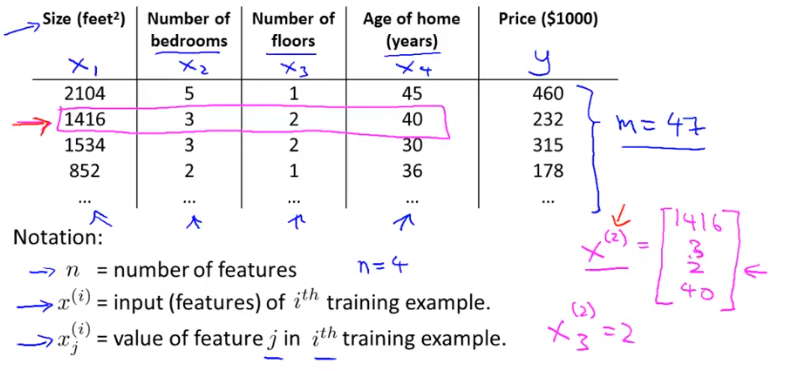
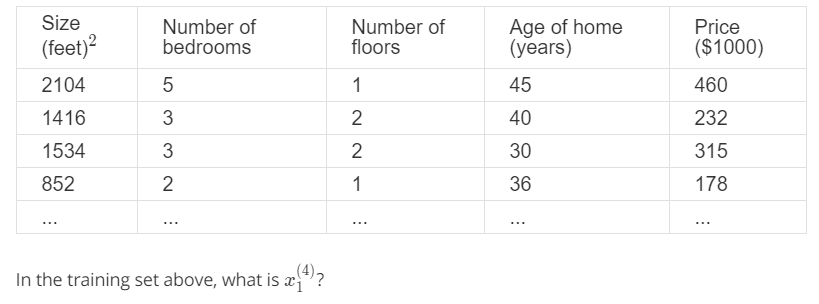
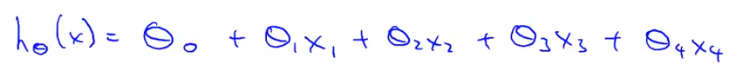
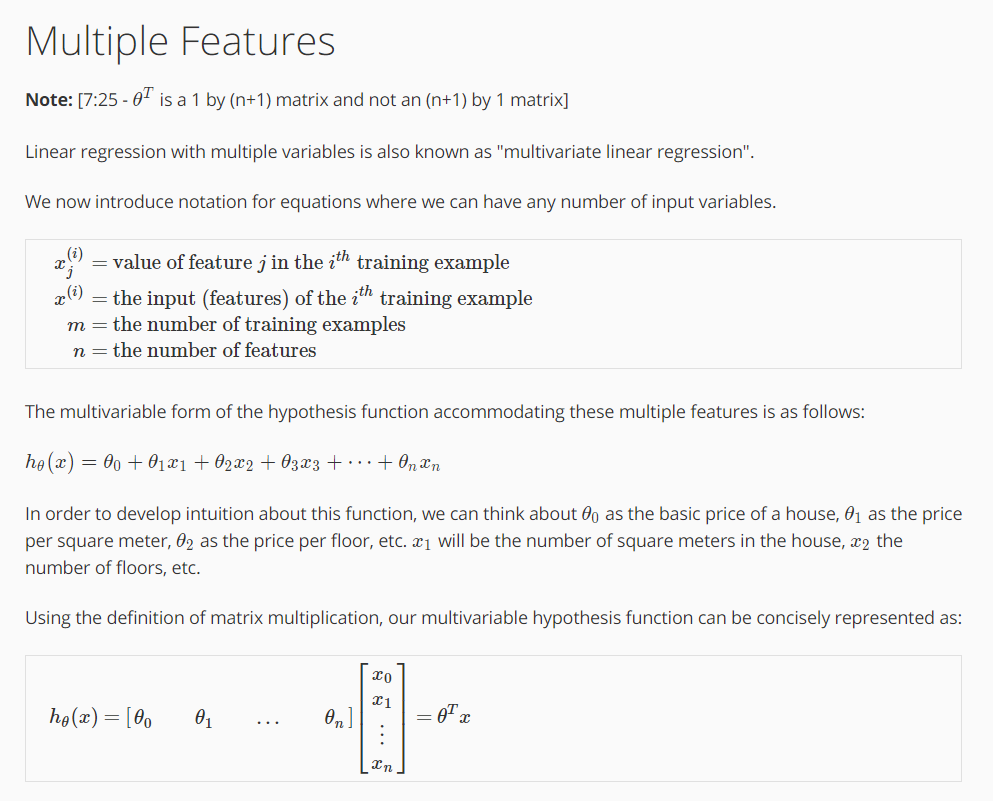
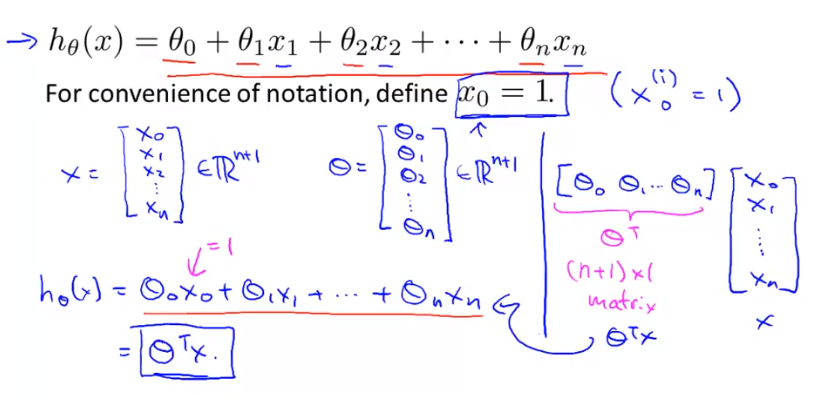
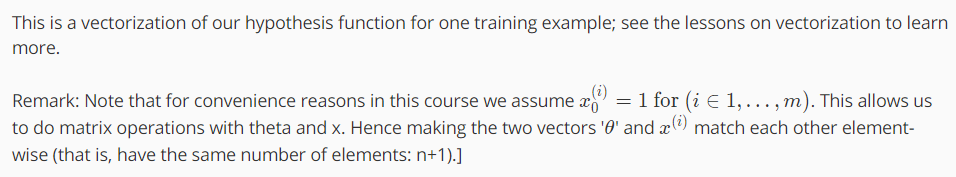
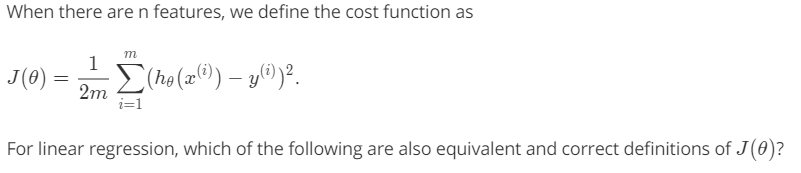
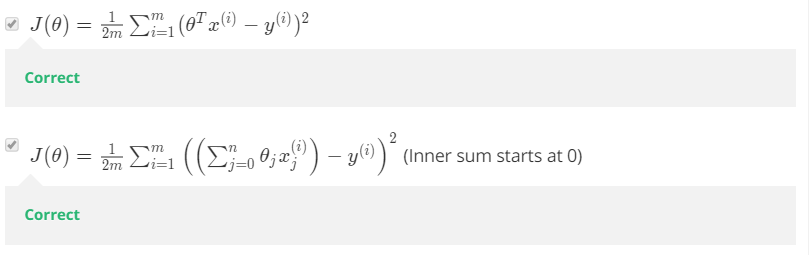
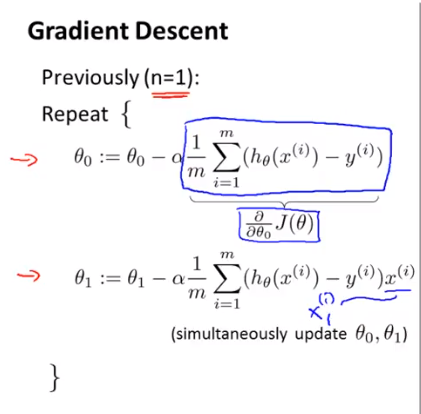
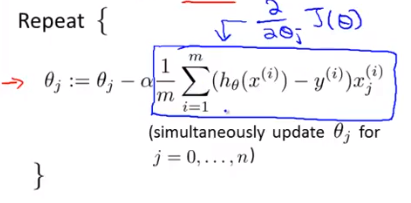
