1. Visualization
    a. **What do you observe? For example, can you make an educated guess on the effectiveness of linear and polynomial regression in predicting the data?**
    The training data looks relatively linear; however the testing data looks like polynomial regression may be more effective. While the overall relationship between the X and Y data appears linear, there are some outliers that polynomial regression will be more robust to.
    b. N/A
2. Linear Regression
    a. N/A
    b. N/A
    c. **Try different alpha = 10^{-4}; 10^{-3}; 10^{-2}; 10^{-1}, and make a table of the coefficients and number of iterations until convergence. How do the coefficients compare? How quickly does each algorithm converge?**
        i.

Alpha	Coefficients	Iterations	Time
0.1	[2.38405089 -2.87906028]	152	0.0521
0.01	[2.44078184 -2.81863861]	616	0.2119
0.001	[2.44555307 -2.81594059]	7330	2.4443
0.0001	[2.44509281 -2.81382739]	64620	20.8695

    We see that the coefficients converge to 2.44, -2.81. The alpha value of 0.01 is sufficient for finding these coefficients. The .001 and .0001 learning rates make the algorithm take much longer to converge although the coefficients are not much different than the other learning rates.
    d.
        i. **How do the coecients compare to those obtained by SGD?**
        The coefficients given are [2.44640709 -2.81635359]. These are very similar to those returned by the SGD method. As alpha becomes smaller they become closer.
        ii. **TIME:**
            1. SGD (0.01): 0.211958885193

2. Closed form: 0.000275135040283

The Closed form method is much faster than even using a 0.01 learning rate for SGD.

e. **Finally, Find a learning rate for SGD that is a function of k (the number of iterations) and converges to the same solution yielded by the closed-form optimization. For this problem, we care more about your learning rate than about whether you match the closed-form solution (that is, your solution may not match exactly, but you should be able to get pretty close). Update PolynomialRegression.fit_SGD(...) with your proposed learning rate. How long does it take the algorithm to converge with your proposed learning rate?**

   i. With a learning rate of 0.0015 the coefficients for SGD are very close to the coefficients for the Closed Form solution. For 0.0015 the coefficients for SGD are [2.44538913 -2.81627046], which is about the closest we can get. With a smaller learning rate the first coefficient will get closer, but the second will get further away. The algorithm takes 4912 and 1.653 seconds to converge.

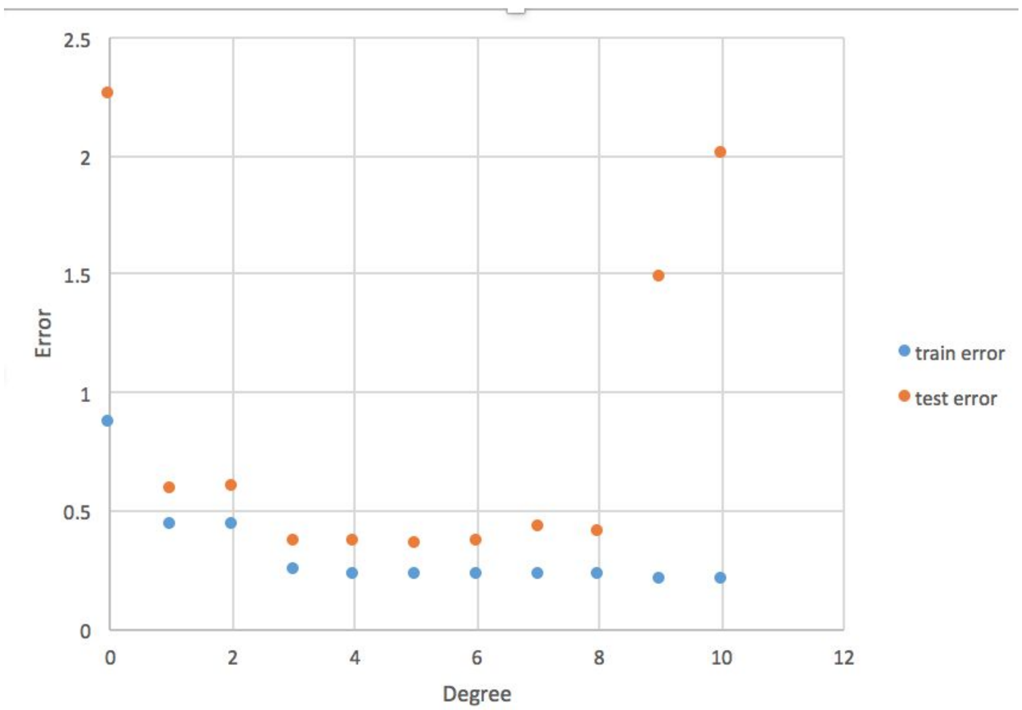
3. Polynomial Regression

   a. N/A

   b. **Why do you think we might prefer RMSE as a metric over J(theta)?**

We know higher degree polynomials are more robust to outliers. Thus, higher degree polynomials will tend to do better with J(theta) as a metric because J(theta) penalizes outliers heavily. However, the lower J(theta) values for higher degree polynomials do not mean that the model is better because this could simply be the model overfitting to the outliers. Thus we may prefer RMSE as a metric because it does not penalize outliers as heavily as J(theta). The errors are squared then square rooted so all errors will be weighed the same.

   c. Generate a plot depicting how RMSE varies with model complexity (polynomial degree) { you should generate a single plot with both training and test error, and include this plot in your writeup. Which degree polynomial would you say best ts the data? Was there evidence of under/overfitting the data? Use your plot to defend your answer.



Based on the plot it looks like polynomials of degree 3 through 5 all have about the same difference between the train error and test error, making those the best degree polynomials to use. Degree 0 has a drastic level of underfitting, as we can see by the vast difference between the test error and the train error for that polynomial, and degrees 9 and 10 have overfitting for the same reason.

4. Extra Credit -- Regularization
   a. N/A
   b. How does regularization affect training and test error? Which lambda value appears to work best?