Part 2: Polynomial Fitting

Variables x and y contain 12 values from an unknown cubic polynomial, i.e.

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$$

Using the values x and y, compute estimates for parameters $\beta_0, \ldots, \beta_3$ using linear regression. **For this problem, you are not allowed to use `fitlm`, `regress`, `polyfit`, or any other linear regression or curve fitting tools.** You must construct the design matrix and calculate parameter estimates via pseudoinversion.

```
X2 = [ones(1,12);x';(x.^2)';(x.^3)']' %set up design matrix
```

```
X2 = 12×4
    1.0000   -2.0000    4.0000   -8.0000
    1.0000   -1.6364    2.6777   -4.3817
    1.0000   -1.2727    1.6198   -2.0616
    1.0000   -0.9091    0.8264   -0.7513
    1.0000   -0.5455    0.2975   -0.1623
    1.0000   -0.1818    0.0331   -0.0060
    1.0000    0.1818    0.0331    0.0060
    1.0000    0.5455    0.2975    0.1623
```

```
    1.0000    0.9091    0.8264    0.7513
    1.0000    1.2727    1.6198    2.0616
      ⋮
```

```
X_p = pinv(X2) %set up pseudoinverse
```

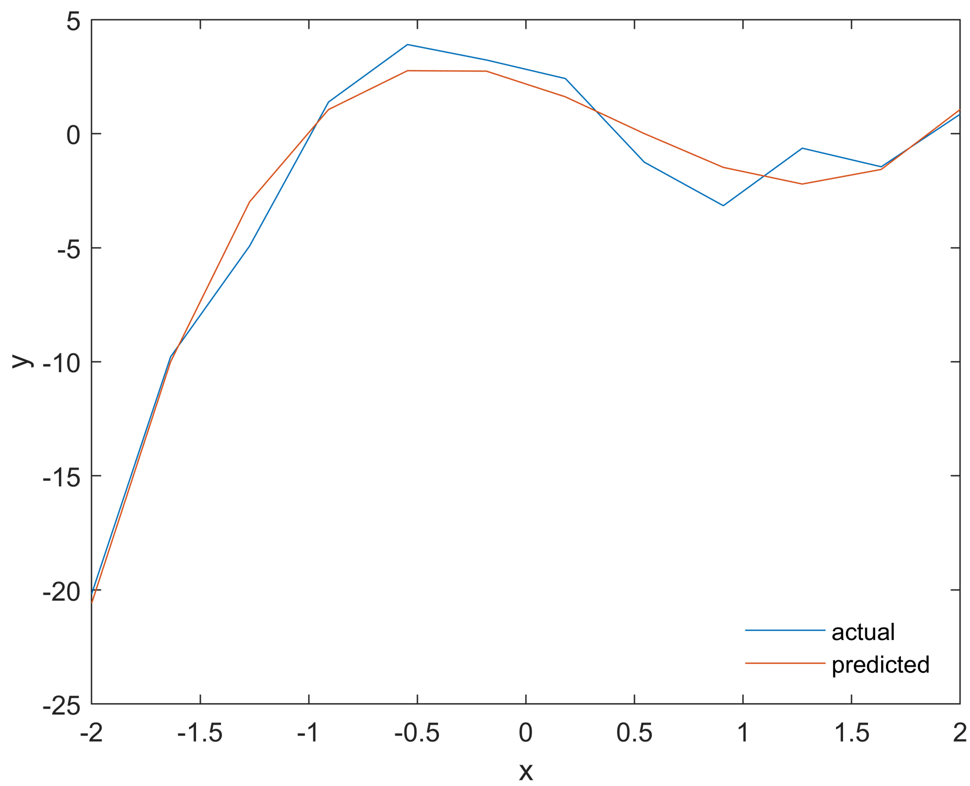
```
X_p = 4×12
   -0.0804    0.0089    0.0804    0.1339    0.1696    0.1875    0.1875    0.1696 ⋯
    0.1440   -0.1092   -0.2262   -0.2373   -0.1726   -0.0626    0.0626    0.1726
    0.1039    0.0472    0.0019   -0.0321   -0.0548   -0.0661   -0.0661   -0.0548
   -0.0889    0.0081    0.0566    0.0673    0.0512    0.0189   -0.0189   -0.0512
```

```
B = X_p*y %calculate betas 0-3
```

```
B = 4×1
    2.2753
   -3.1669
   -3.0092
    2.1444
```

