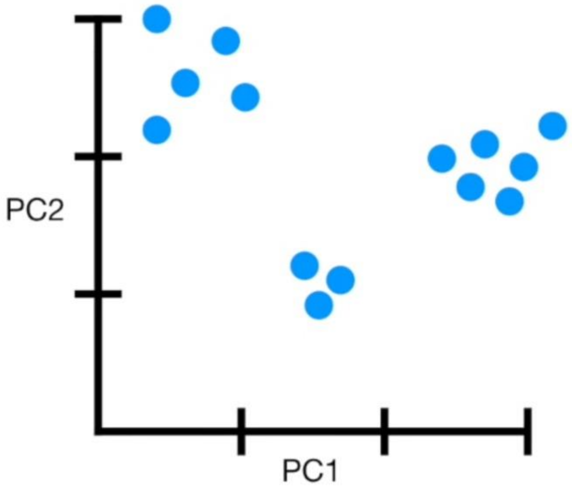
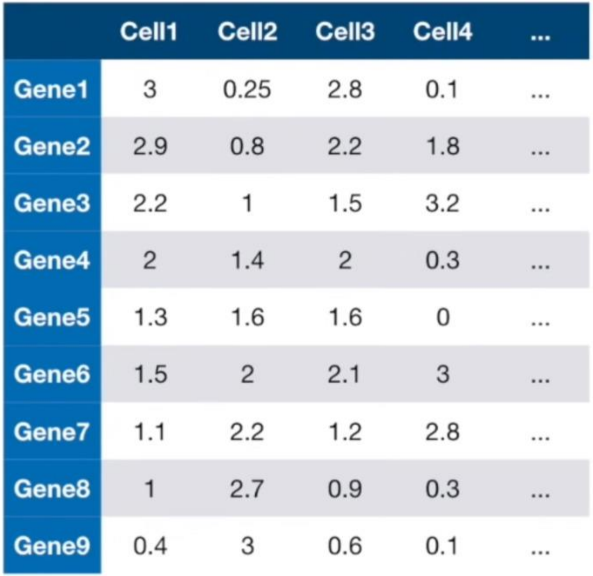
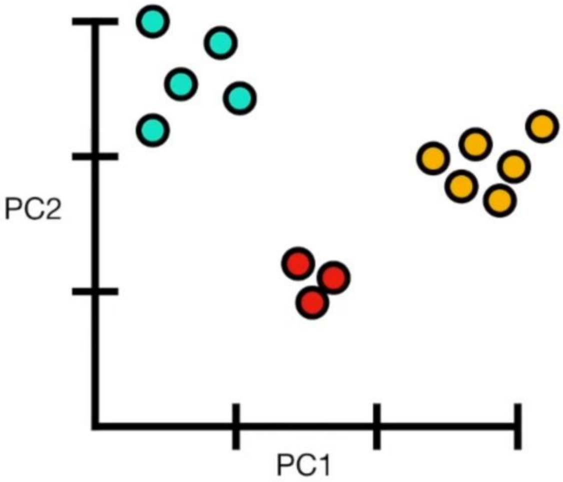
A PCA plot converts the correlations (or lack there of) among all of the cells into a 2-D graph.

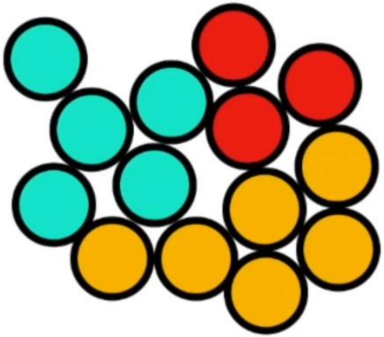




	Cell1	Cell2	Cell3	Cell4	...
Gene1	3	0.25	2.8	0.1	...
Gene2	2.9	0.8	2.2	1.8	...
Gene3	2.2	1	1.5	3.2	...
Gene4	2	1.4	2	0.3	...
Gene5	1.3	1.6	1.6	0	...
Gene6	1.5	2	2.1	3	...
Gene7	1.1	2.2	1.2	2.8	...
Gene8	1	2.7	0.9	0.3	...
Gene9	0.4	3	0.6	0.1	...



