Principal Component Analysis - Covariance and Eigenvectors

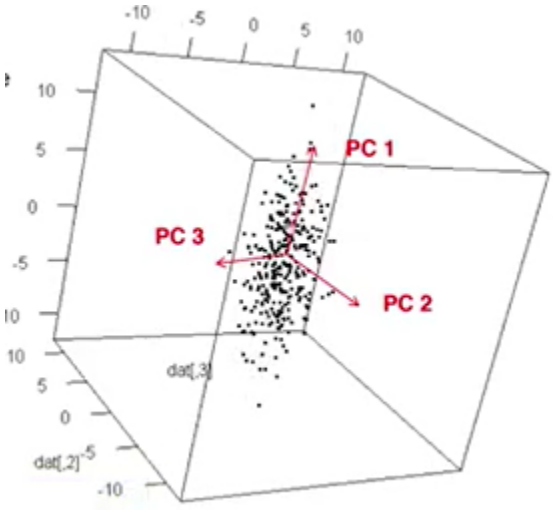
In this lecture, we will talk a bit further about what Principal Component Analysis (PCA) actually does.

So, there are different definitions of it and we have already discussed the intuition behind it in the last lecture.

Definition 1: **Maximize projection variance**

Start with centered data $X \in R^{n * p}$

- PC 1 is the direction of largest variance.
- PC 2 is
  - perpendicular to PC 1
  - again largest variance
- PC 3 is
  - perpendicular to PC 1, PC 2
  - again largest variance
- etc.

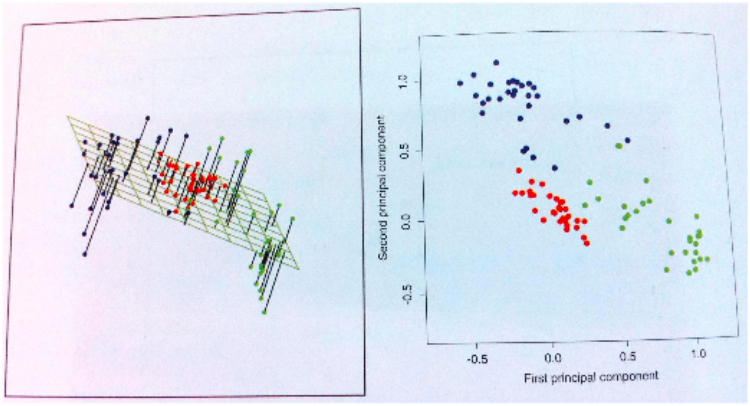


So first we have our samples. And they live in some high-dimensional space. We only showed three dimensions here in the image, but they actually live in high dimensional space. And what we are trying to find are the directions that vary the most. These directions that vary the most are the only ones we are going to keep, like what we saw in the previous lecture.

For example, if we want to get a two dimensional representation, we just keep the two directions that vary the most. The intuition is quite clear, we don't need to keep the various directions that don't vary because they don't really contain much information about what makes them different from each other and so that's the first way of seeing what PCA does.

Definition 2: Minimize projections residuals

- PC 1: Straight line with the smallest orthogonal distance to all points
- PC 1 & PC 2: Plane with the smallest orthogonal distance to all points.
- etc.



So, we will discuss the second way of seeing what PCA does. In fact, both the previous method and this method are quite the same, but intuitively they are slightly different. So, either we can keep the directions that vary the most, but actually, this is equivalent to removing the directions that carry the least signal.

So if we are trying to remove the ones that have the least signal it means that whatever the signal we are using (for example, we can see points in a three-dimensional plane in the above image) we want to find the projection of that, and eventually, all of these signals are going to be lost if we are projecting them all out into this two-dimensional space, and this signal we lost is the smallest one possible.
So that's what we mean by minimizing the projection residuals. So residuals are whatever is left out that we weren't able to capture by this projection and that signal that is being lost needs to be as small as possible.