Using your parameter estimates, plot the points in variables x and y and a line corresponding to the best fit polynomial. Both the points and the line should be on the same plot.

```
plot(x,y) %actual data
hold on
plot(x, X2*B) %predicted
xlabel('x');
ylabel('y');
legend('actual', 'predicted', 'Location', 'southeast');
legend('boxoff')
hold off
```

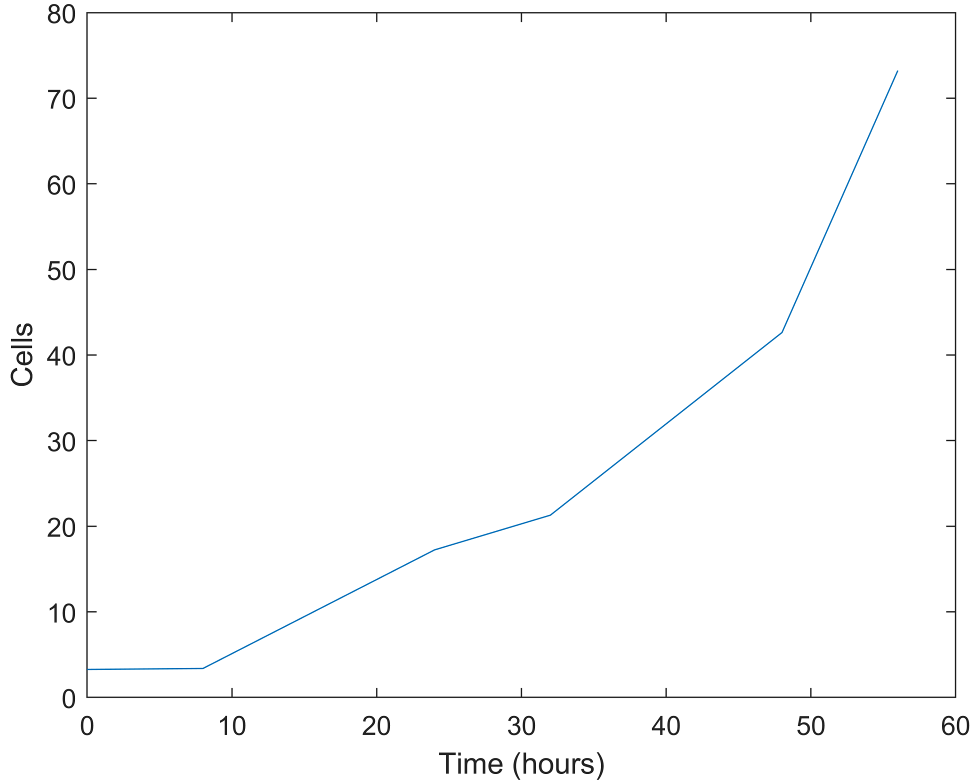


Part 3: Cell Growth

Variables `t` and `cells` contain six cell counts for dividing mammalian cells in a culture dish. (The times in `t` are in hours.) Your task is to find the exponential growth rate of the cells using linear regression. **For this problem, you are not allowed to use `fitlm`, `regress`, `polyfit`, or any other linear regression or curve fitting tools.**

a.) Plot the number of cells over time.

```
plot(t, cells)
xlabel('Time (hours)');
ylabel('Cells');
```



b.) Set up a design matrix for the linearized exponential growth equation from section 9.4.

```
X3 = [ones(1,6); t']'
```

```
X3 = 6×2
     1     0
     1     8
     1    24
     1    32
     1    48
     1    56
```

c.) Calculate the pseudoinverse of the design matrix and use it to fit your model.

```
X3_p = pinv(X3)
```

```
X3_p = 2×6
    0.4933    0.4000    0.2133    0.1200   -0.0667   -0.1600
   -0.0117   -0.0083   -0.0017    0.0017    0.0083    0.0117
```

```
cells3 = log(cells)
```

```
cells3 = 6×1
    1.1856
    1.2199
    2.8472
    3.0584
    3.7525
    4.2936
```