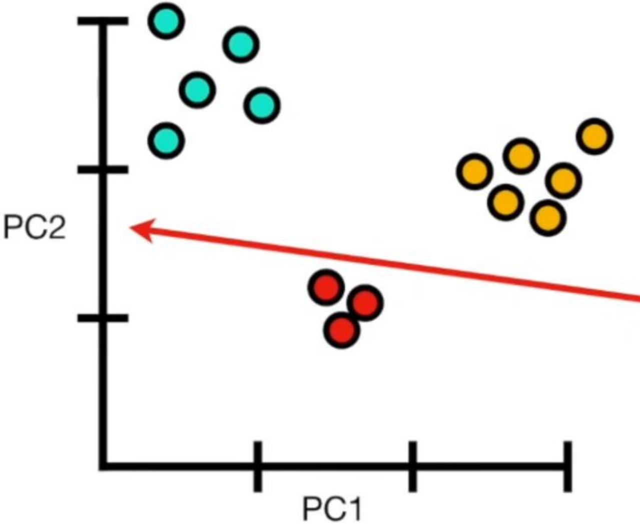
Once we've identified the clusters in the PCA plot, we can go back to the original cells…



…and see that they represent 3 different types of cells doing 3 different things with their genes!!!!

Here’s one last main idea about how to interpret PCA plots:

The axes are ranked in order of importance.

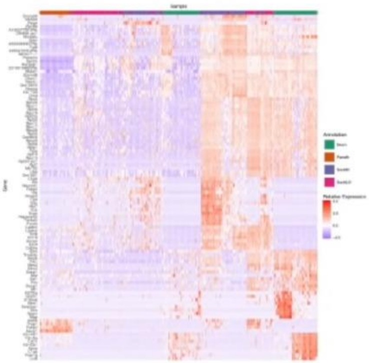


Differences along the first principal component axis (PC1)…

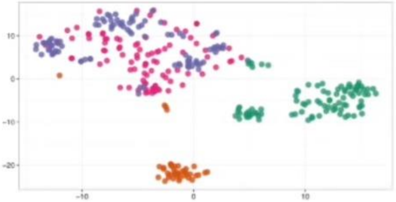
…are more important than differences along the second principal component axis (PC2).

Before we go, you should know that PCA is just one way to to make sense of this type of data. There are lots of other methods that are variations on this theme of “dimension reduction”.

