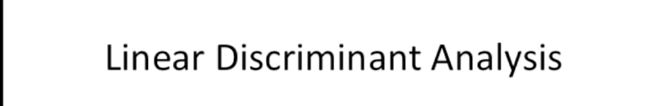
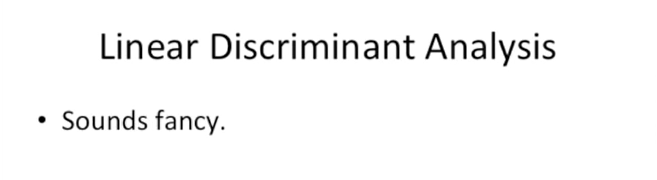
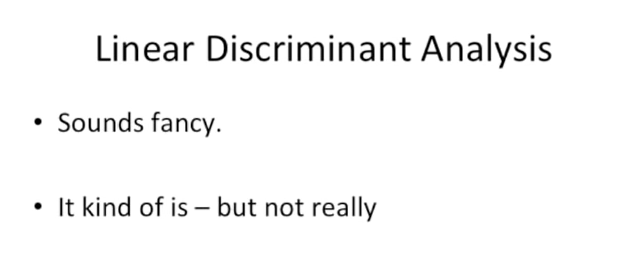
<https://www.youtube.com/watch?v=azXCzI57Yfc&list=PLblh5JKOoLUICTaGLRoHQDuF_7q2GfuJF&index=29>



Today we're going to be talking about linear discriminant analysis.

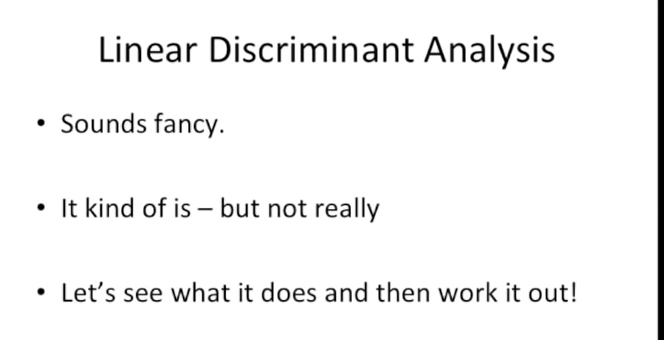


Which let's be honest sounds really fancy.



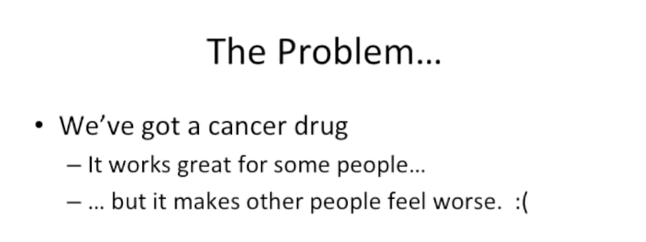
And it kind of is but not really.

I think we can understand it.



Let's see what it does and then we'll work it out !

That is let's look at some examples of why we might need linear discriminant analysis and then we'll talk about the details of how it works.

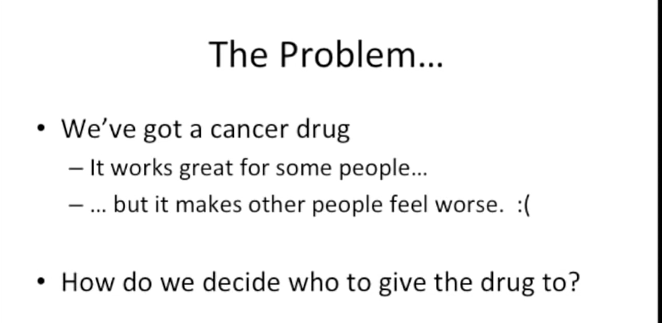


Imagine that we have this cancer drug

and that cancer drug works great for some people

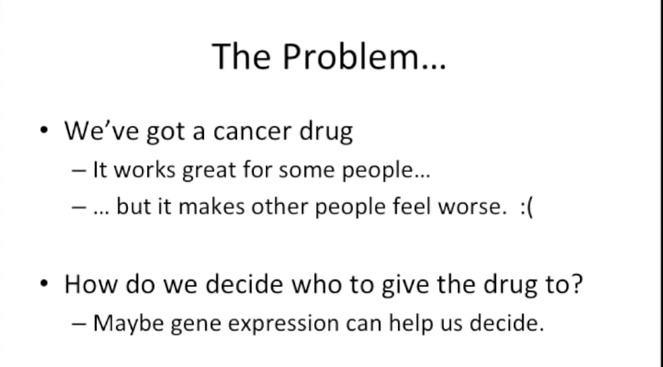
but for other people it just makes them feel worse.

Wha-wha.



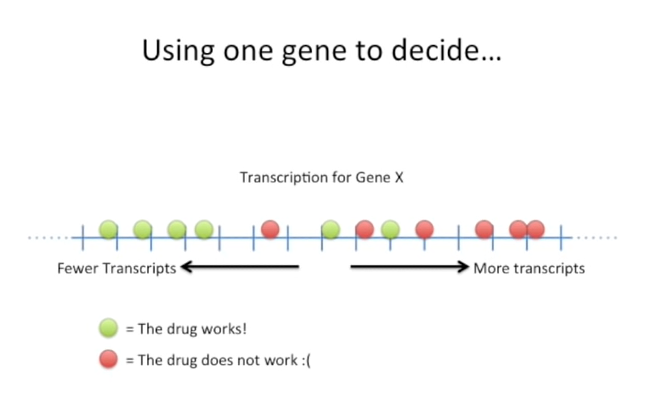
We want to figure out who to give the drug to.

We want to give it to people who it's going to help but we don't want to give it to people that it might harm.



And since I'm a geneticist and I work in a genetics department the way I answer all my questions is to look at gene expression.

Maybe gene expression can help us decide.



Here's an example using one gene to decide who gets the drug and who doesn't.

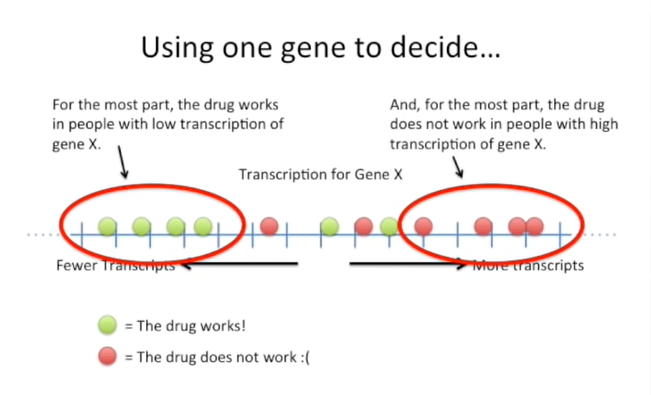
We've got a number line, and on the left side we've got fewer transcripts, and on the right side we've got more transcripts the dots represent individual people.

The green dots are people who the drug works for.

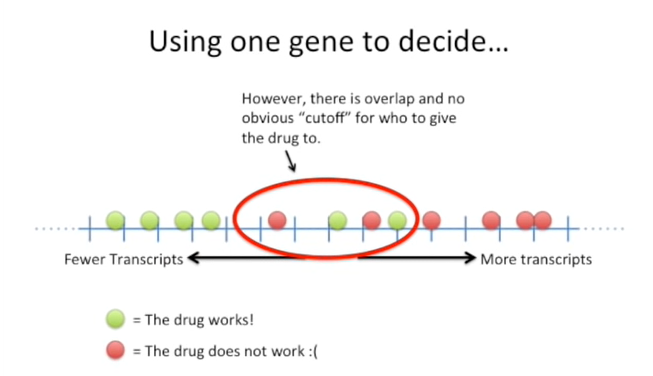
The red dots represent people whom the drug just makes them feel worse.



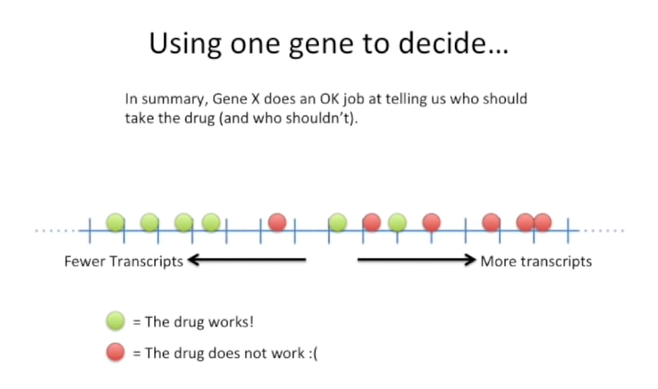
We can see that for the most part the drug works for people with low transcription of gene X.



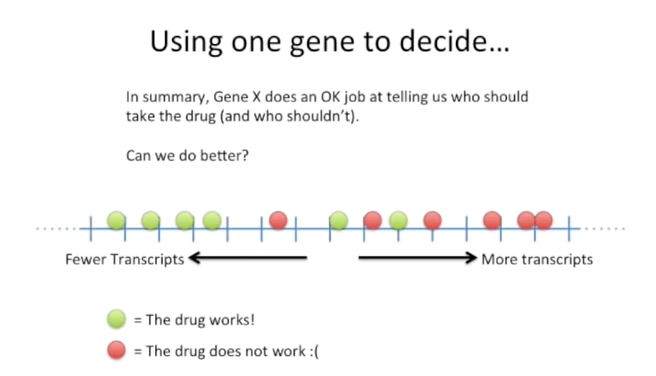
And for the most part the drug does not work for people with high transcription of gene X.