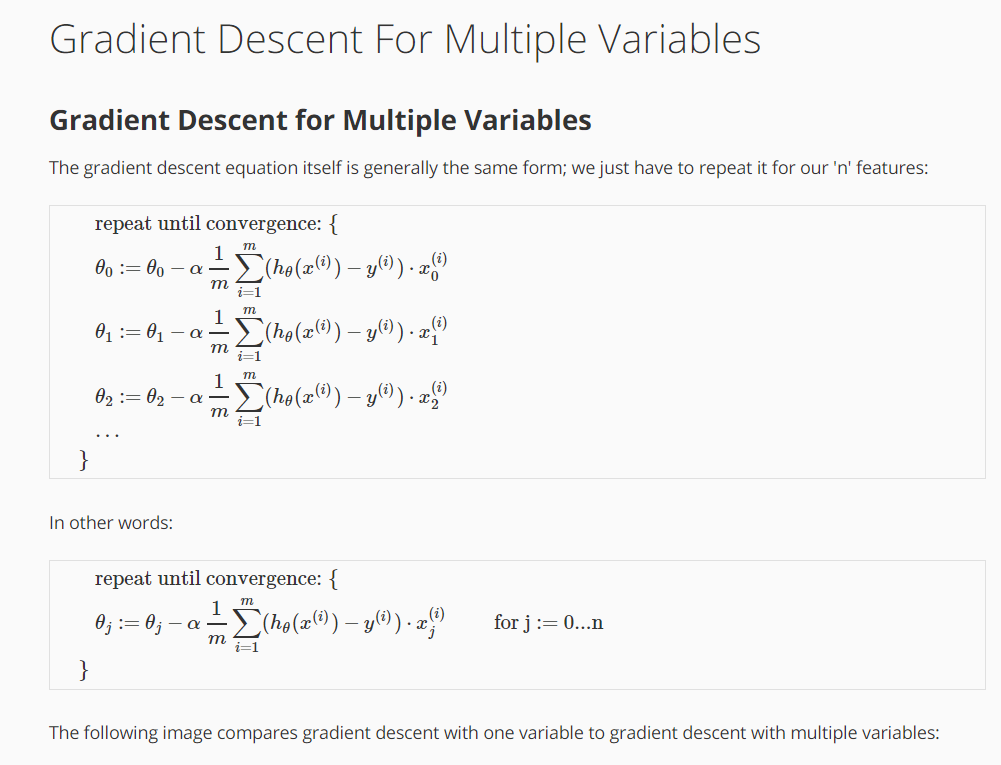
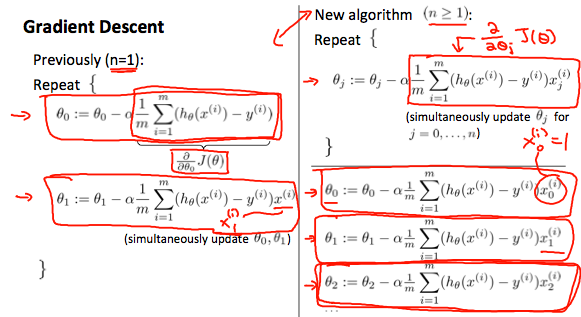
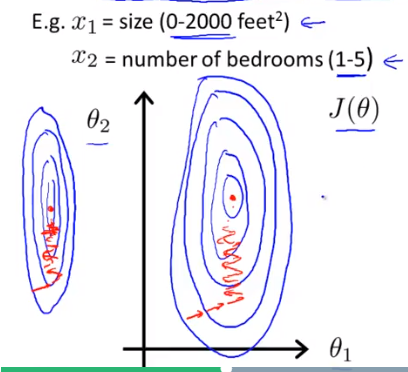
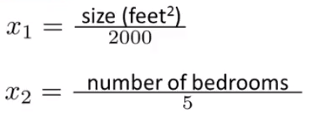
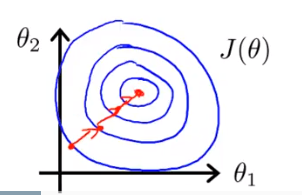
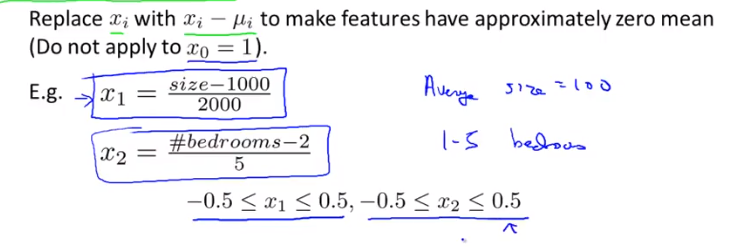
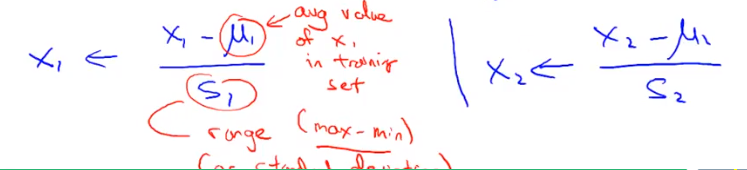
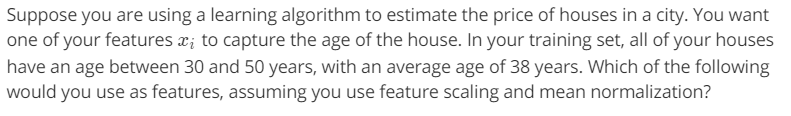
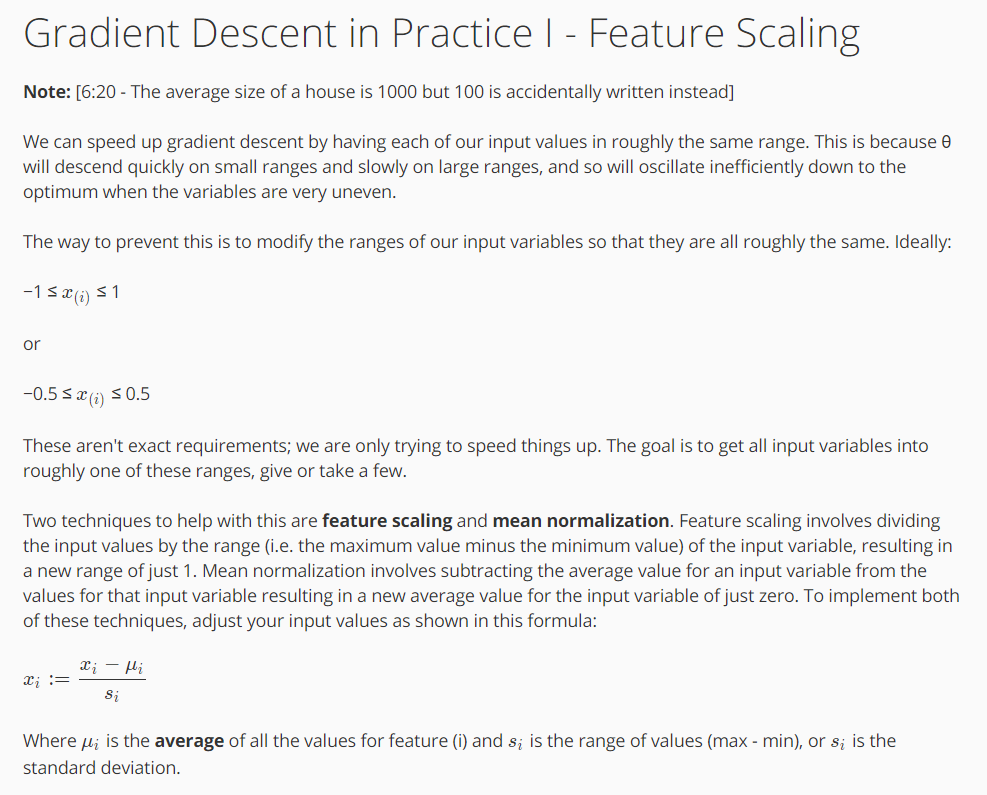
Multivariate Linear Regression (多元线性回归)