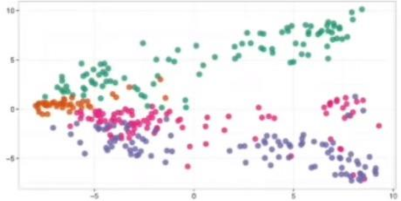
Heatmaps

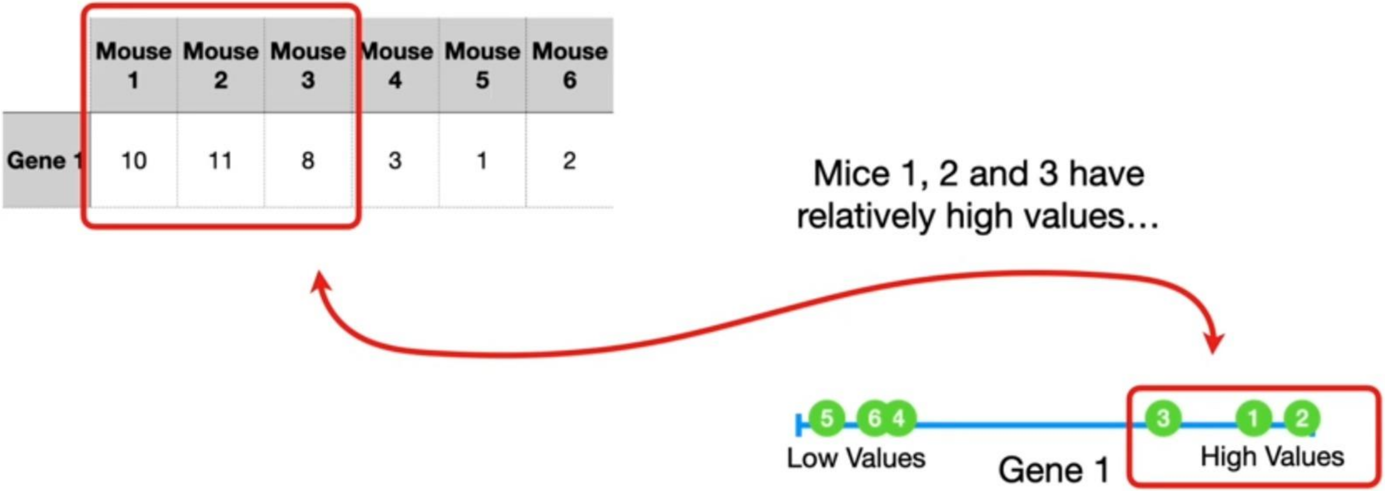


t-SNE Plots



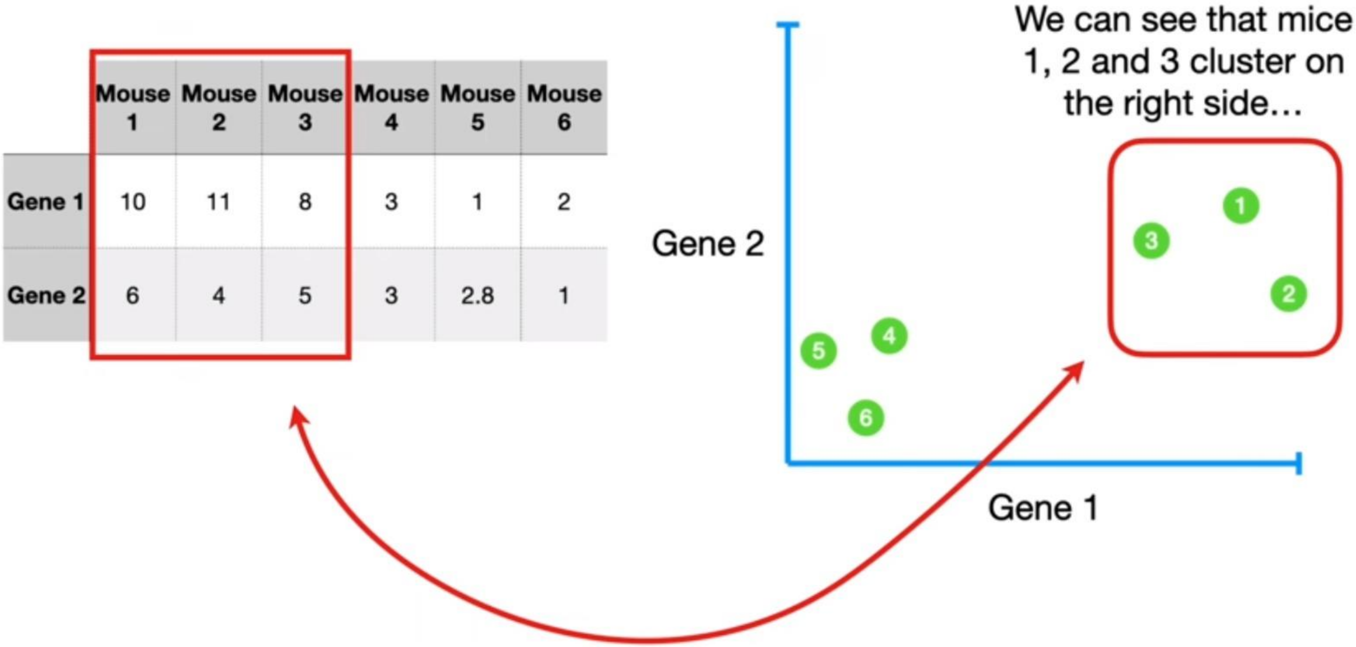
Multi-Dimensional Scaling (MDS)





