Quiz 1

Suppose m=4 students have taken some class, and the class had a midterm exam and a final exam. You have collected a dataset of their scores on the two exams, which is as follows:

Midterm Exam	**(midterm exam)^2**	**Final Exam**
89	7921	96
72	5184	74
94	8836	87
69	4761	78

You'd like to use polynomial regression to predict a student's final exam score from their midterm exam score. Concretely, suppose you want to fit a model of the form $h\theta(x)=\theta_0+\theta_1 x_1+\theta_2 x_2$, where x_1 is the midterm score and x_2 is (midterm score)^2. Further, you plan to use both feature scaling (dividing by the "max-min", or range, of a feature) and mean normalization.

What is the normalized feature $x_2^{(4)}$? (Hint: midterm = 69, final = 78 is training example 4.) Please round off your answer to two decimal places and enter in the text box below.

-0.47

You run gradient descent for 15 iterations with $\alpha=0.3$ and compute $J(\theta)$ after each iteration. You find that the value of $J(\theta)$ decreases quickly then levels off. Based on this, which of the following conclusions seems most plausible?

$\alpha=0.3$ is an effective choice of learning rate.

Suppose you have m=14 training examples with n=3 features (excluding the additional all-ones feature for the intercept term, which you should add). The normal equation is $\theta=(X^TX)^{-1}X^Ty$. For the given values of m and n, what are the dimensions of θ, X, and y in this equation?

X is 14×4, y is 14×1, θ is 4×1

Suppose you have a dataset with m=50 examples and n=200000 features for each example. You want to use multivariate linear regression to fit the parameters θ to our data. Should you prefer gradient descent or the normal equation?

Gradient descent, since $(X^TX)^{-1}$ will be very slow to compute in the normal equation.

Which of the following are reasons for using feature scaling?

It speeds up gradient descent by making it require fewer iterations to get to a good solution.

Quiz 2

Suppose I first execute the following Octave commands:

```
A = [1 2; 3 4; 5 6];
B = [1 2 3; 4 5 6];
```

Which of the following are then valid Octave commands? Check all that apply. (Hint: A' denotes the transpose of A.)

C = A * B;

C = B' + A;

Let A be a 10x10 matrix and x be a 10-element vector. Your friend wants to compute the product Ax and writes the following code:

```
v = zeros(10, 1);
for i = 1:10
  for j = 1:10
    v(i) = v(i) + A(i, j) * x(j);
  end
end
```

How would you vectorize this code to run without any for loops? Check all that apply.

v = A * x;

Say you have two column vectors v and w, each with 7 elements (i.e., they have dimensions 7x1). Consider the following code:

```
z = 0;
for i = 1:7
  z = z + v(i) * w(i);
end
```

Which of the following vectorizations correctly compute z? Check all that apply.

z = v' * w;

In Octave, many functions work on single numbers, vectors, and matrices. For example, the sin function when applied to a matrix will return a new matrix with the sin of each element. But you have to be careful, as certain functions have different behavior. Suppose you have an 7x7 matrix X. You want to compute the log of every element, the square of every element, add 1 to every element, and divide every element by 4. You will store the results in four matrices, A,B,C,D. One way to do so is the following code:

```
for i = 1:7
  for j = 1:7
        A(i, j) = log (X(i, j));
    B(i, j) = X(i, j) ^ 2;
        C(i, j) = X(i, j) + 1;
    D(i, j) = X(i, j) / 4;
  end
end
```

Which of the following correctly compute A,B,C, or D? Check all that apply.

B = X .^ 2;

C = X + 1;

D = X / 4;