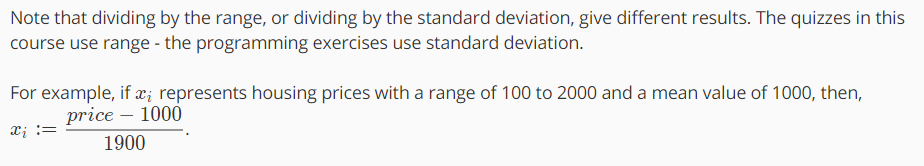
* Definition
  + 在回归分析中，如果有两个或两个以上的自变量，就称为多元回归。
* Multiple Features
* Gradient Descent for Multiple Variables
* Gradient Descent in Practice I – Feature Scaling
* Gradient Descent in Practice II – Learning Rate
* Features & Polynomial Regression
* Multiple Features
* Features = variables.
* Notation:  
    
  m: the number of the total training examples.  
  i: index.  
  i.e.   
    
  exercise:   
    
  answer: 
* Hypothesis / multivariate linear regression:  
  Previous hypothesis for single feature:   
  a hypothesis for four features: (h-theta is the y, price of house in the last notation part. Second, the x1 is equal to 1 in the formula.)  
    
  
* Gradient Descent (梯度下降) for Multiple Variables
* Hypothesis:   
  parameter: θ, a (n+1)-dimensional vector.  
  exercise:  
     
  answer:  
  
* Gradient Descent Algorithm  
  when feature = 1:  
    
    
  when feature >= 1:  
    
    
  same colours are the same formula in different representations:  
    
    
    
  
* Gradient Descent in Practice I – Feature Scaling
* Feature scaling  
  idea: make sure features are on a similar scale. If different features have different ranges, it would be slower the process to find the right weight for the final value. Like the pictures below:  
    
    
  thus, make the features into , the shape would be like 
* Rules of feature scaling  
    
  also, don’t make the range too small. Like, 0.0001.  
  generally, the range could be (-3, 3). (-1/3, 1/3).  
  each feature does not have to be in the same range. At least they need to close to each other.
* Another way to scale feature: Mean normalization  
    
  more general rule:  
  s1 is the range of the value of that feature.