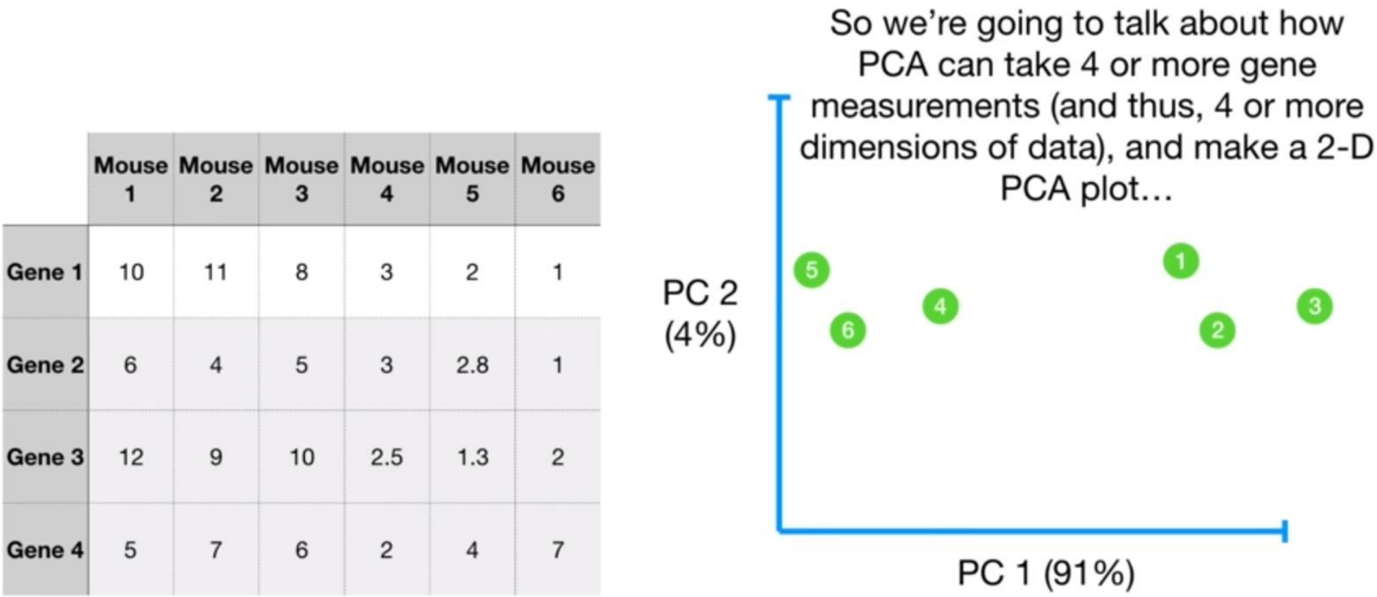
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	1	2

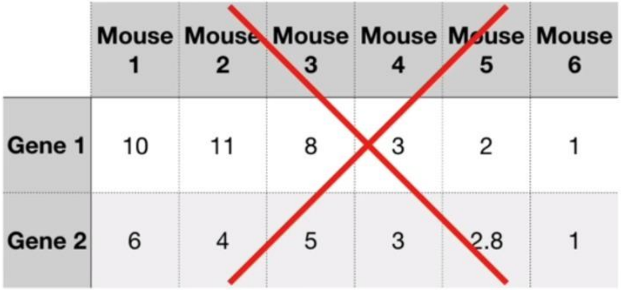
Mice 1, 2 and 3 have relatively high values…



	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	1	2
Gene 2	6	4	5	3	2.8	1

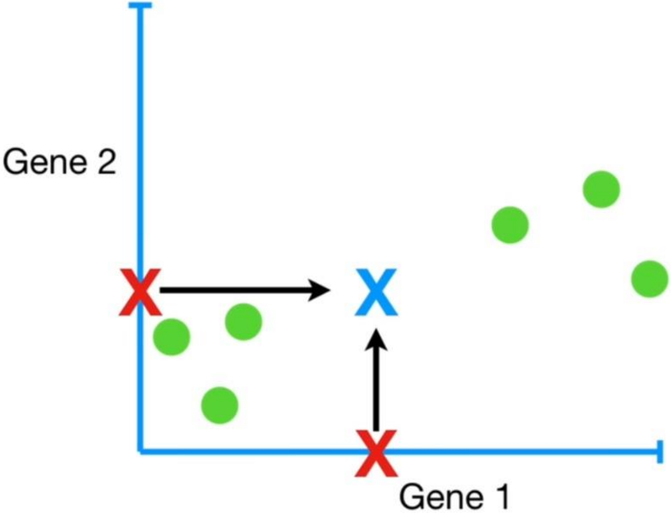
We can see that mice 1, 2 and 3 cluster on the right side…