In the previous method, we kept the biggest signal possible that we could keep when we were going from a 10,000-dimensional space to a two-dimensional space.
So both of these methods are very intuitive ways of two things that we may want to do but in fact, they are quite the same. So, we can try to find the projection that keeps the

directions of the largest variation, or we can try to find a projection that minimizes the amount of signal lost, and we can show that these two things are exactly the same,

Now the question is how do we even compute it i.e how do we find directions that maximize the amount of variation captured or find the directions that lets us lose the least amount of signal?
How do we actually do this? How does a computer do it?
It's really quite exciting that a computer can do it by just doing something that is known as a **Spectral decomposition**.

- Covariance matrix (or correlation matrix) $R = \frac{1}{n}X^TX$ is symmetric and positive semidefinite.
- **Spectral Decomposition Theorem:** Every real symmetric matrix R can be decomposed as

  $$R = V \wedge V^T$$

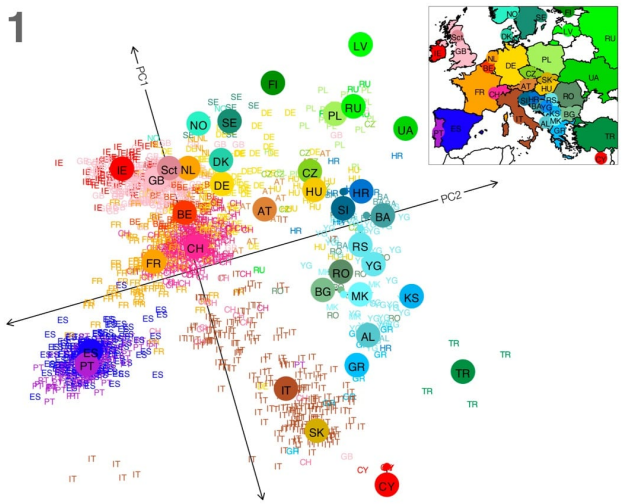
  Where $\wedge$ is diagonal and V is orthogonal.
- Columns of V (= Eigenvectors of R) are the PCs.
- Diagonal entries of $\wedge$ (= Eigenvectors of R) are variances along PCs.

So what it actually does is to take the covariance matrix of the data or the correlation matrix which says how are two different variables i.e buying behavior of one person and buying behavior of the other person, correlated with each other?

And in this correlation matrix, all we have to do is to just find the Eigenvectors corresponding to the largest Eigenvalues, and the computer can do this very easily.
In fact, a lot of what we do mathematically, when we're doing things with data analysis is finding these Eigenvectors and Eigenvalues, which is one of the most standard computations that we can do with a matrix. So it's just finding Eigenvectors and Eigenvalues.

In fact, we only have to find the directions here i.e the direction of the largest spreads, or the directions that minimize the amount of signal loss that corresponds to the Eigenvector corresponding to the largest Eigenvalue. So if we want to represent it in one dimension then we are done. But if we want to go to two dimensions, then we take the next direction i.e take the Eigenvector corresponding to the second largest

Eigenvalue. Similarly, if we want to represent data in three dimensions, we also can take the Eigenvector corresponding to the third-largest Eigenvalue. So, we just keep as many of these Eigenvectors as we want in our representation to be, and often we would only go up to two dimensions because we would like to find the representation in two dimensions.



That's exactly what was done in this particular example. So we just kept the Eigenvector corresponding to the largest Eigenvalue, that's PC1. And then the Eigenvector corresponding to the second largest Eigenvalue, that's PC2, and that's it i.e just representing the data in this space corresponding to these two Eigenvectors is exactly what was done to get this plot here.

So this is a very standard mathematical tool that people use. So when we have a matrix we do this Eigenvector decomposition and we just keep the largest Eigenvectors and we represent our data in that space and this is exactly what PCA does. We can see here that it's been very effective in this particular application.