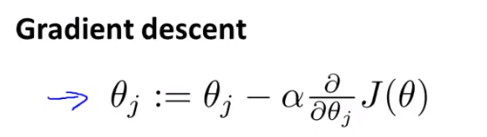
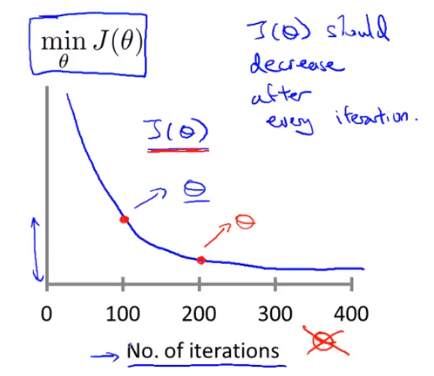
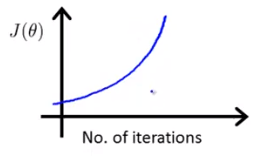
µ1 is the average value of x1 in training set.  


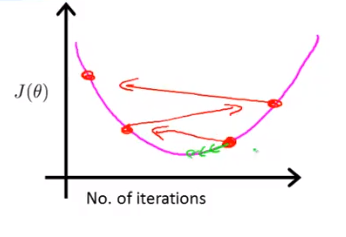
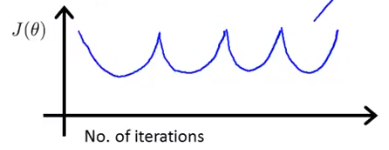
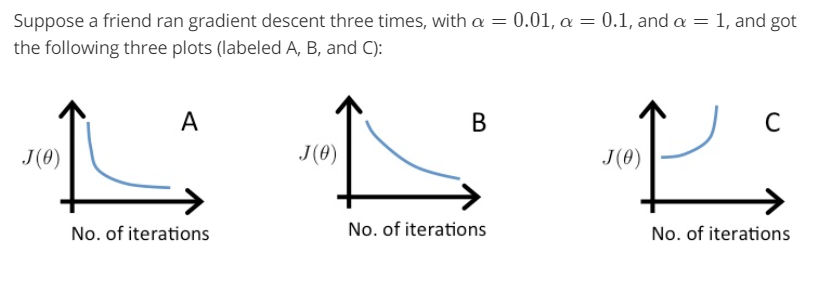
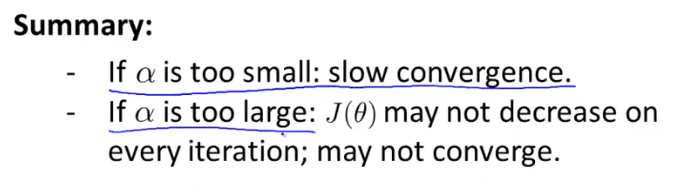
* Exercise

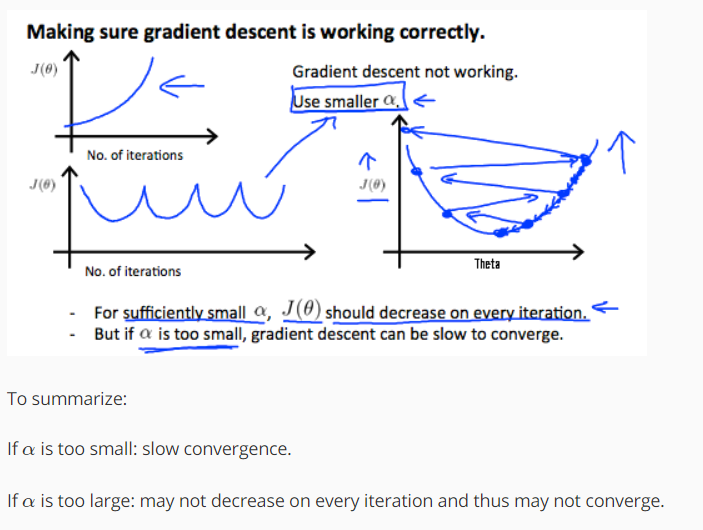
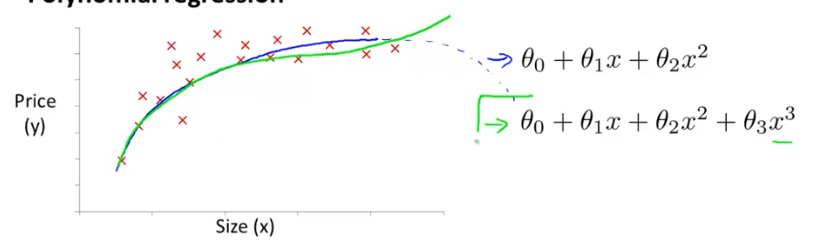
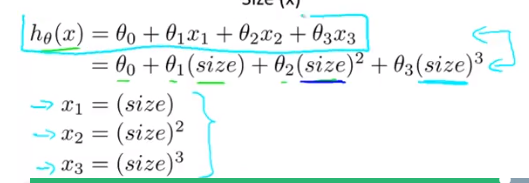
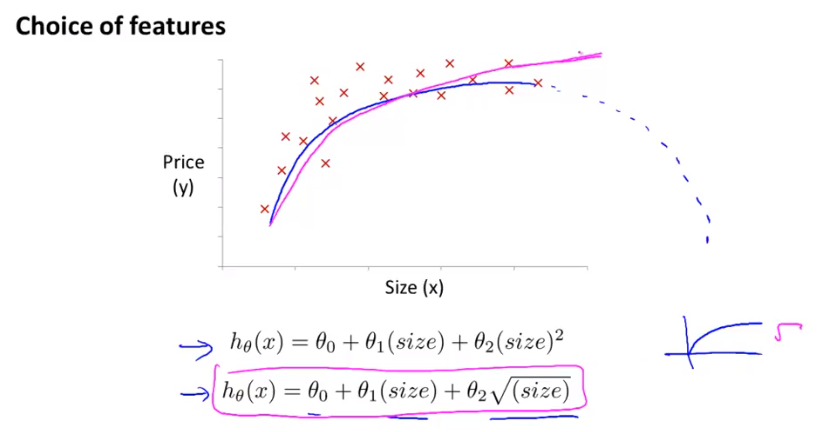
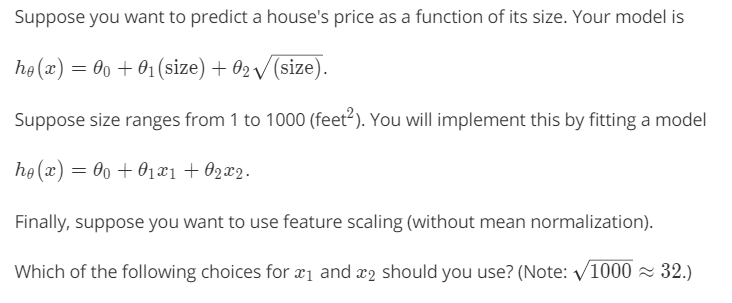
  
answer: 