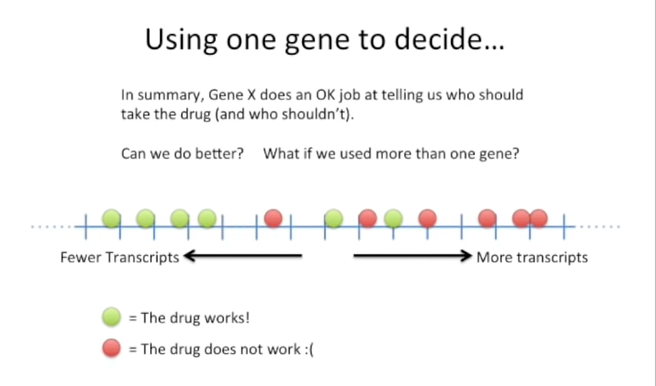
In the middle we see that there's overlap and that the no obvious cutoff for who to give the drug to.



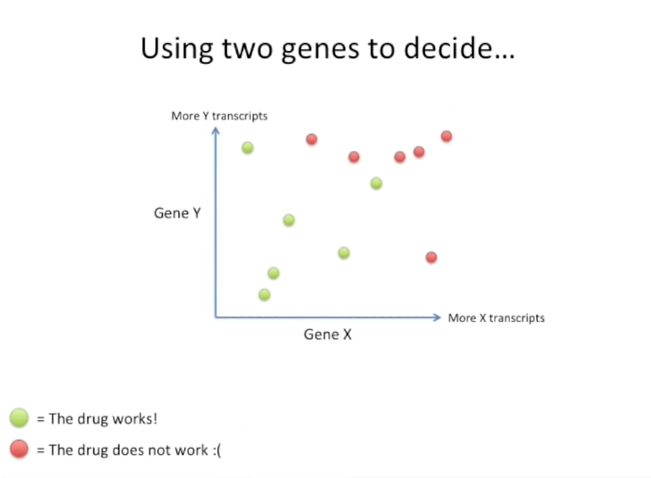
In summary gene X does an okay job at telling us who should take the drug and who shouldn't.



Can we do better ?



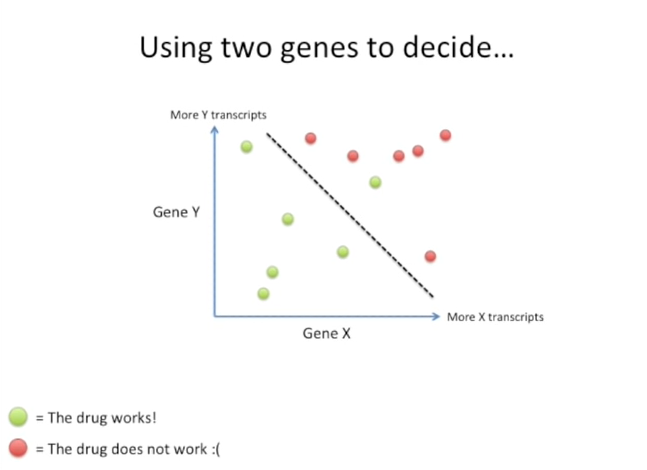
What if we used more than one gene to make a decision ?



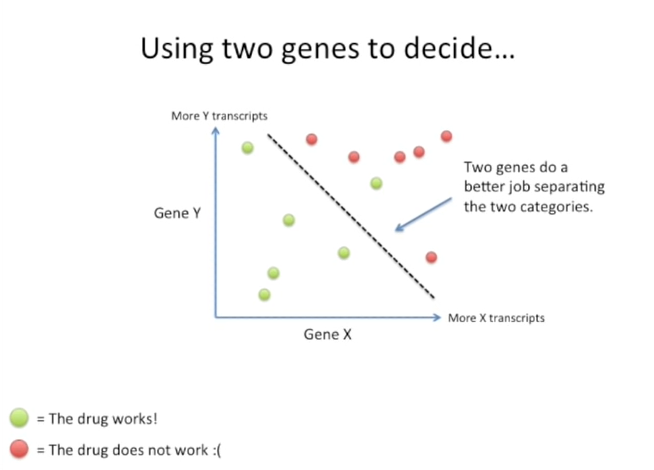
Here's an example of using two genes to decide who gets the drug and who doesn't.

On the x-axis we have gene X

and on the y-axis we have gene Y.

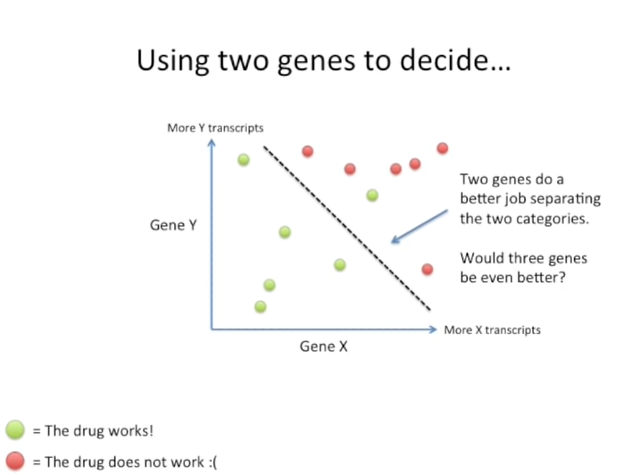


Now that we have two genes we can draw a line that separates the two categories : the green where the drug works and the red where the drug doesn't work.

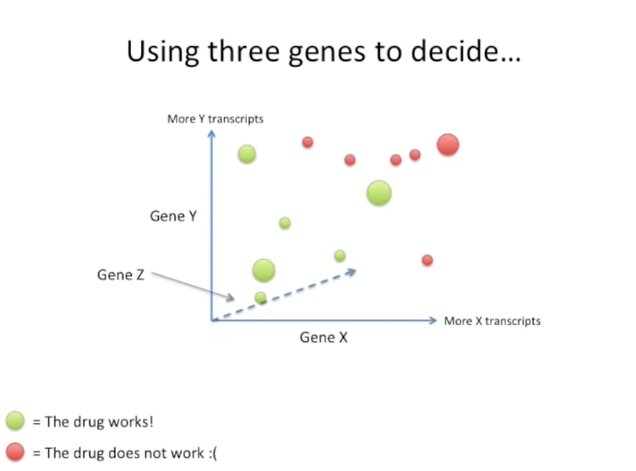


And we can see that using two genes does a better job separating the two categories than just using one gene.

However it's not perfect.



With using three genes be even better ?



Here I've got an example where we're trying to use three genes to decide who gets the drug and who doesn't.

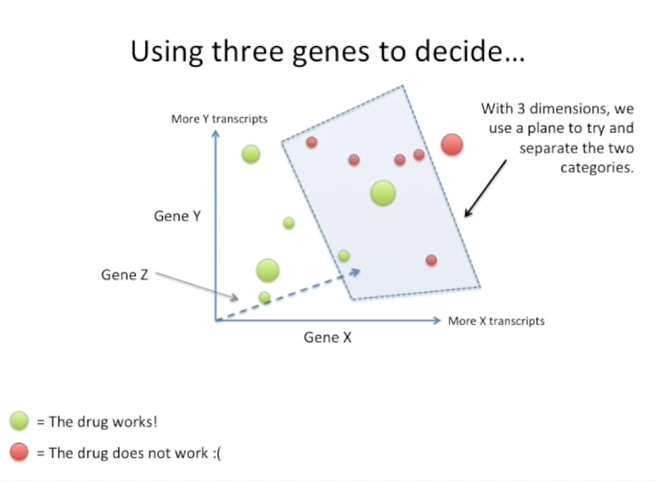
Gene Z is on the z axis which represents depth.

So imagine a line going through your computer screen and into the wall behind it.

And the big circles or the big samples are the ones that are closer to you

and the smaller circles smaller samples are the ones that are further away.

And those are along the z axis.

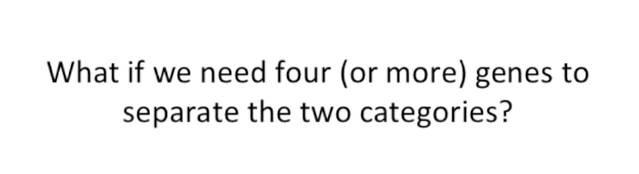


When we have three dimensions we use a plane to try to separate the two categories.

Now I'll be honest I drew this picture but even for me it's hard to tell if this plane separates the two categories correctly.

It's hard for us to visualize three dimensions on a flag computer screen.

We need to be able to rotate the figure and look at it from different angles to really know and that's tedious.



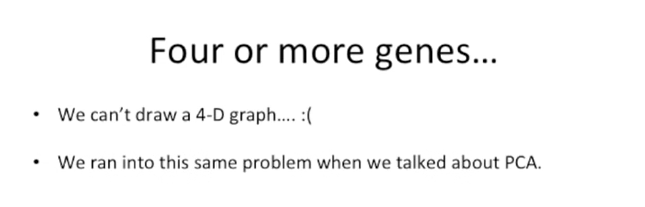
What if we need four or more genes to separate two categories ?



Well the first problem is we can't draw a four dimensional graph or a 10,000 dimensional graph, we just can't draw it.

That's a bummer.

