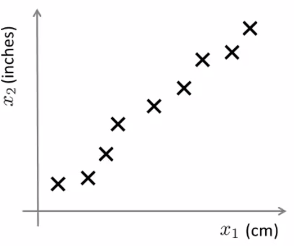
***Dimensionality Reduction***

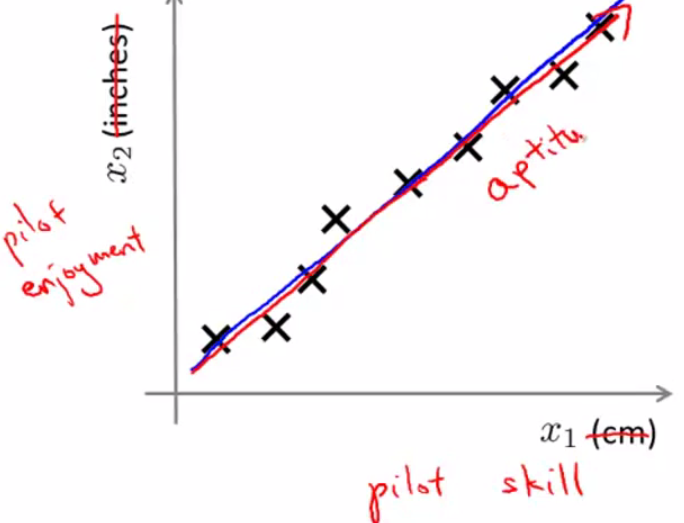
***Motivation***

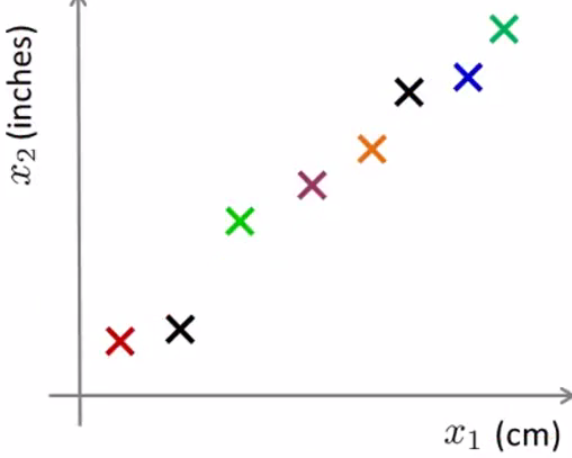
1. **MOTIVATION I: DATA COMPRESSION**

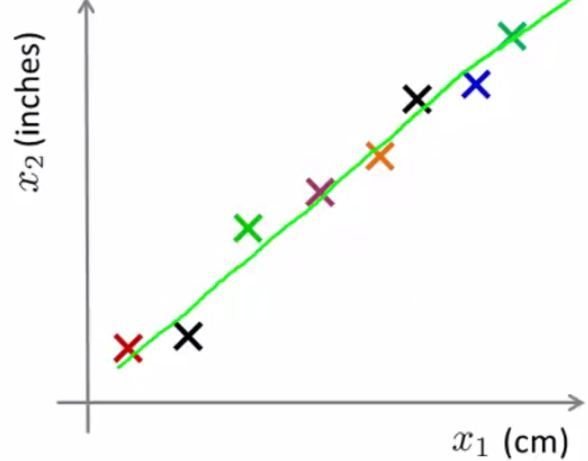
* There are a couple of different reasons why one might want to perform a 2nd type of unsupervised learning problem, **dimensionality reduction,** such as data compression
* **Data compression** not only allows us to compress data (+therefore have it use up less memory or disk space), but also allows us to speed up learning algorithms.
* Let's say that we've collected a data set w/ many features + we've plotted just 2 of them:



* Unknown to us, the 2 features were actually the length of something in both centimeters + in inches
* This gives us a highly redundant representation
* Maybe instead of having 2 separate features x1 + x2, both of which measure length, we want to **reduce** the data to 1-dimension + have just 1 feature measuring this length.
* In case this example seems a bit contrived, this “cm + in” example is actually not that unrealistic + not that different from things happening in industry.
* If you have hundreds or thousands of features, it is often easy to lose track of exactly what features you have.
* Sometimes may have a few different engineering teams who give you features
* Maybe 1 engineering team gives you 200 features, a 2nd gives you another 300, + a 3rd gives you 500, so you have a 1K features all together
* It would become hard to keep track of exactly which features you got from which team, + it's actually not that unusual to end up w/ highly redundant features like these.
* If the length in cm were rounded off to the nearest cm + length in in. was rounded off to the nearest in., that explains why these examples don't lie perfectly on a straight line (b/c of round-off error)
* We can + want to reduce the data to 1 dimension instead of 2 dimensions to reduce the redundancy
* Ex: Working w/ pilots that fly helicopters + you do a survey/test of different pilots + end up 1 feature, x1 = the *skill* of these helicopter pilots, + maybe x2 could be pilot enjoyment.
* Maybe these 2 features will be highly correlated + what you really care about might be this direction of the relationship 🡺 a different feature that really measures pilot aptitude.