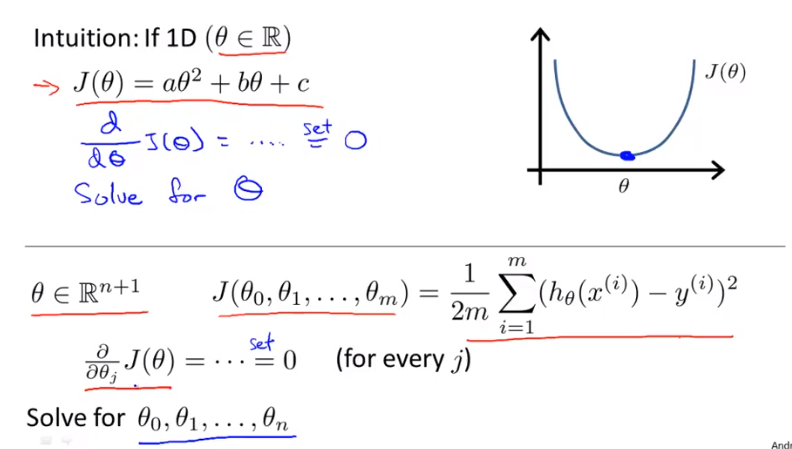
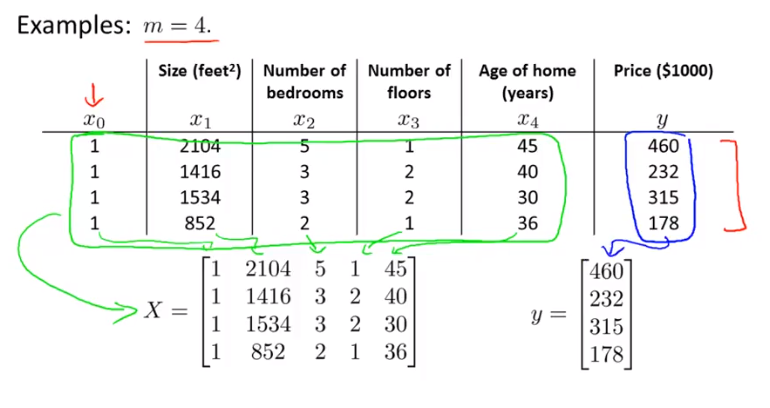
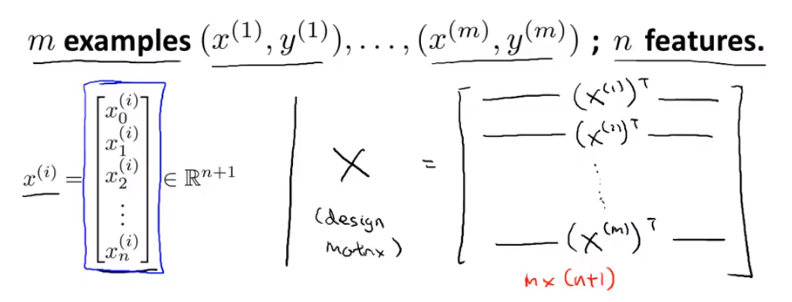
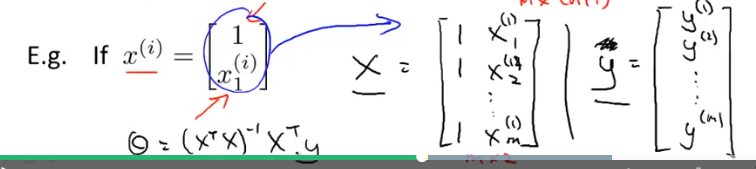
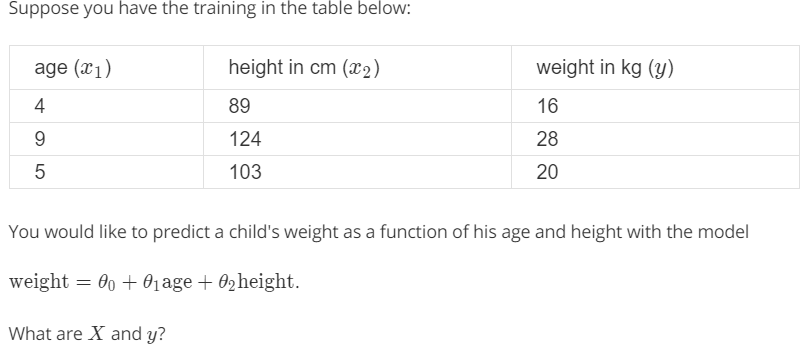
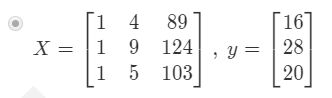
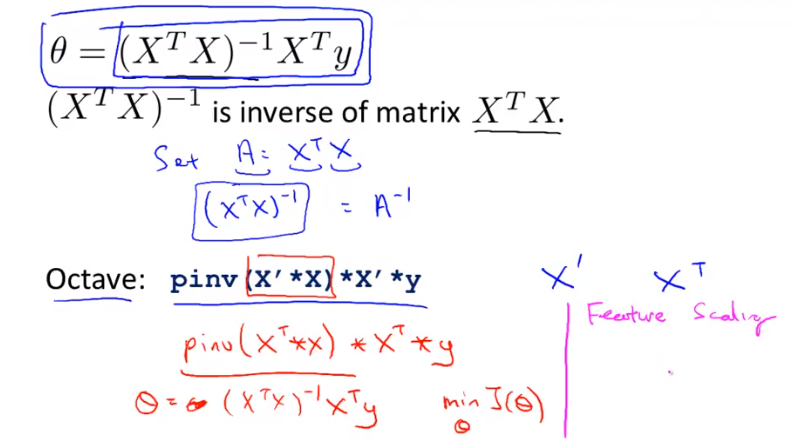
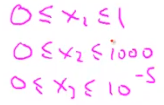
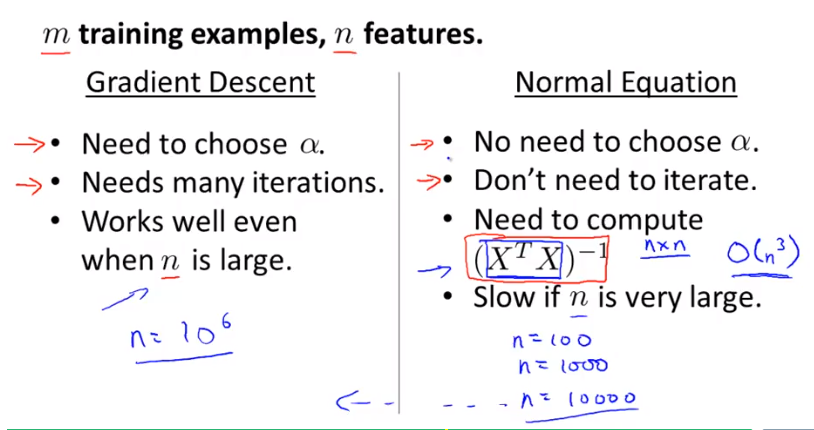