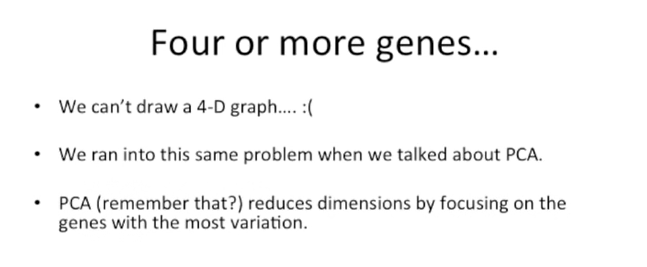
Wha-wha.



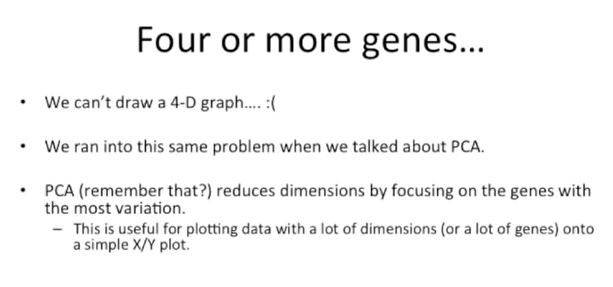
We ran into the same problem when we talked about principal component analysis or PCA.

And if you don't know about principal component analysis be sure to check out the stat quest on that subject.

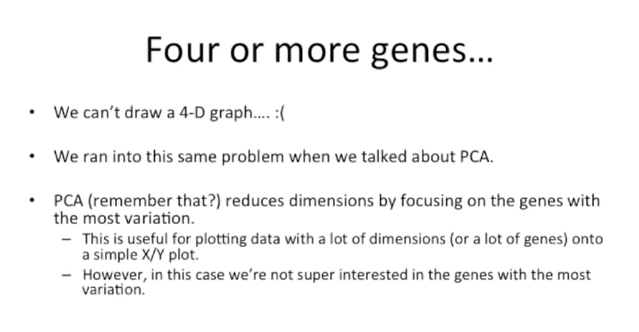
It's got a lot of likes and it's helped a lot of people understand how it works and what it does.



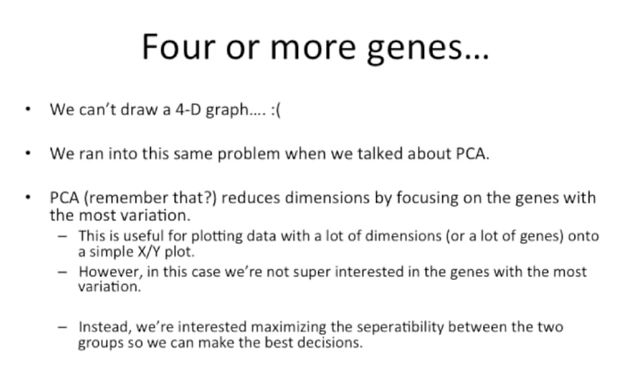
PCA if you can remember about it reduces dimensions by focusing on genes with the most variation.



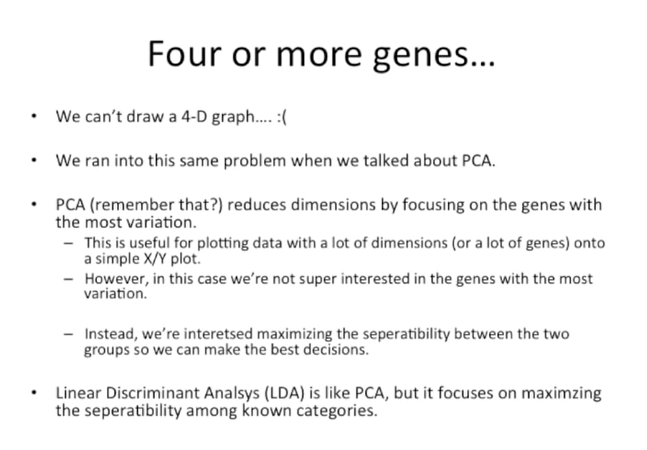
This is incredibly useful when you're plotting data with a lot of dimensions or a lot of genes onto a simple XY plot.



However in this case we're not super interested in the genes with the most variation.

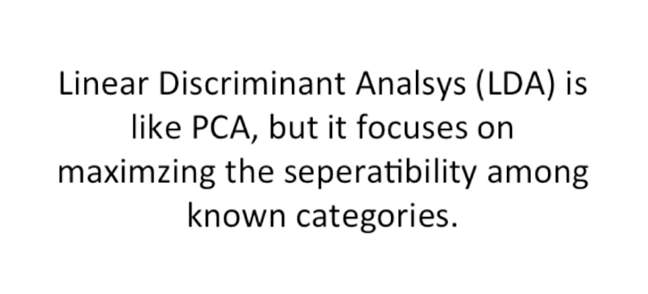


Instead we're interested in maximizing the separable 'ti between the two groups so that we can make the best decisions.

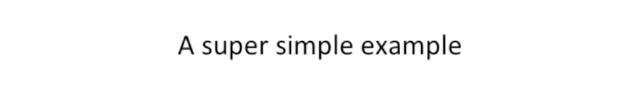


Linear discriminant analysis LDA is like PCA it reduces dimensions.

However it focuses on maximizing the separability among the categories.



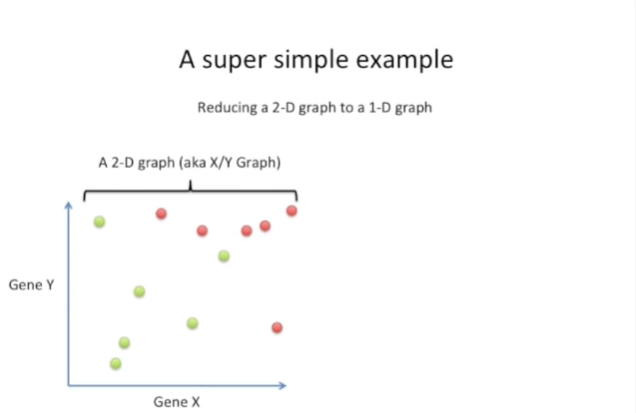
Let's repeat that to emphasize the point : linear discriminant analysis LDA is like PCA but it focuses on maximizing the separability among the known categories.



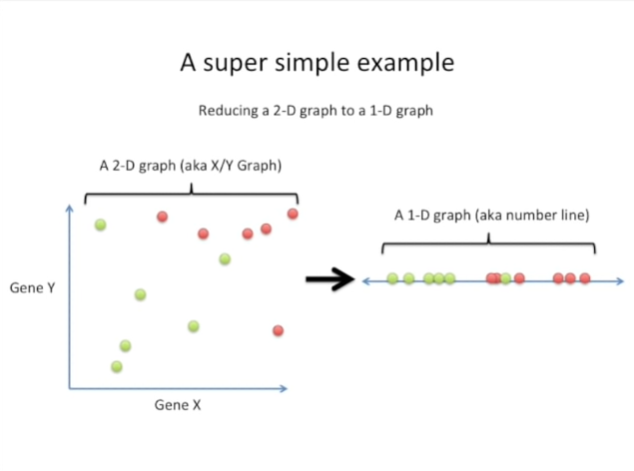
Here we're going to start with a super simple example.



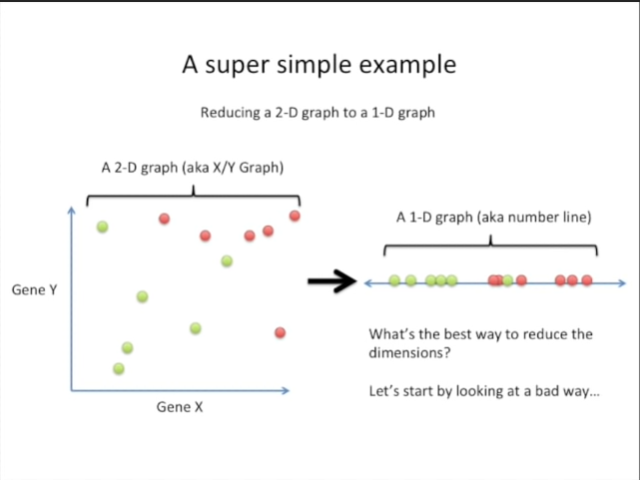
We're just going to try to reduce a two-dimensional graph to a 1d graph.



That is to say we want to take this two-dimensional graph aka and XY graph

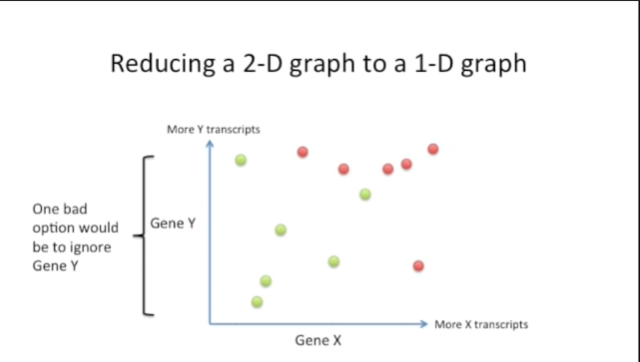


and reduce it to a one-dimensional graph aka a number line in such a way that maximizes the separable 'ti of the two categories.