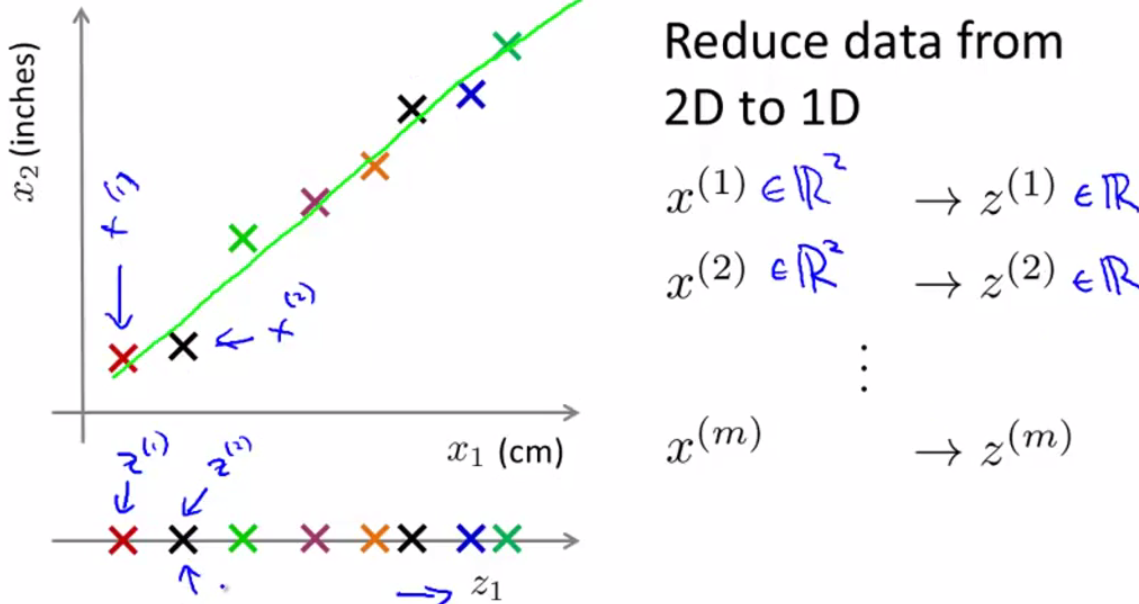


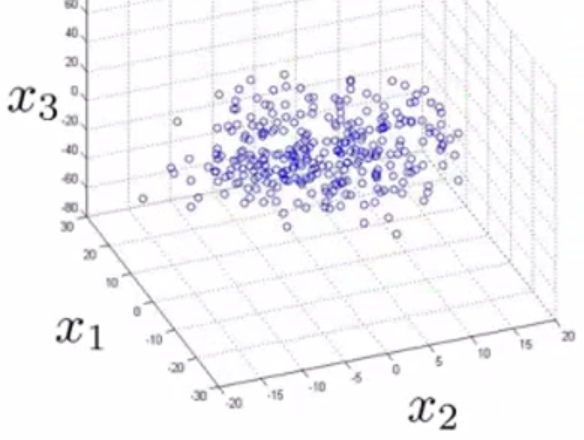
* So if you highly correlated features, maybe you really want to reduce the dimension.
* What does it really mean to **reduce** the dimensions of the data from 2 dimensions down to 1 dimension?
* Color in examples using different colors.
* 
* By reducing dimensionality, I would like to find maybe a line/direction on which most of the data seems to lie + project all the data onto that line
* By doing so, what I can do is measure the position of each of the examples on that line + come up w/ a new feature, z1, to specify the position on the line