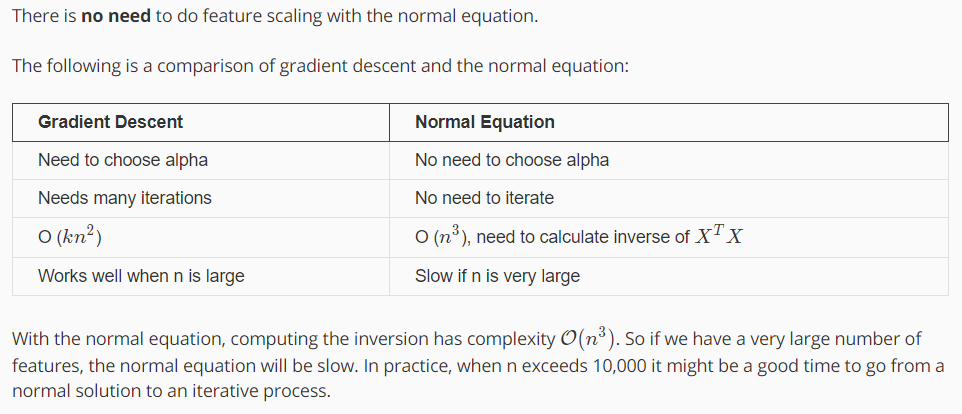
* Gradient Descent in Practice II – Learning Rate  
  
* Debugging: how to make sure gradient descent is working correctly.  
  the right result: J(θ) should decrease after every iteration. Like below:  
    
    
    
  another example: when you the diagram as below, you know the gradient descent isn’t working:  
  

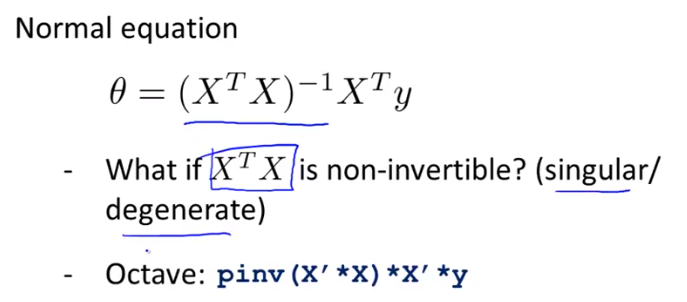
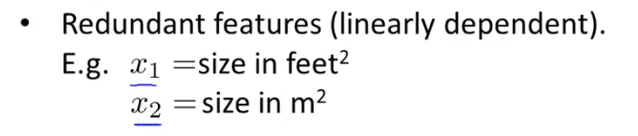
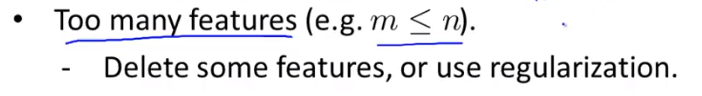
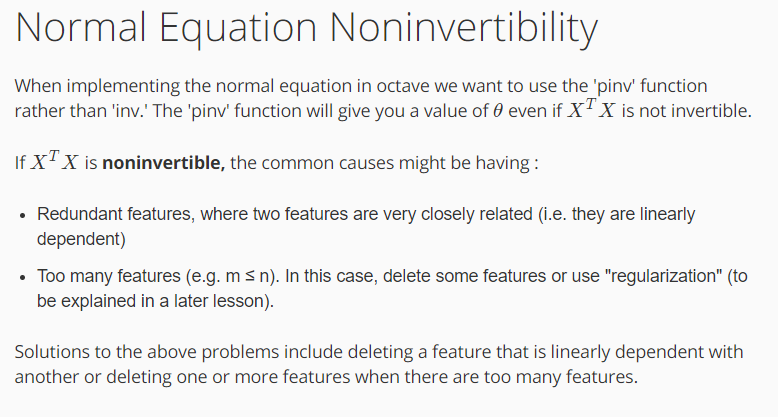
Need to use smaller α if you see below:  
or   
  
  
  
exercise:  
  
answer:   
  
  
but in some cases, if α is too large, slow converge also possible.

* How to choose learning rate α  
  …, 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1, … the range between two numbers is around 3 times.  
    
    
  
* Features & Polynomial Regression (多项式回归)  
    
    
    
    
  - Choice of features  
     
    
  exercise:  
    
  answer:   
    
  

