best fit model is the 3rd Order Polynomial



For “test2.txt”, the best fit model is the 3rd Order Polynomial