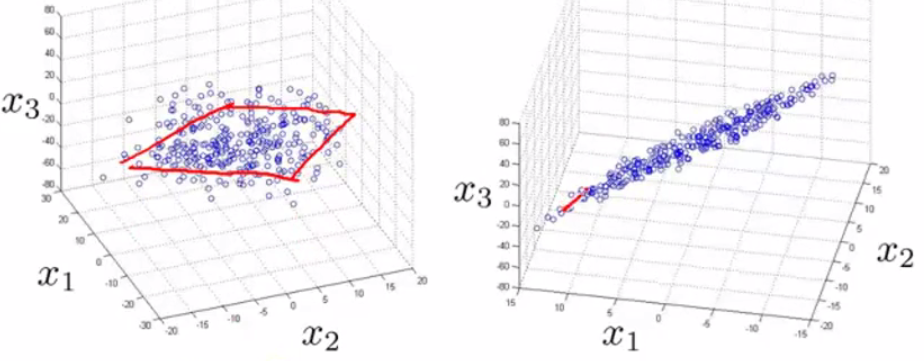




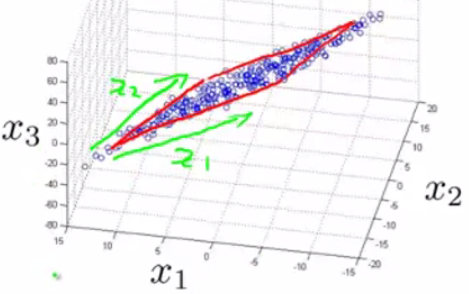
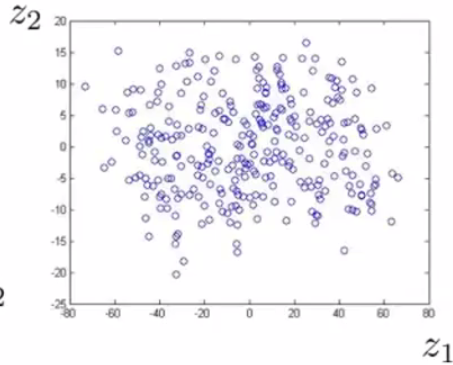
* So we end up w/ only 1 number, z1 = a *new feature* that specifies the location of each of those points on this new line.
* And what this means is that whereas previously, if we had an example x1, in order to represent the original x1, I needed a 2D feature vector.
* Instead now I can represent it w/ z1 w/ a real number



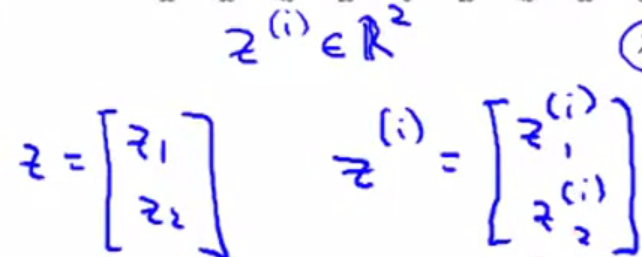
* To summarize, if we allow ourselves to approximate the original data set by projecting all the original examples onto the green line, then I need only 1 real number to specify the position of a point on the line
* Therefore, what I can do is use this 1 number to represent the location of each training example after they've been projected onto that line.
* This is an *approximation* to the original training set b/c I’ve projected all training examples onto a line, but now, I keep only 1 number for each example + this halves the memory requirement for storing my data.
* Perhaps more interestingly + more importantly, is that this will allow us to make learning algorithms run more quickly
* That is perhaps the more interesting application of this data compression, rather than reducing memory/disk space requirements for storing data.
* In a more-typical example of dimensionality reduction, we might have 1K-dimensional (1000D) data that we might want to reduce to let's say 100-dimensional or 100D
* Ex: Reducing data from three dimensional 3D to two dimensional 2D.
* 
* We have a set of examples x(i) which are points in ℝ ^3 (3 dimensions)
* All of this data lies roughly on the plane + what we can do w/ dimensionality reduction is take all of that data + project it down onto a 2D plane.

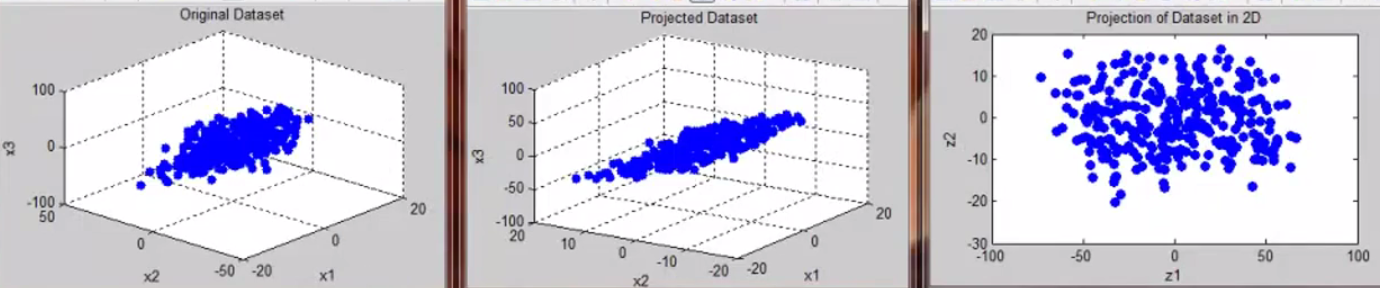


* Now, finally, in order to specify the location of a point w/in this new plane, we need 2 numbers to specify the location of a point along the 2 new axes axis, z1 + z2

* We can now represent each training example using 2 numbers, z1 + z2.
* From this, data can be represented using vector z in ℝ ^2 (2D vector where if I have some particular example z(i), it is represented by a 2D vector, z(i)1, z(i)2.



