**Multi Variant Linear Regression**

* We denote X1, X2, X3 as the number of variables used as conditions.
* For housing price example this can be size of house, number of bedrooms

, Number of floors, age of house etc.

* Remember that X1, X2, X3 were used for mth training example.
* The number of variables is denoted by n.

**Hypothesis for more than 1 variable**

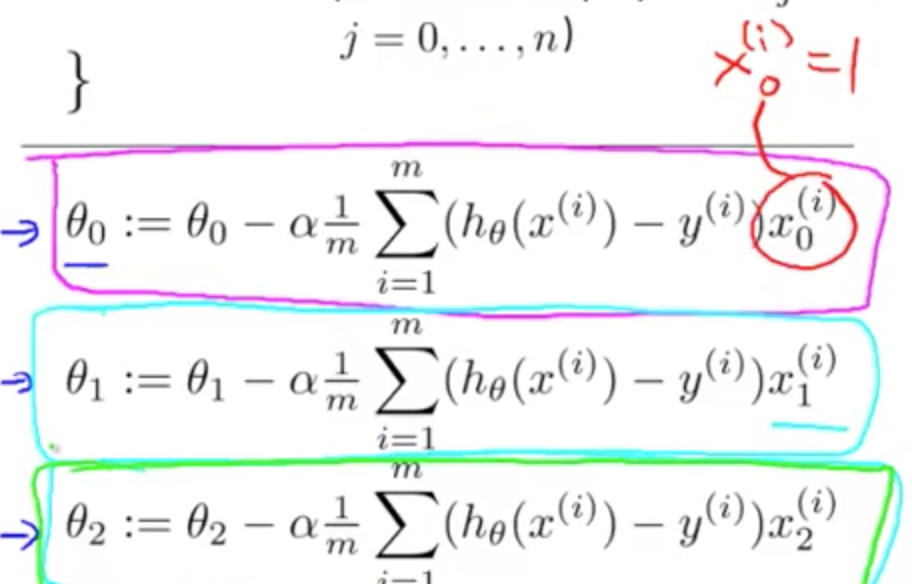
* Our hypothesis will also change as the number of variables change.
* For multiple variables the hypothesis will look something as shown below:
  + h**Θ (x) =** theta0 + theta1 x1 +theta2 x2  + theta3 x3 ….
  + We can make with this that x0(i) = 1.
  + This equation becomes **ΘTX** (T is transpose of theta matrix).

**Cost Function for more than 1 variable**

* Cost Function will also change if there are more than 1 variable.
* The new cost function becomes something like below:
  + J (Θ) =1/2m  Summation I from 1-m(Summation J from 1-n(Θj h Θ(x(i)j))– y(i) )2

**Gradient descent for more than 1 variable**

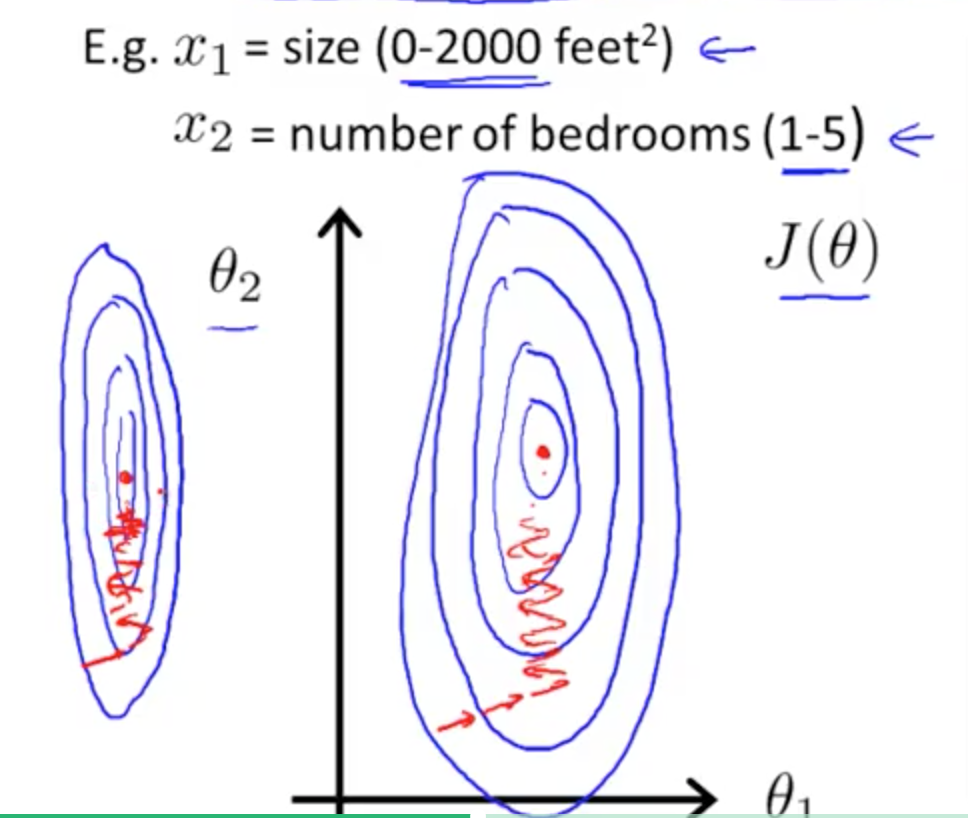
* The gradient descent algorithm also changes, we have already seen scenarios for theta0 and theta1, others will look as shown below:



* X can be denoted as Xj  for the number of variables.

**Multivarient Linear Regression**

* In case of multiple variable we have to make sure the features are on the same scale.
* If they are on different scale the countour plot will be very skinny along one axis and gradient descent will take very longer to minimize the cost.

****

* The way we can avoid this is with the help of scaling down the X1 values.
* We usually scale the features in the range of -1<=xj<=1.
* Sometimes we also use the formula X1 = (X1 – mu1)/S1. Where mu is the average value of X over all m and S1 = max value- min value.
* Feature scaling makes gradient descent run much faster.