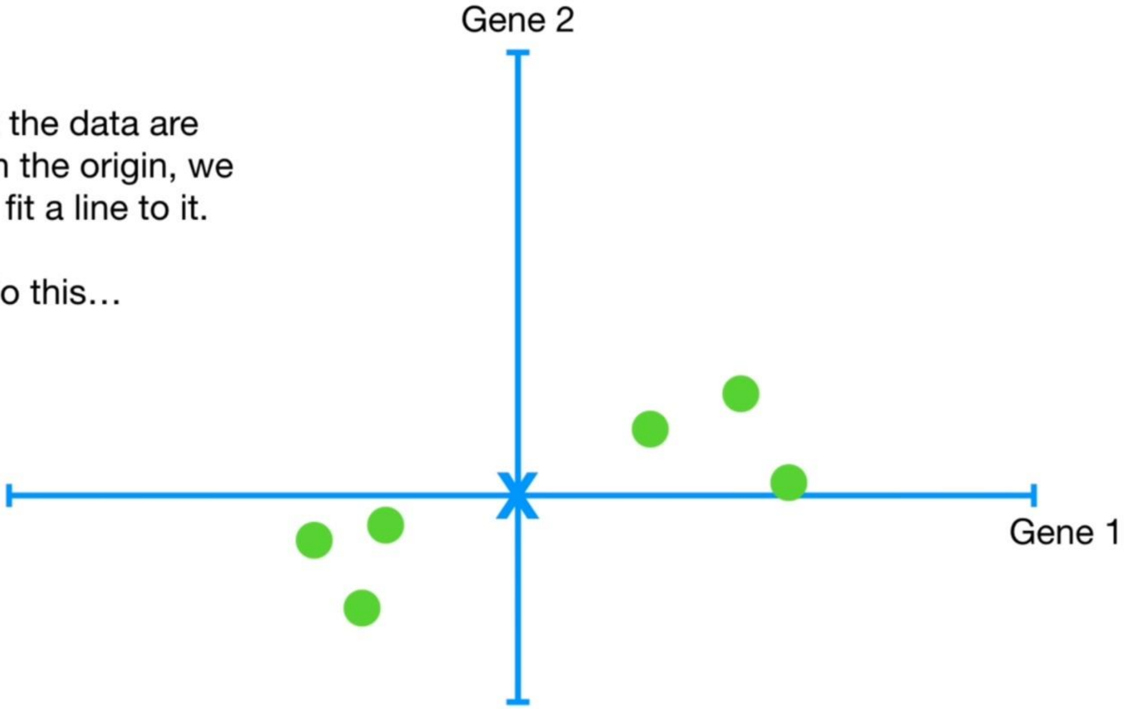
	Mouse 1	Mouse 2	Mouse 3	Mouse 4	Mouse 5	Mouse 6
Gene 1	10	11	8	3	2	1
Gene 2	6	4	5	3	2.8	1

From this point on, we'll focus on what happens in the graph; we no longer need the original data…