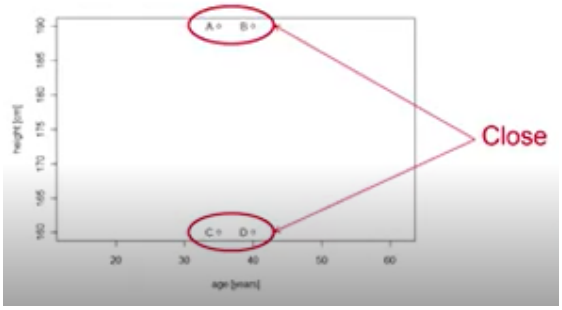
This course is about also giving you intuition for things that you have to be careful about. You also need to be careful about something here when we're talking about Principal Component Analysis.

So let's say we have here a very simple dataset, where we have people (for example we have only 4 people here), and what we're collecting is data on age and data on height.

Person	Age (years)	Height (cm)
A	35	190
B	40	190
C	35	160
D	40	160

So, of course, the age can be measured in years, in days, and similarly, in months. And for heights, here we have measured it in centimeters, but you can also measure it in feet, etc. So let's think through whether it actually matters which kind of units we are measuring things in when we're doing Principal Component Analysis.

It's really important to think through this because you don't really want that your picture changes when you're just changing the units. For example, when we are going in height from cm to feet, we would like the picture to remain the same because the data is the same. It's just that we changed the units.



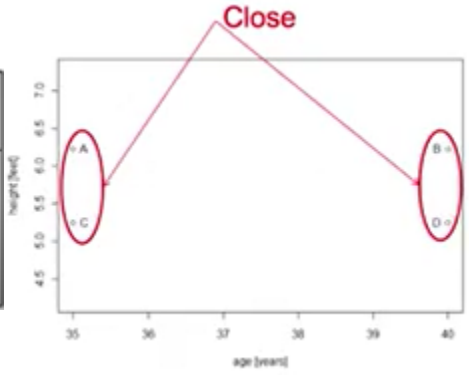
But think about what PCA does. It tries to find the directions of the largest spread. Well, if we change something and just multiply our numbers by 1000 or 100,000 in one direction, **then suddenly this direction becomes the direction of the largest amount of variability just by how we are scaling the data.**

So it's a problem for PCA that in its standard form, it actually does depend on the units that we choose.

So for example, here in this little data set, if we measure height in cm (centimeters), the direction of the largest spread is in a direction which is along the height.

However, if you change it to feet, the direction of the largest spread is in a direction here which is along the age.

Person	Age (years)	Height (feet)
A	35	6.232
B	40	6.232
C	35	5.248
D	40	5.248



So we see that in PCA, the picture that we're getting of our first principal component is in fact different depending on whether we're measuring height in feet or whether we're measuring height in centimeters.

It is just because centimeters have a larger spread as we are multiplying it with some value.

So that's really problematic in terms of PCA and when you have datasets which have very different units. For example, here we have age and we have height. They're very different units, since one is measured in years and one is measured in feet. So it's not clear how to normalize them against each other.

Then what we should do actually is instead of the covariance matrix, we should always use the **correlation matrix** when we have different units in different entries.

It's important to use the correlation matrix because the correlation matrix normalizes every one of the directions of variables so that all of them carry the same weight.
So here we have age carrying the same weight as height. We're normalizing all these different variables against each other.

This is really important. Because if we change the units it can just completely change our first principal components. So this is something that we really have to take into account when we're doing PCA.

If we have the same kinds of units, it doesn't matter as much. If all of the variables are measured in feet then the differences actually mean something. So then we can use the covariance matrix. But if we're measuring many different types of units in the matrix-like age, height, weight, etc. and these are all units that we have in our data set. Then it is really important to use the correlation matrix. Because depending on how we scale things, our plots are going to become very different.

So that was Principal Component Analysis.
Next, we'll talk about a nonlinear method, and then see how these two methods actually compare against each other.