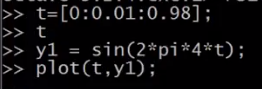
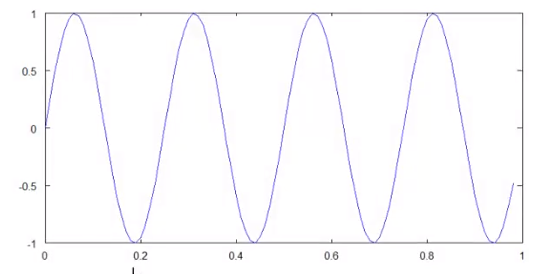
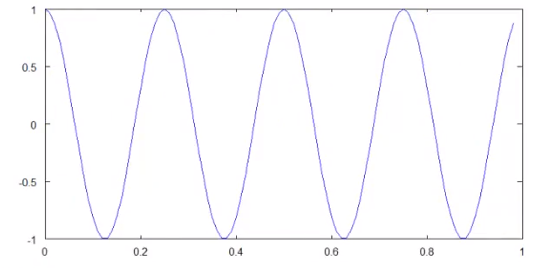
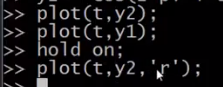
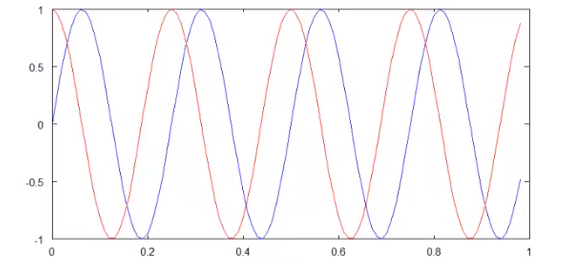
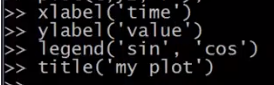
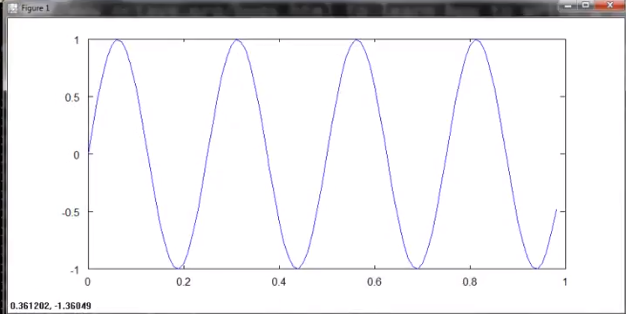
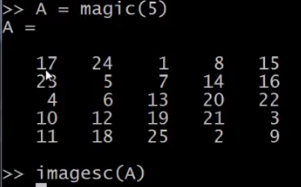
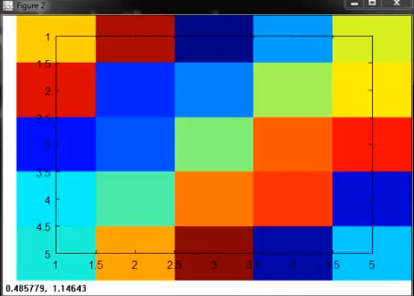
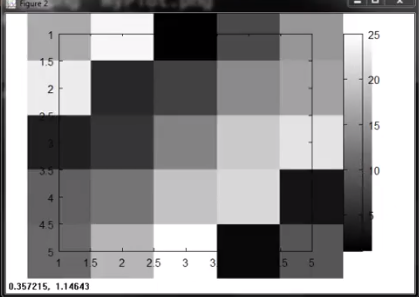
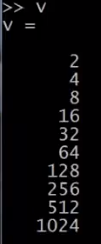
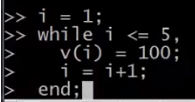
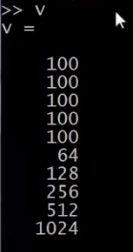
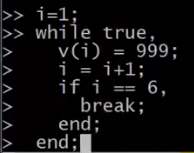
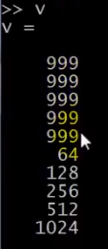
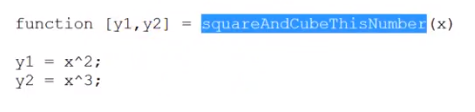
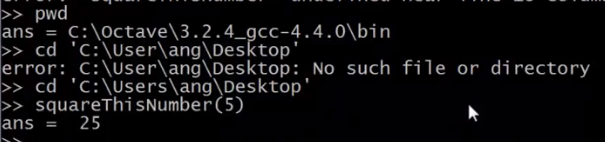
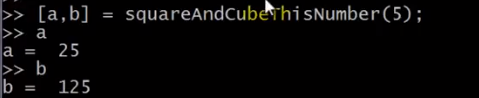
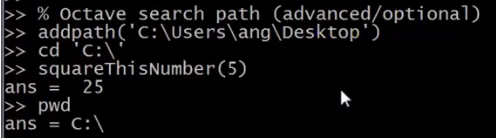
Computing Parameters Analytically

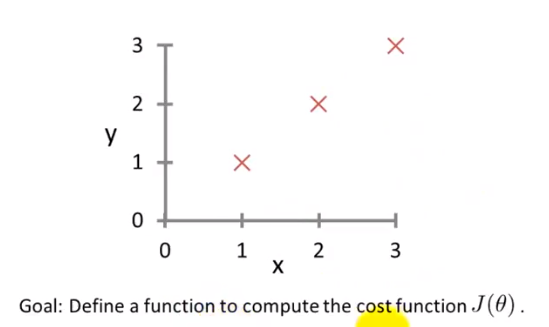
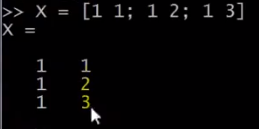
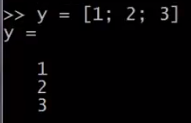
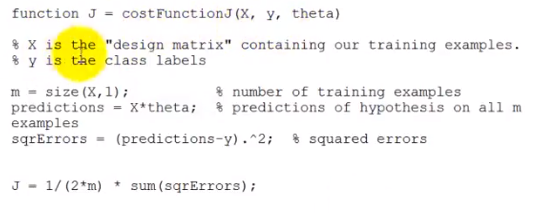
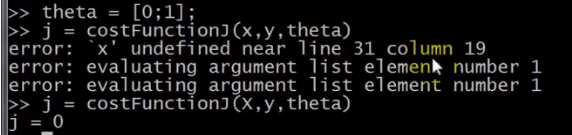
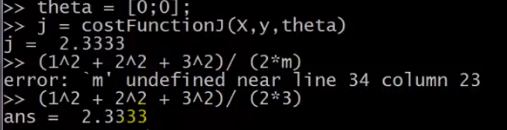
* Normal Equation: method to solve for θ analytically.  
    
  - example:  
    
  so, X, the design matrix, is a m by (n+1) matrix. Y is a m-dimensional vector. m is the number of training examples. n is the number of features.  
  🡨 give you the θ that minimize your cost function.
* In more general case:  
    
    
    
  exercise:  
    
  answer: 
* More about the formula:   
  pinv / inv: is a function in Octave to calculate inverse even though XTX is non-invertible. 
* Notice  
  if you are using normal equation, you don’t need to do feature scaling if your features are:  
    
  However, if you are using gradient descent, you need to do feature scaling if features are looking like above.
* When should use gradient descent? When should use normal equation?  
  Some advantages and disadvantages are below:  
    
    
  the cost of converting a matrix is: O(n3).  
  Thus, if n is very large, we may choose to use gradient descent.