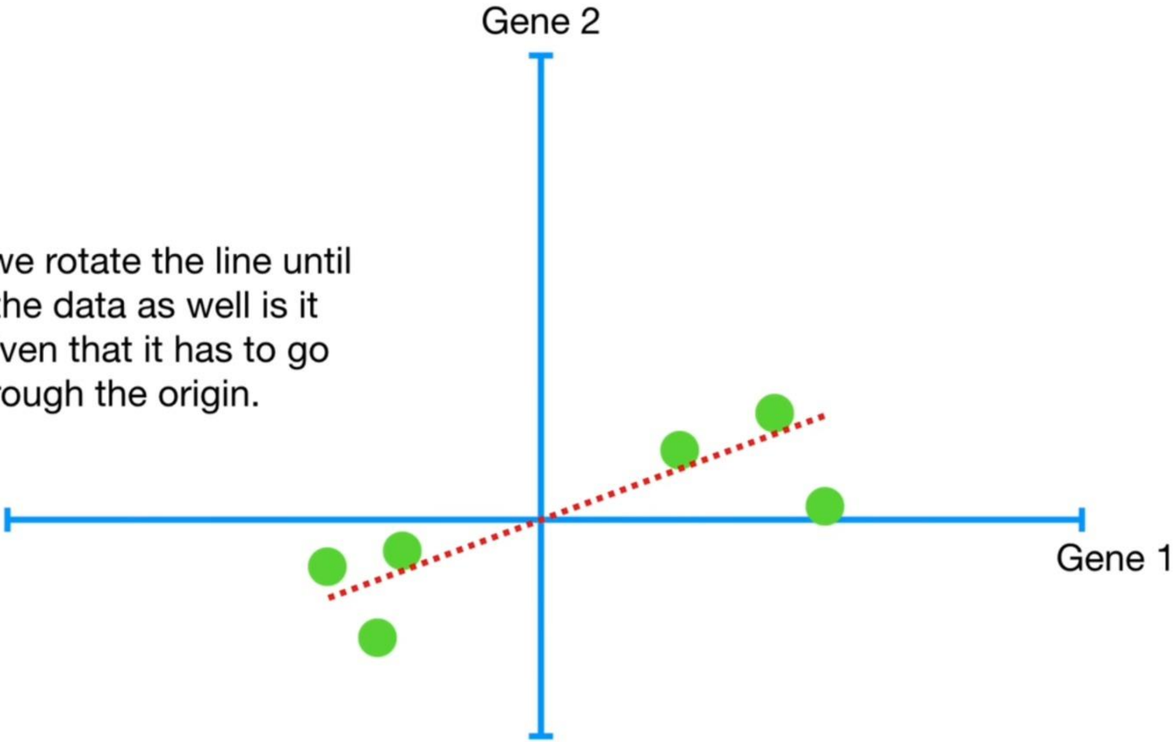


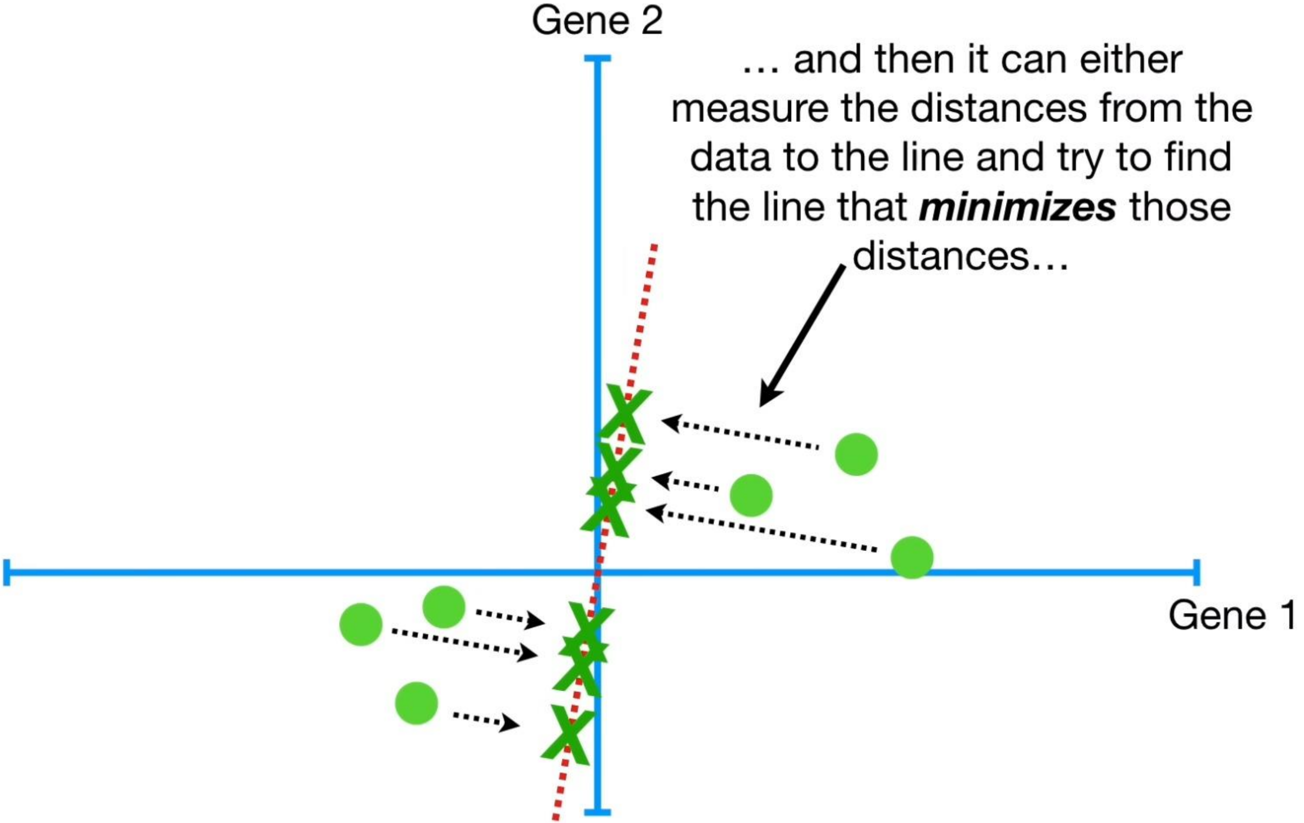
Now that the data are centered on the origin, we can try to fit a line to it.