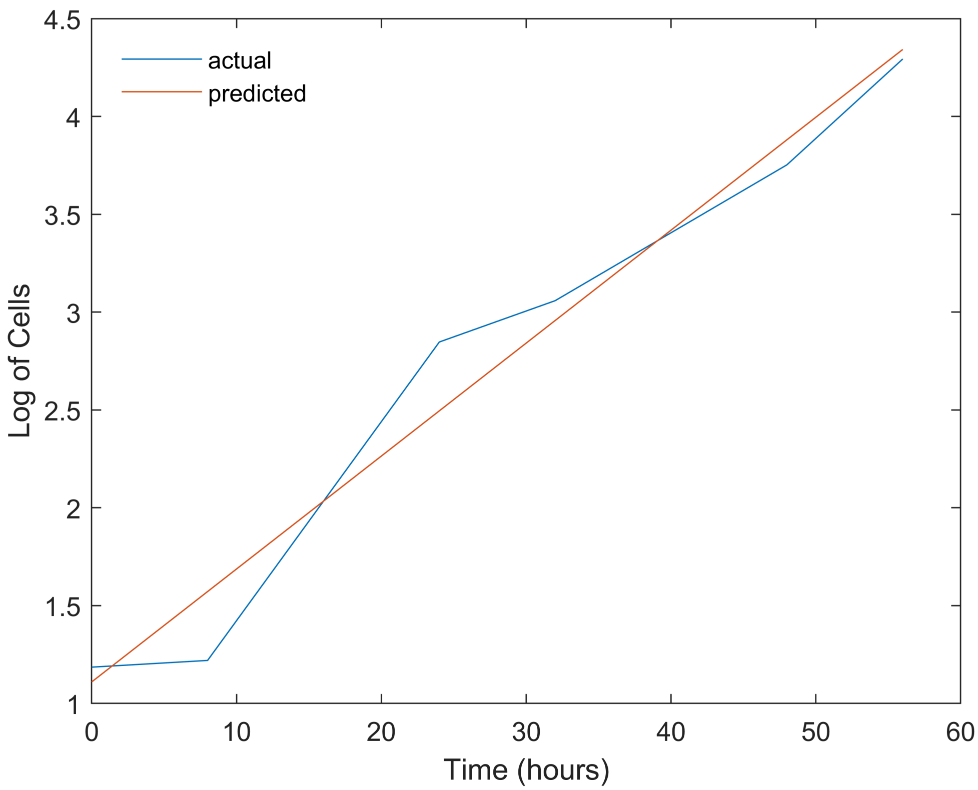
```
B3 = X3_p*(cells3)
```

```
B3 = 2×1
    1.1101
    0.0577
```

```
plot(t, cells3) %actual data
hold on
y3 = X3*B3
```

```
y3 = 6×1
    1.1101
    1.5719
    2.4953
    2.9571
    3.8805
    4.3423
```

```
plot(t, y3) %predicted
xlabel('Time (hours)');
ylabel('Log of Cells');
legend('actual', 'predicted', 'Location', 'northwest');
legend('boxoff')
hold off
```



d.) Calculate the exponential growth rate of the cells. What are the units?

$\mu = 0.0577 hour^{-1}$

e.) Use the fitted parameters to find the inital number of cells. How does this value compare with the number of cells at t = 0 h in your data?

$exp(ln(N_0) + \mu * t) = exp(1.1101) = 3.035 cells$ This estimate is 7.54% difference.