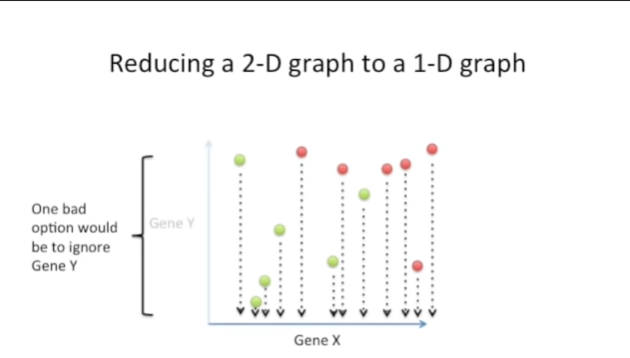


What's the best way to reduce the dimensions well ?

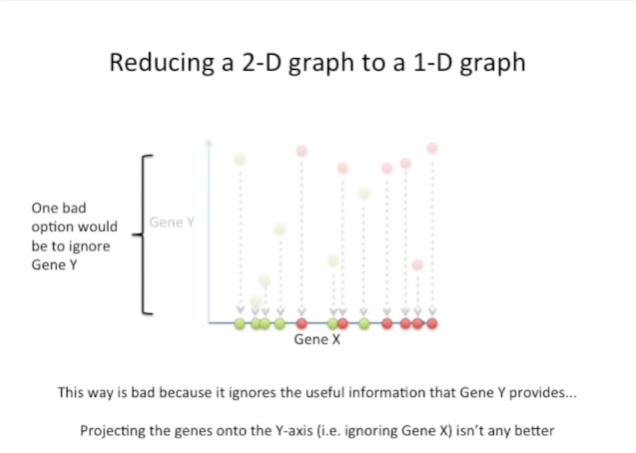
To answer that let's start by looking at a bad way and understanding what its flaws are.



One bad option would be to ignore gene Y.

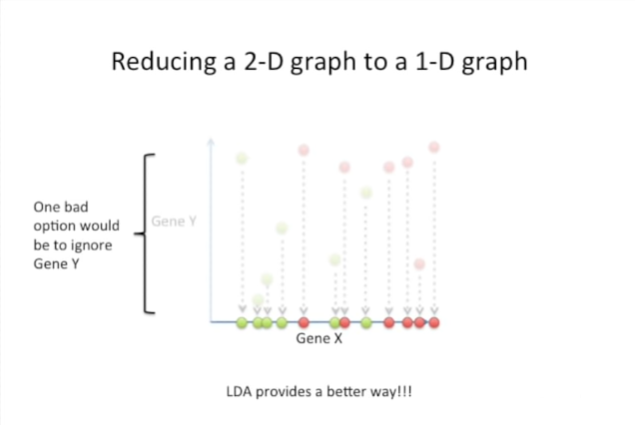


And if we did that we would just project the data down on to the x axis.

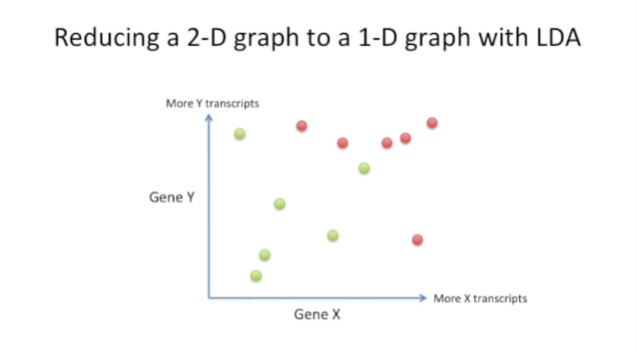


This is bad because it ignores the useful information that gene Y provides.

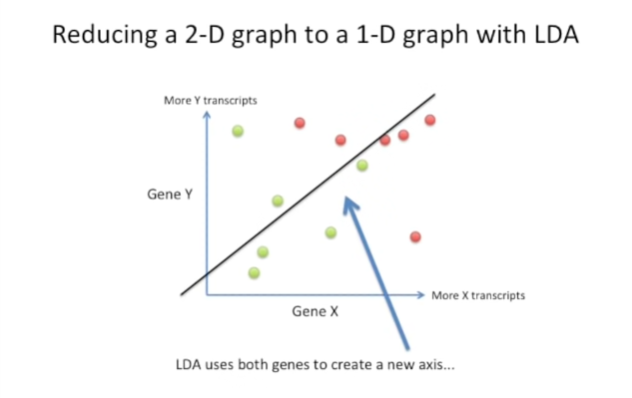
Projecting the genes onto the y axis ie ignoring the gene X isn't any better.



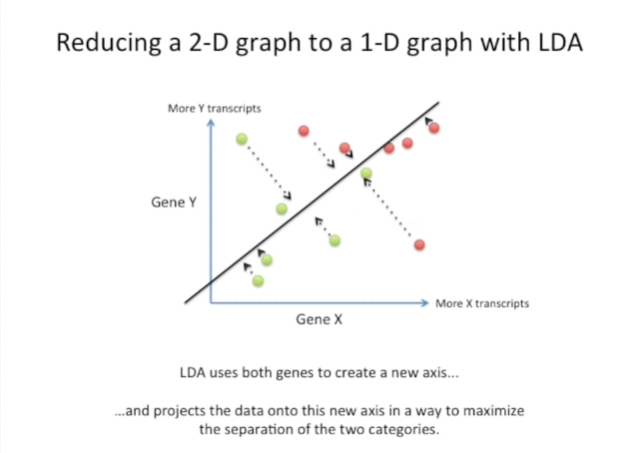
LDA provides a better way.



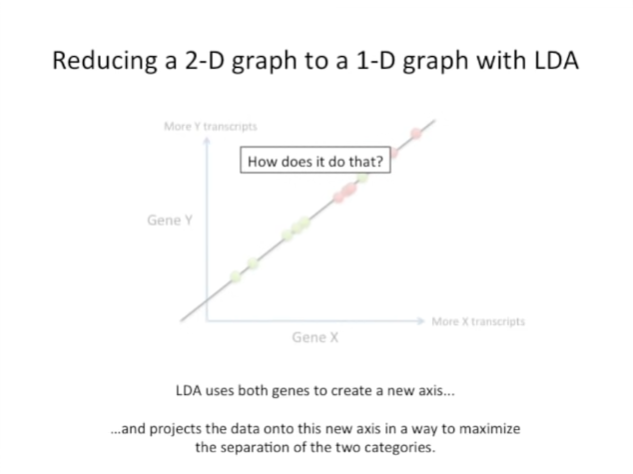
Here we're going to try to reduce this two-dimensional graph to a 1d graph using LDA.



LDA uses the information from both genes to create a new axis



and it projects the data onto this new axis in a way to maximize the separation of the two categories.

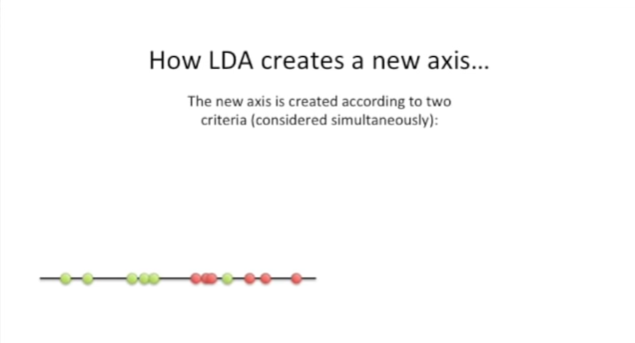


So the general concept here is that LDA creates a new axis and it projects the data onto that new axis in a way that maximizes the separation of the two categories .

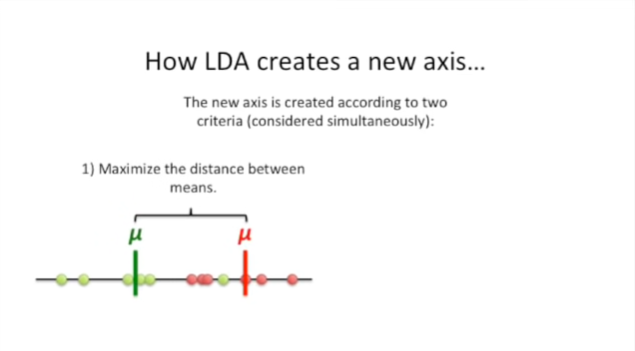
Now let's look at the nitty-gritty details and figure out how LDA does that.



How does LDA create the new axis.

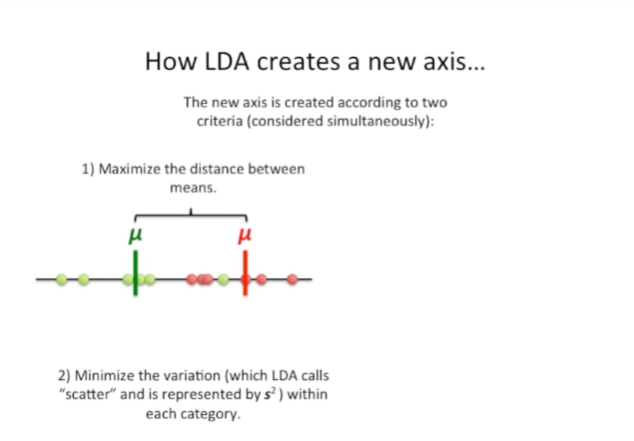


The new axis is created according to two criteria that are considered simultaneously.