To do this…



…then we rotate the line until it fits the data as well is it can, given that it has to go through the origin.

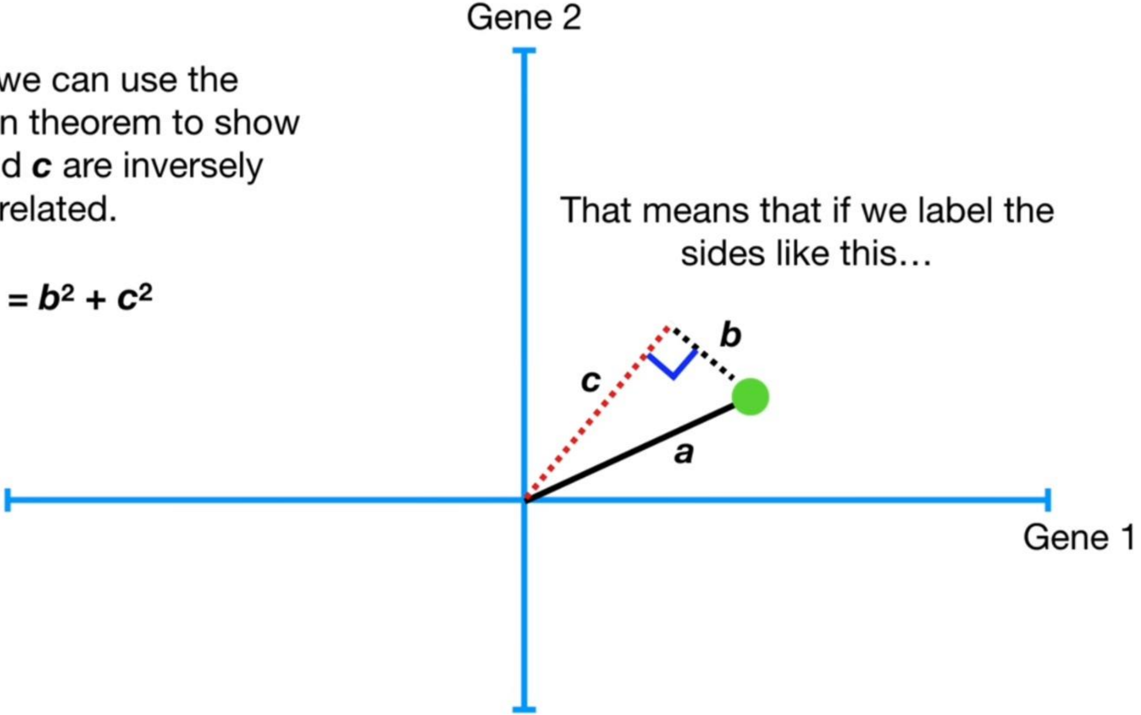




... and then it can either measure the distances from the data to the line and try to find the line that ***minimizes*** those distances...