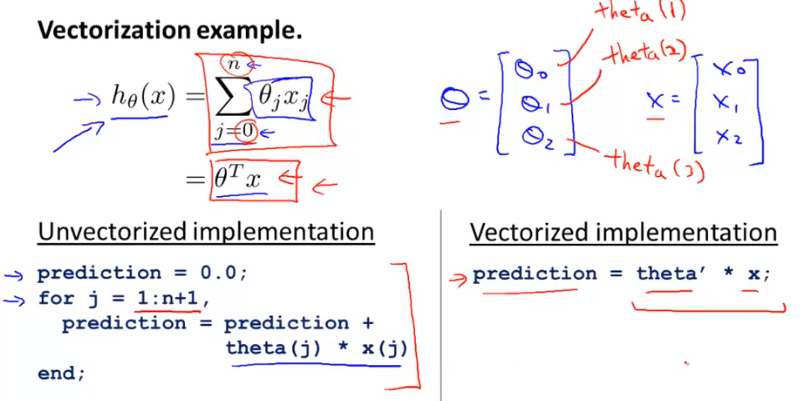
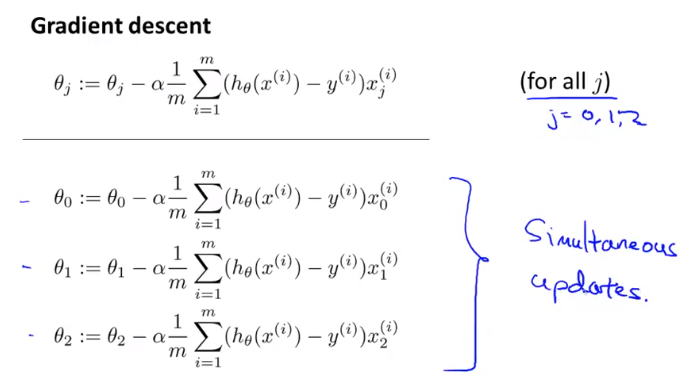
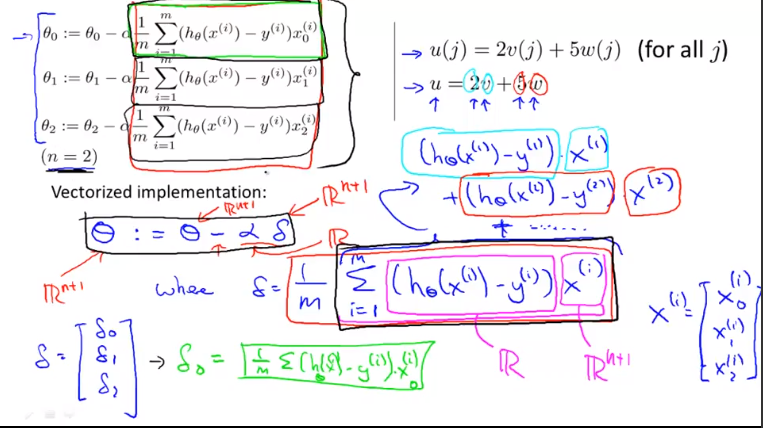
  
  


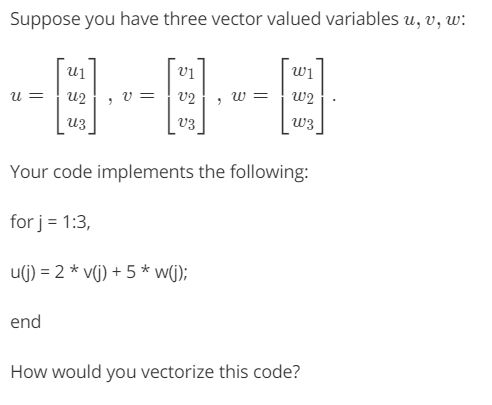
* Normal Equation Non invertibility (不可逆性)
* What if XTX is non-invertible? This should happen very rarely.  
    
  the situations that XTX is non-invertible:  
  1.   
  so, x1 is always equal to 3.28x2, therefore, XTX is non-invertible. To solve this problem, just delete one feature.  
  2.  
    e.g. too many features (n) and less training samples (m).  
    
  

Octave Tutorial

* Plotting Data
* Example  
    
    
    
    
    
  Then, how to combine them together?  
  . ‘r’ stands for red colour.  
    
    
    
    
  save the graph:  
    
  >> help plot %help you save the image into other form instead of png.  
    
  >> close % close the figure window  
    
    
    
    
    
    
  
* 
* >> clf; % clear the figure.
* Another example  
    
   🡨 each area represents a different value.  
  or:  
    
  
* Comma chaining of function calls: a good way to put a lot of commands on the same line.  
  >> a=1, b=2, c=3  
  print out a,b,c  
  >> a=1; b=2; c=3;  
  doesn’t print out a,b,c
* Control Statements: for, while, if statement
* Example 1  
  >> v = zeros(10,1)  
  >> for i=1:10,  
  >> v(i) = 2^I;  
  >> end;  
  result:   
     
    
  another way:  
  >> indices = 1:10;  
  >> for i = indices,  
  > disp(i);  
  > end;  
  result:  
   
* Example 2  
    
  result:  
   
* Example 3  
    
    
  result:  
   
* Example 4  
  
* Function Define
* Example  
  store the function in a .m file with the name of that function.  
  🡨 this function return y, and take in x.  
  🡨 return multiple values.  
    
  then, call it in Octave:  
  first, Octave needs to know where to find the file. Go to the path where you store the file by using pwd and cd. Then, just call that function.  
    
    
    
    
  add path:  
  

Example 2  
  
input x matrix:  
  
  
  
  
  
  
  
  
  
  
typing ctrl-c will stop the program in the middle of its run.

* Vectorization (矢量)
* Example  
    
    
  \*In matlab, all index starts from 1.  
    
    
    
  

Exercise  
  
answer:   
  
\* rule of multiplying matrix: 左乘矩阵的列数 = 右乘矩阵的行数

* 3
* 4
* 5