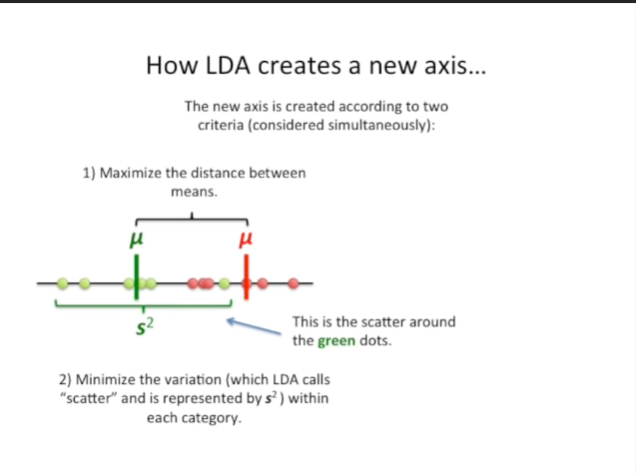


The first criteria is that once the data is projected onto the new axis, we want to maximize the distance between the two means.

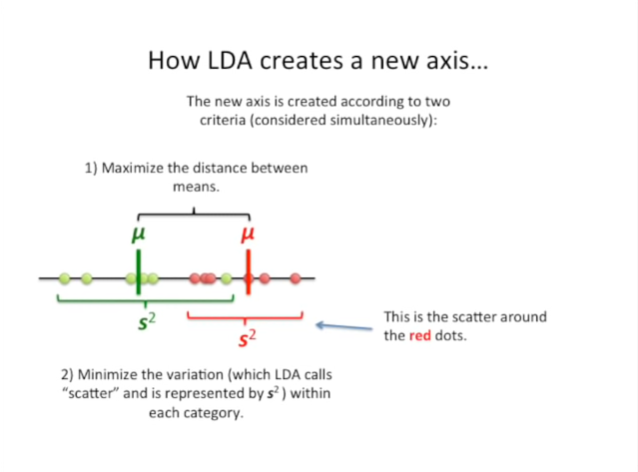
Here we have a green mu character which is a Greek character, representing the mean for the green category and a red mu representing the mean for the red category.



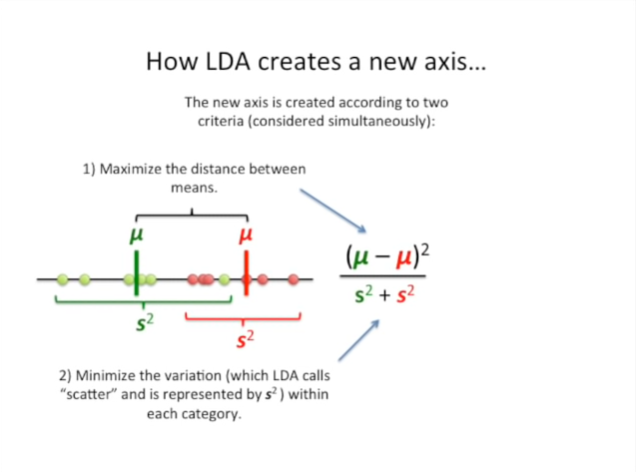
The second criteria is that we want to minimize the variation which LDA calls scatter and is represented by s squared within each category.



On the left side we see the scatter around the green dots.



On the right side we see the scatter around the red dots.



And this is how we consider those two criteria simultaneously.

We have a ratio of the difference between the two mean squared over the sum of the scatter.

The numerator is squared because we don't know if the Green mu is going to be larger than the red mu or the red Mu is going to be larger than the green mu.

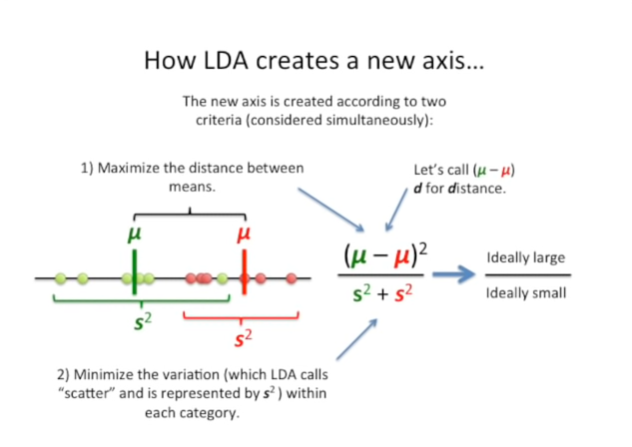
We don't want that number to be negative we want it to be positive.

So whatever it ishu um whether it's negative or positive begin with we square it and it becomes a positive number.

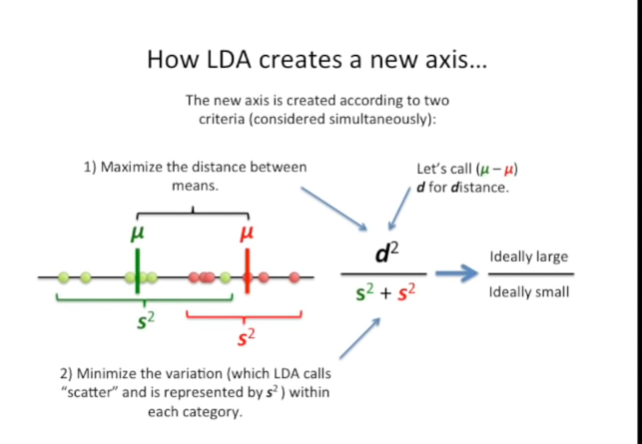
 Now ideally the numerator would be very large there'd be a big difference or a big distance between the two means.

And ideally the denominator would be very small and that the scatter ,the variation of the data around each mean in each category, would be small.

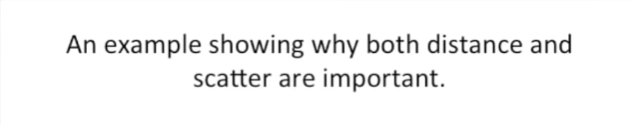
Now I know this isn't a very complicated equation but to make things simple later on in this discussion.



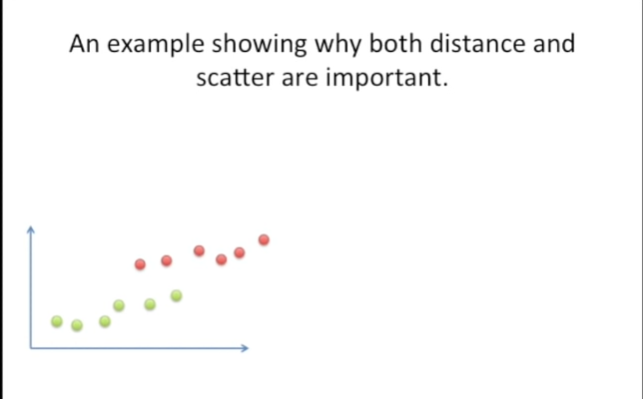
Let's call the difference between the two means D for distance.



So we can replace the difference between the two means with D.

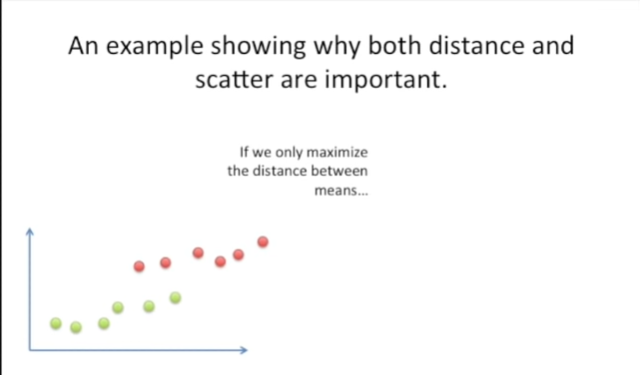


Now I want to show you an example of why both the distance between the two means and the scatter are important.

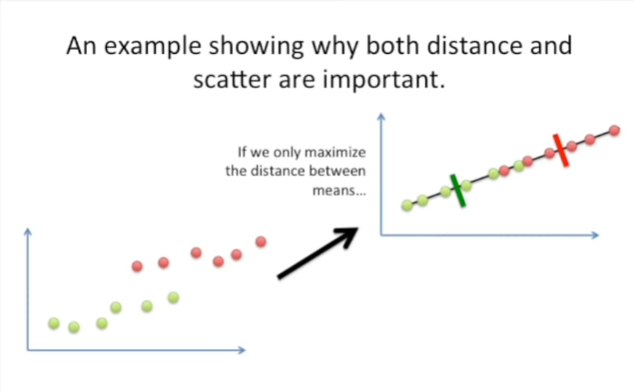


Here's a new data set we still just have two categories green and red.

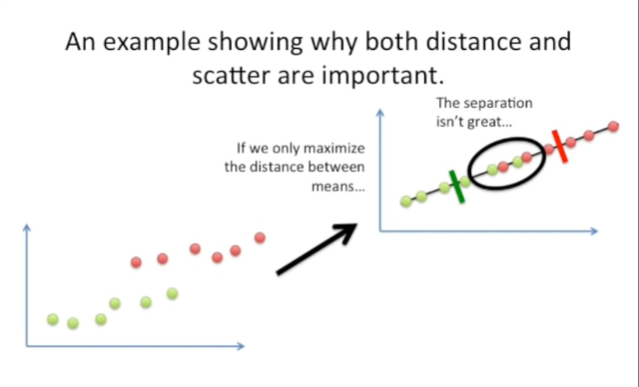
In this case, there's a little bit of overlap on the y-axis but lots of spread along the x-axis.



