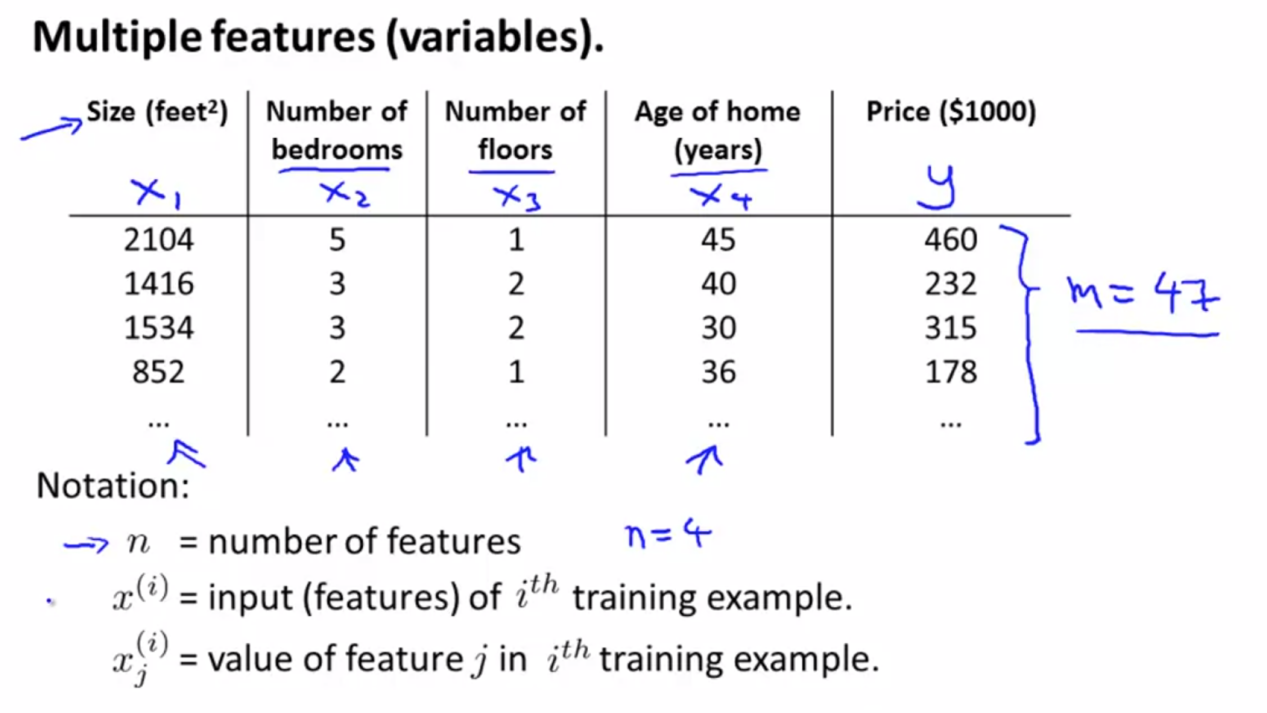
MACHINE LEARNING

WEEK 2

**Multivariate Linear Regression**

1. **Multiple Features**

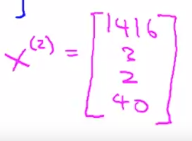
We have multple feature



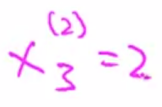
So we have multiple feature, with

n (number of columns/features) = 4

x (number of rows) = 47



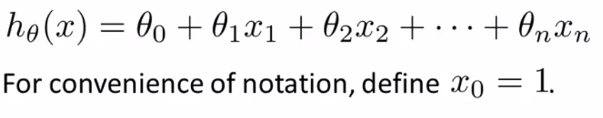
In this matrix have values in all of data in row 2



This value means, in row 2 and column 3

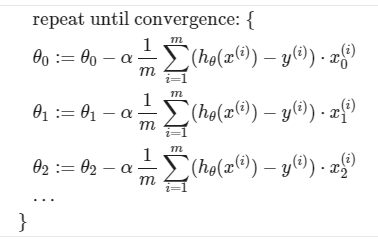
1. **Cost Function for Multiple Features**

And in multiply feature , we have hipotesis function like this.

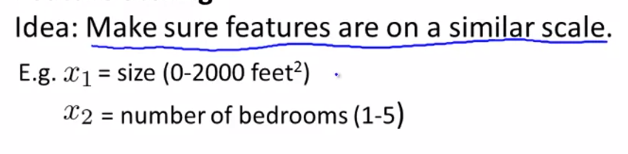




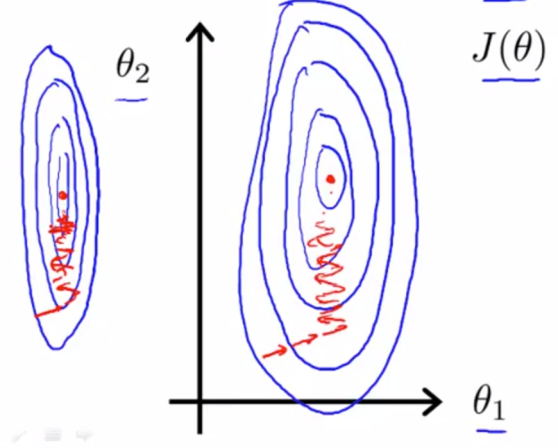
1. **Gradien Descent for Multiple Variable**



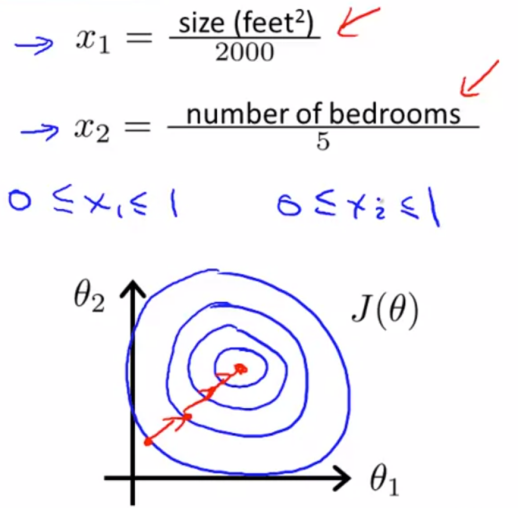
1. **Feature Scaling**



We can try to visualization this data



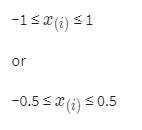
The data can very can be tall skinny counturs, and can make to be harder to find cost function and much time. So in this case we can try to some trick is it feature scaling. like this



After we scalling the data, data more be stable and more easy to find cost function.

We can speed up gradient descent by having each of our input values in roughly the same range. This is because θ will descend quickly on small ranges and slowly on large ranges, and so will oscillate inefficiently down to the optimum when the variables are very uneven.

The way to prevent this is to modify the ranges of our input variables so that they are all roughly the same. Ideally:

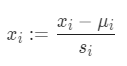


Two techniques to help with this are ****feature scaling**** and ****mean normalization****.

**- Feature scaling** involves dividing the input values by the range (i.e. the maximum value minus the minimum value) of the input variable, resulting in a new range of just 1.

**- Mean normalization** involves subtracting the average value for an input variable from the values for that input variable resulting in a new average value for the input variable of just zero.

To implement both of these techniques, adjust your input values as shown in this formula:



Where μi is the ****average**** of all the values for feature (i) and **si** is the range of values (max - min), or **si** is the standard deviation.

Note that dividing by the range, or dividing by the standard deviation, give different results. The quizzes in this course use range - the programming exercises use standard deviation.

1. Plynomial Regression