* Now, let’s make sure these figures make sense + reshow these exact 3 figures again but w/ 3D plots.
* 

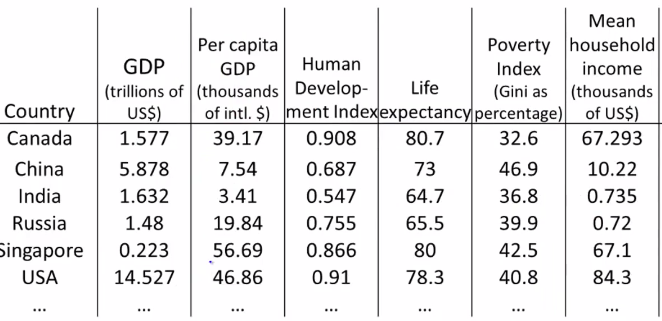




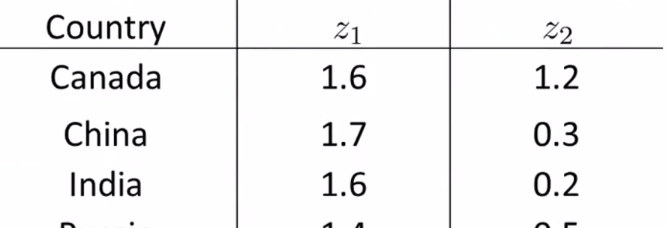


**II. MOTIVATION II: VISUALIZATION**

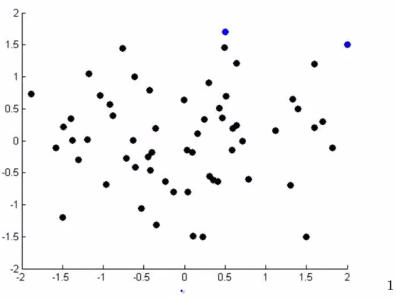
* Dimensionality reduction can also help to visualize data.
* For a lot of ML applications, it really helps to develop effective learning algorithms if we can understand our data better + if there is some way of visualizing the data better
* Dimensionality often reduction offers a useful tool to do so



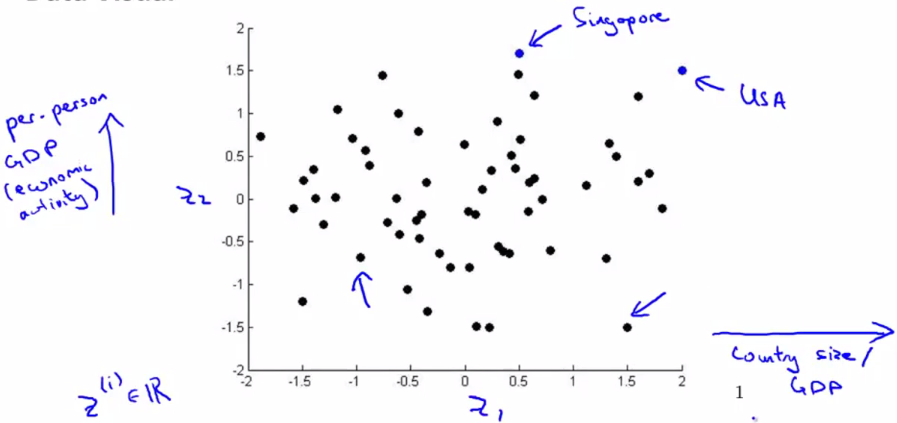
* Say we've collected a large data set of many statistics + facts about different countries around the world w/ maybe 50 features for every country w/ a huge set of countries.
* So is there something we can do to try to understand our data better?
* It's very difficult to plot 50-dimensional data. What is a good way to examine this data?
* Using dimensionality reduction, instead of having each country represented by a 50D feature vector, x(i), we can come up w/ a different feature representation w/ z vectors in ℝ ^2



* If that's the case (we just have a pair of numbers, z1 + z2 to summarizes my 50 features), we can plot these countries in ℝ ^2 + use that to try to understand the space of these features of different countries a bit better
* We reduce the data from 50D to 2D + plot this as a 2 dimensional plot
* If you look at the output of the Dimensionality Reduction algorithms, It usually doesn't ascribe a physical meaning to these new features z1 and z2
* It's often up to us to figure out roughly what these features means.



* Here, every country is represented by a point z(i), an R2
* For each of these, you might find, for example, That the horizontal/z1 axis corresponds roughly to the overall economic activity (GDP) of a country
* Whereas the vertical axis, z2, might correspond to the GDP/economic activity per person
* You might find that, given our 50 features, these are the 2 main dimensions of deviation



* You might find that the axes z1 + z2 can help most succinctly capture the 2 main dimensions of the variations amongst different countries.