If we only maximize the distance between the means

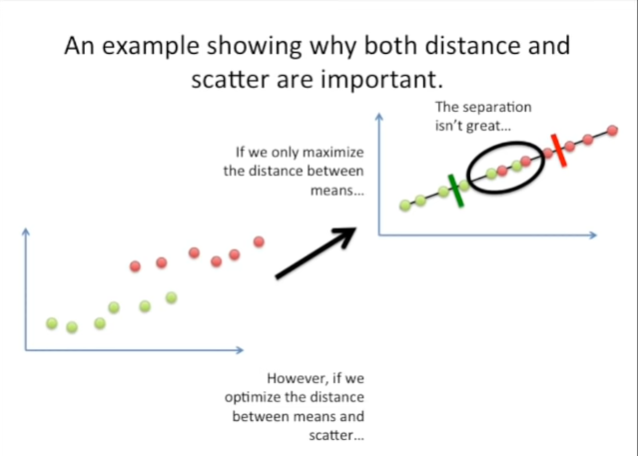


then we'll get something like this.

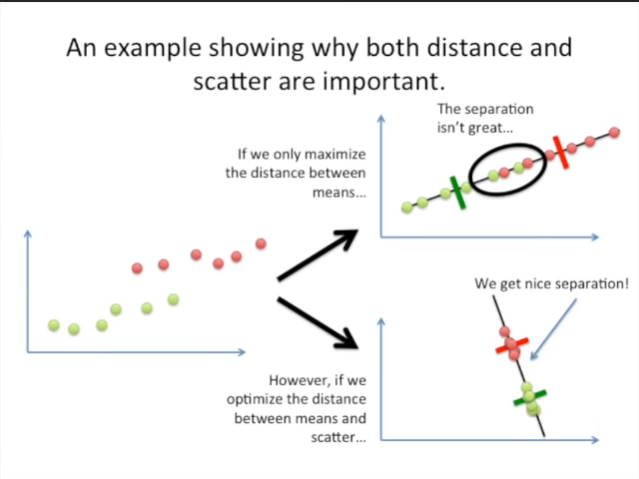


And the result is we'll have a lot of overlap in the middle.

This isn't great separation.

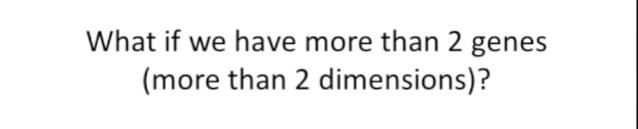


However if we optimize the distance between the means and the scatter.



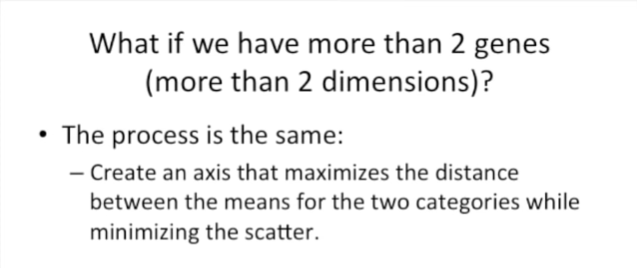
Then we get nice separation.

Here the means are a little closer to each other than they were in the graph on the top, but the scatter is much less. So if we optimize both criteria at the same , we can get good separation.



So what if we have more than two genes ?

That is to say what if we have more than two dimensions ?



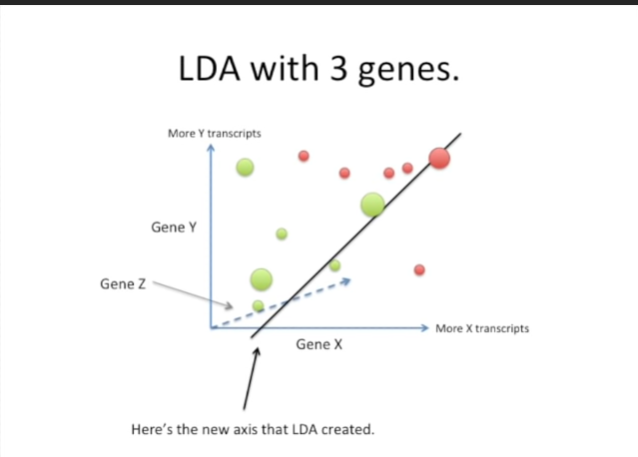
The good news is that the process is the same.

We create a new access that maximizes the distance between the means for the two categories while minimizing the scatter.

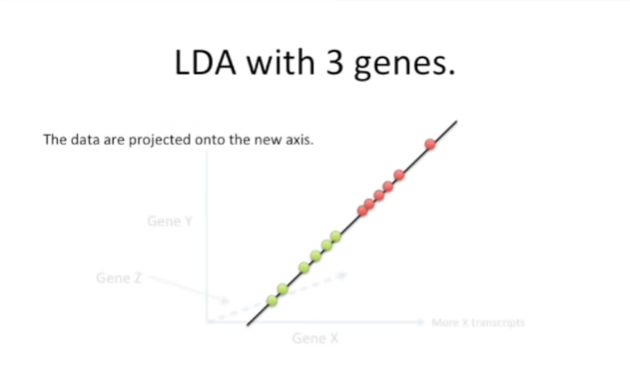


So here's an example of trying to do LDA with three genes.

We've got that three dimensional graph that I showed you earlier.



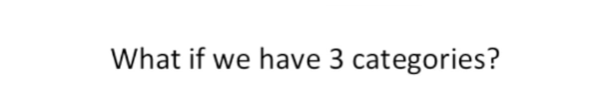
Here we've created a new axis.



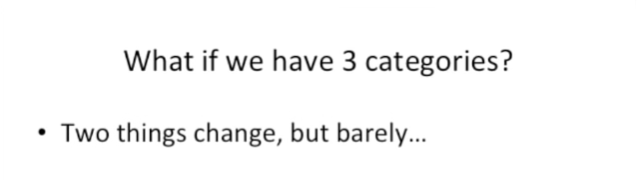
And the data are projected onto the new axis.



This new axis was chosen to maximize the distance between the two means, between the two categories, that is while minimizing the scatter.



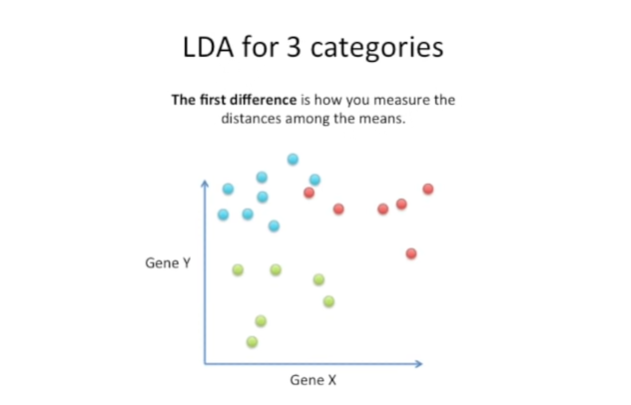
What if we have three categories ?



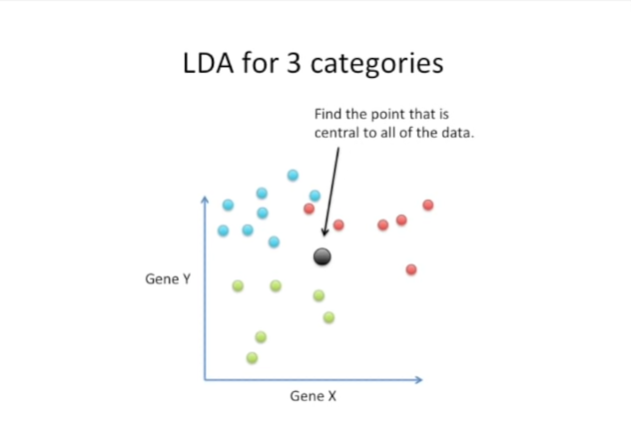
In this case two things change but just barely.



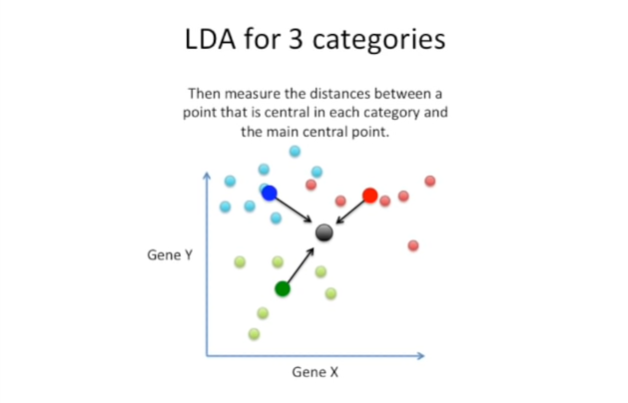
Here's a plot that has two genes, but now we have three categories.



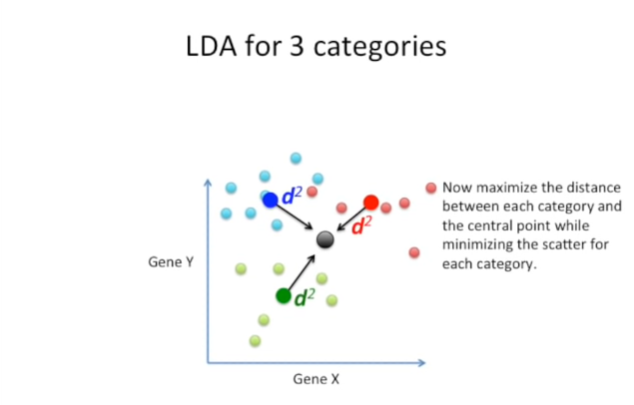
The first difference between having three categories as opposed to just two categories like we have four, is how we measure the distances among the means.



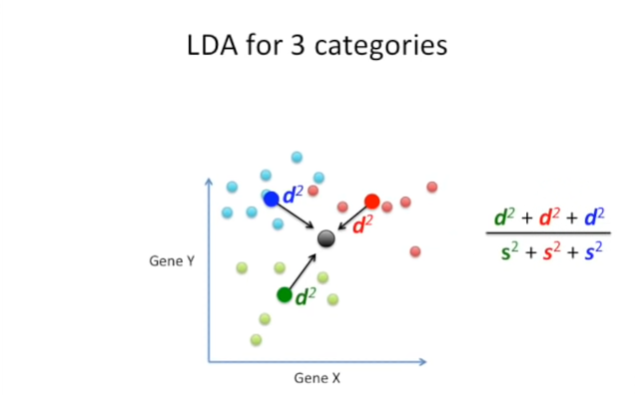
Instead of just measuring the distance between the two means, we first find a point that is central to all of the data.



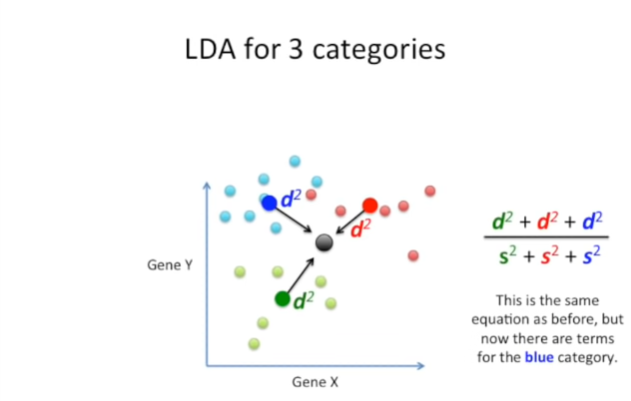
Then we measure the distances between a point that is central in each category and the main central point.



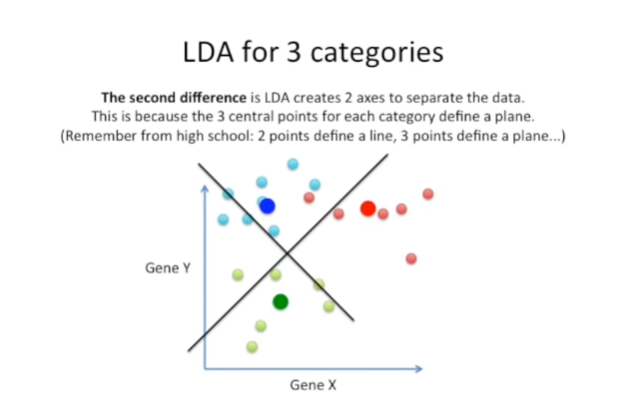
Now we want to maximize the distance between each category in the central point while minimizing the scatter for each category.



And here's the equation that we want to optimize.



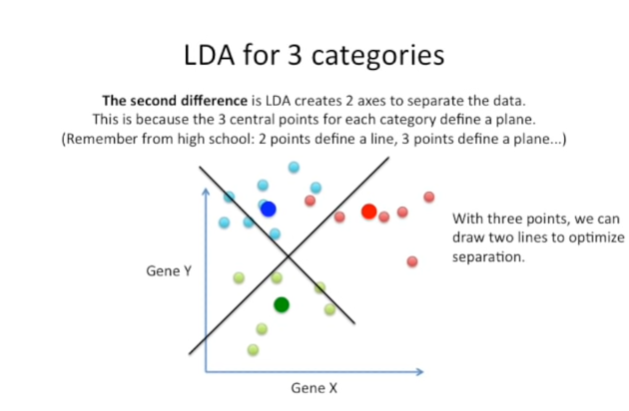
And this is the same equation as before but now there are terms for the blue category.



The second difference is LD a creates two axes to separate the data.

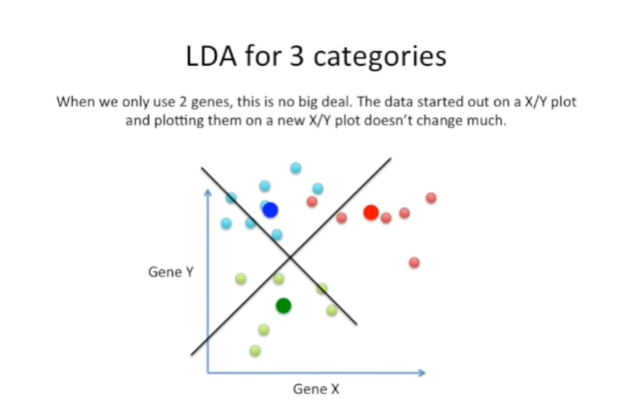
This is because the three central points for each category define a plane.

Remember from high school two points to find a line and three points to find a plane.



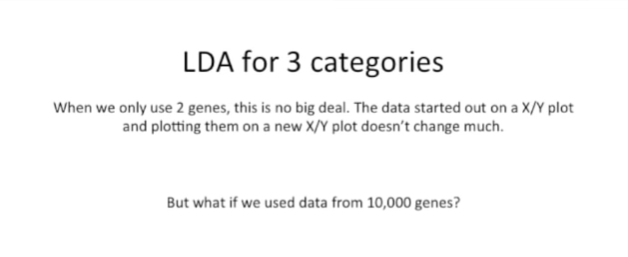
That is to say we create new x and y axes.

However these are now optimized to separate the categories.

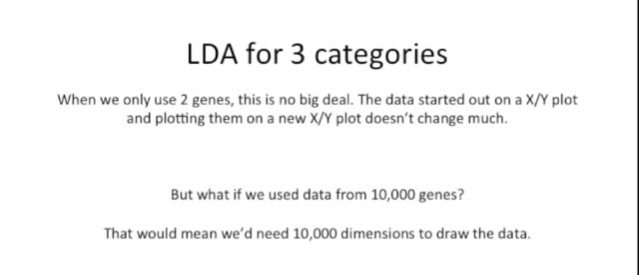


When we only use two genes this is no big deal.

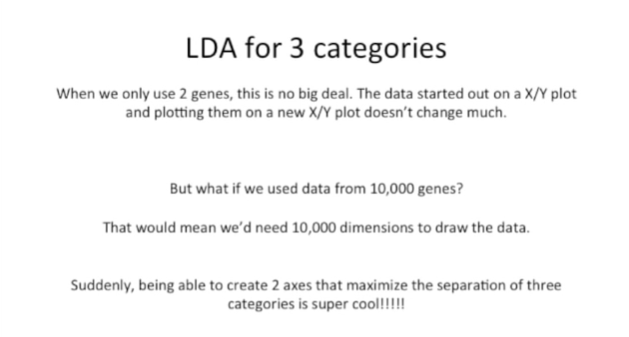
The data started out on an X/Y plot and plotting them on a new X/Y plot doesn't change all that much.



But what if we use data from 10,000 genes ?

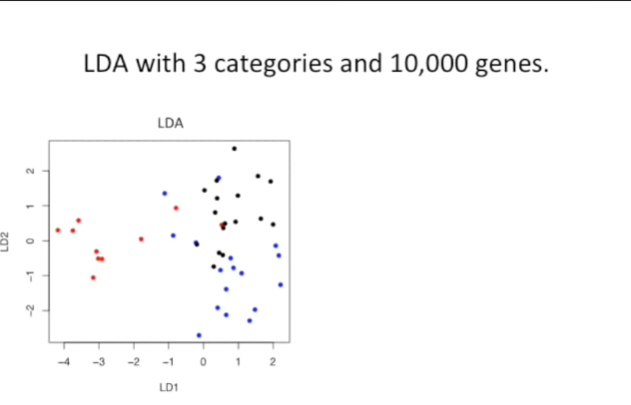


That would mean we need 10,000 dimensions to draw the data.



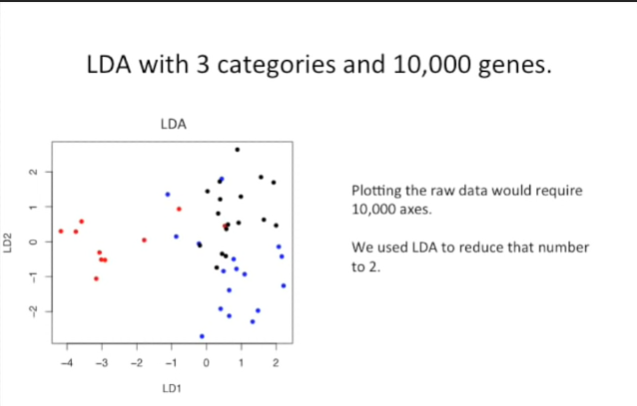
Suddenly being able to create two axes that maximize the separation of the three categories is super cool.

It's way better than drawing a 10,000 dimension figure that we can't even imagine what it would look like.

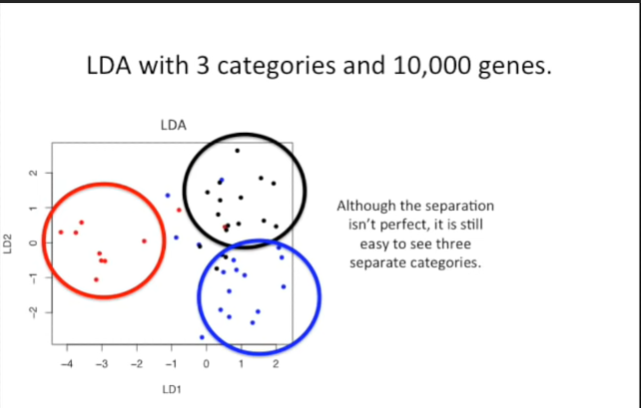


Here's an example using real data I'm trying to separate three categories.

And I've got 10,000 genes.



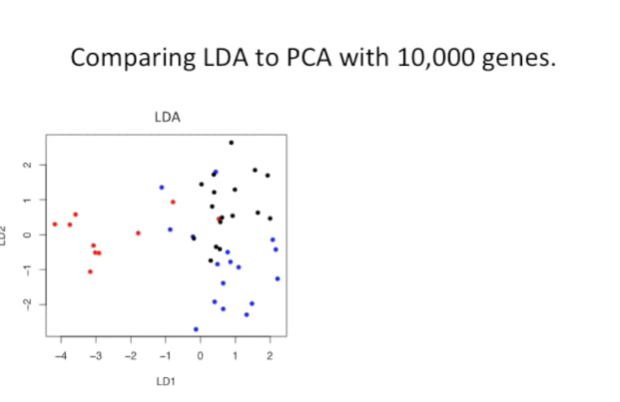
Plotting the raw data would require 10,000 axes we used LDA to reduce the number to two.



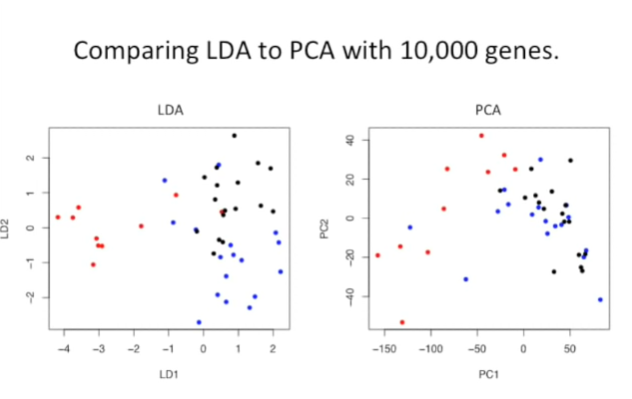
And although the separation isn't perfect, it is still easy to see three separate categories.



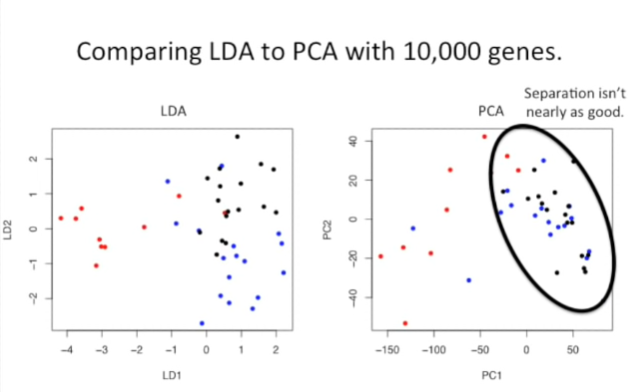
Now let's use that same data set to compare LDA to PCA.



Here's the LDA plot that we saw before.



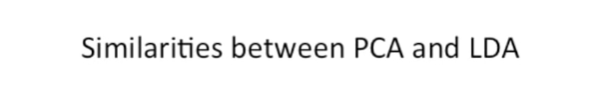
And now we've applied PCA to the exact same set of genes.



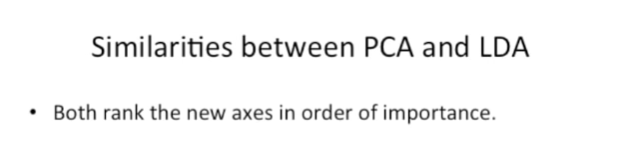
PCA doesn't separate the categories nearly as well.

We can see lots of overlap between the black and the blue points.

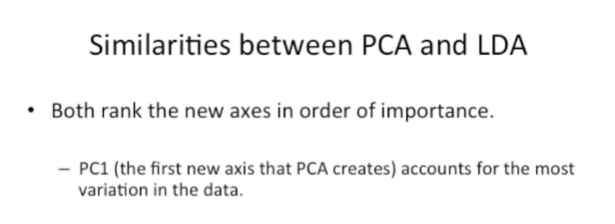
However PCA wasn't even trying to separate those categories, it was just looking for the genes with the most variation.



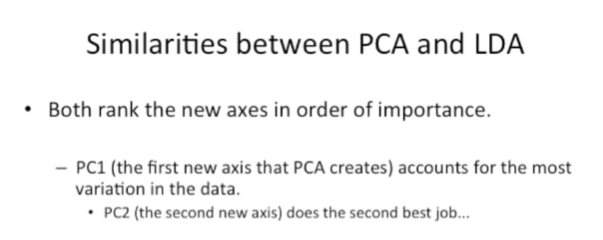
So we've seen the differences between LDA and PCA, but now let's talk about some of the similarities.



The first similarity is that both methods rank the new axes that they create in order of importance.

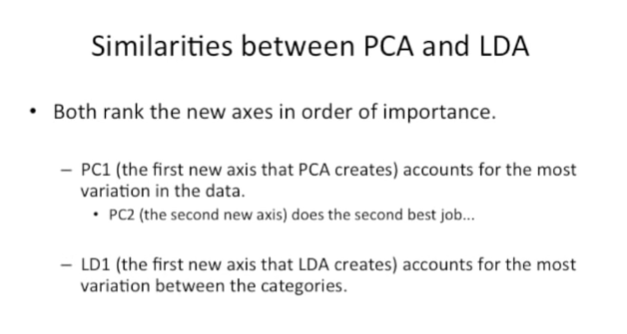


PC1 the first new axis that PCA creates, accounts for the most variation in the data.

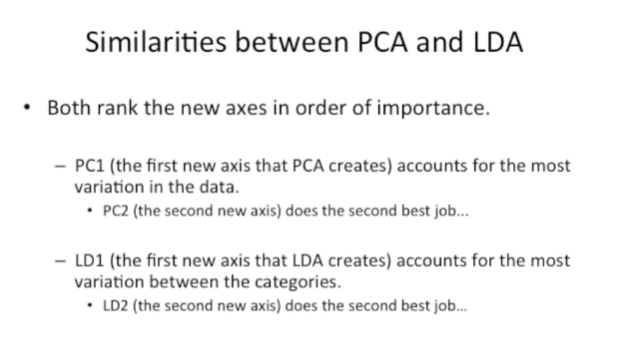


Likewise PC2, the second new access, does the second best job.

And this goes on and on for the number of axis that are created from the data.

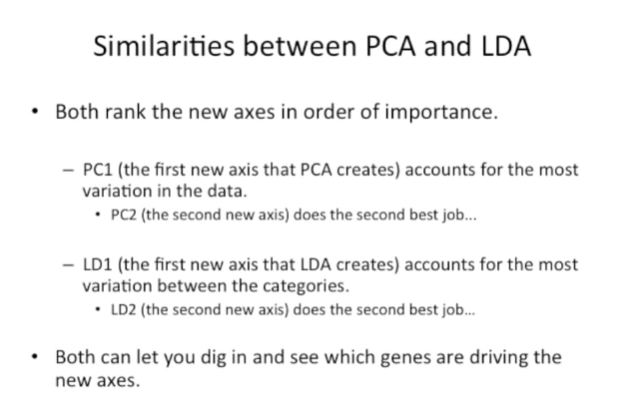


LD1 the first new axis that LDA creates, accounts for the most variation between the categories.



LD2, the second new axis, does the second best job.

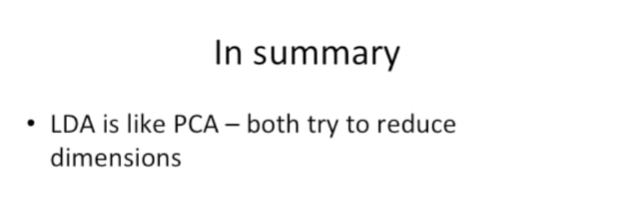
Etc etc etc.



Also both methods let you dig in and see which genes are driving the new axes.

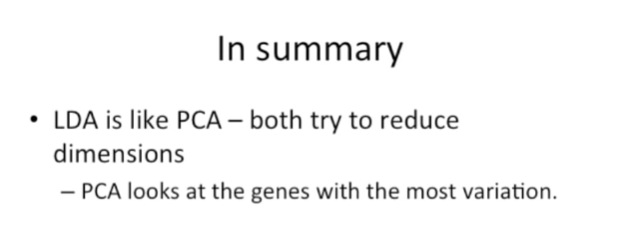
In PCA this means looking at the loading scores.

In LDA one thing you can do is look and see which genes or which variables correlate with the new axis.

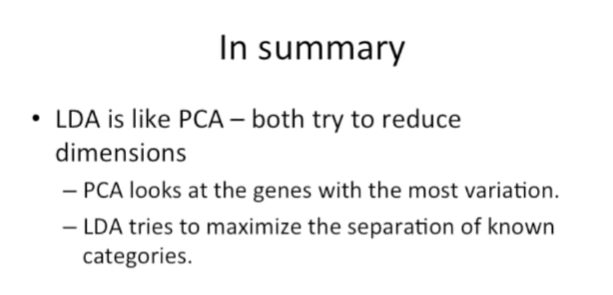


So in summary LDA is like PCA.

Both try to reduce dimensions.



PCA does this by looking at the genes with the most variation.



In contrast LDA tries to maximize the separation of known categories and that's it.