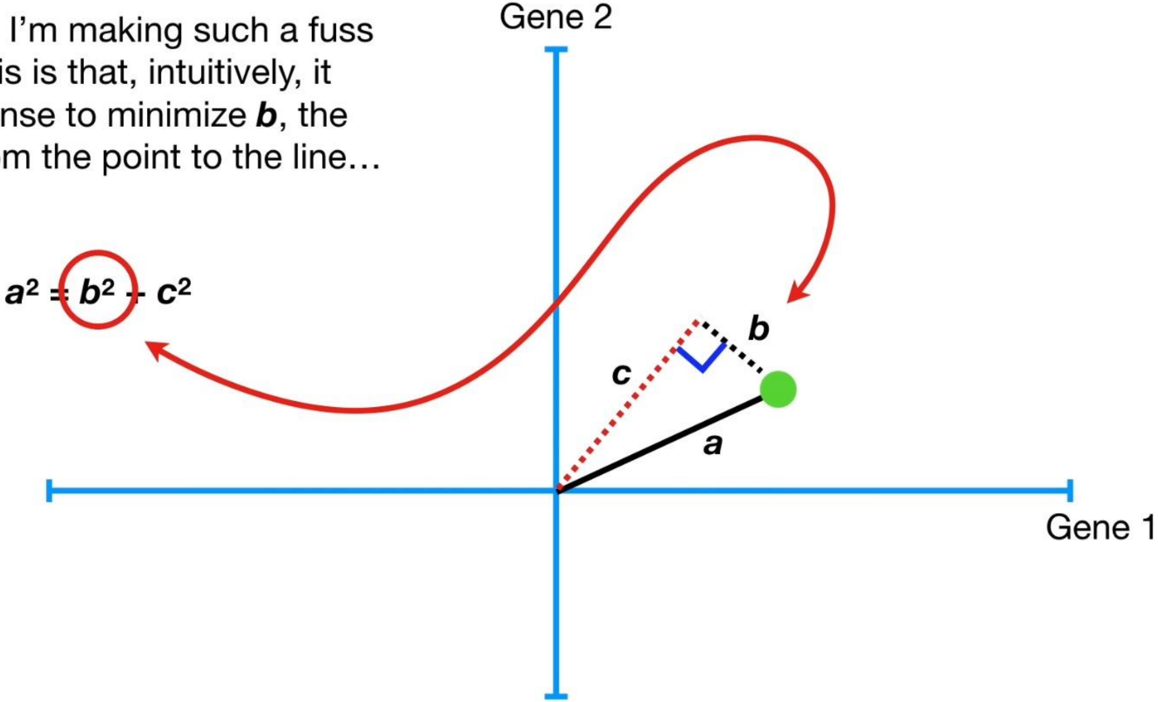
…then we can use the Pythagorean theorem to show how ***b*** and ***c*** are inversely related.

$$a^2 = b^2 + c^2$$



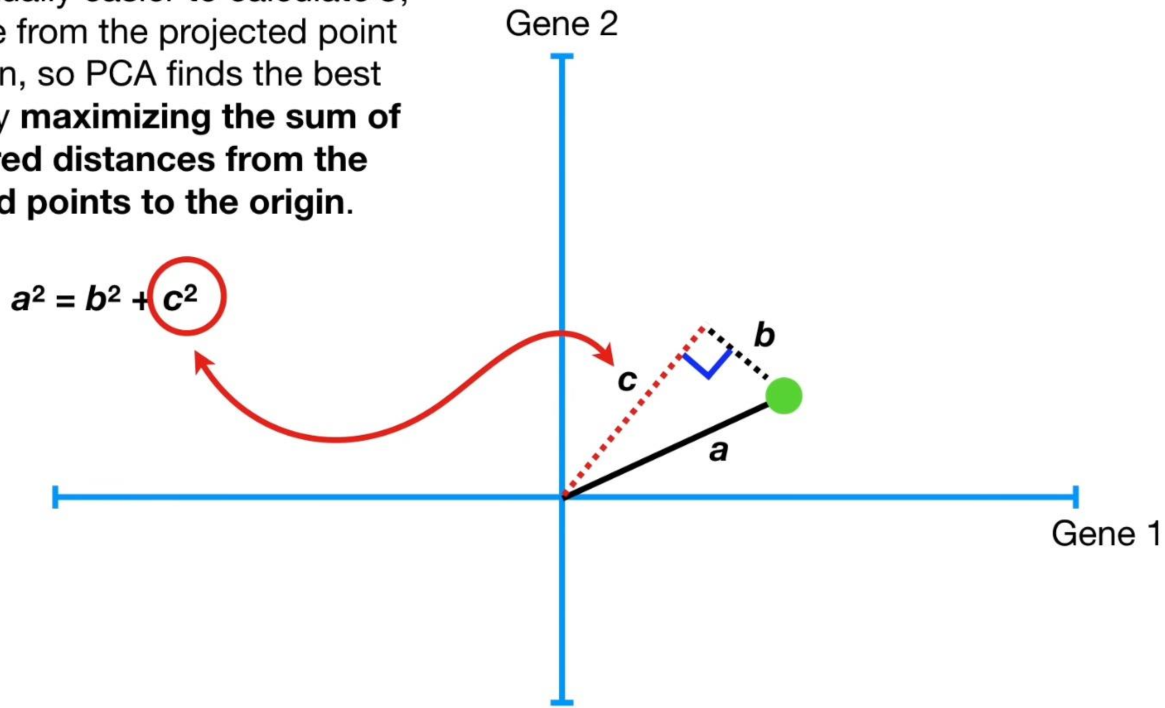
That means that if we label the sides like this…

The reason I'm making such a fuss about this is that, intuitively, it makes sense to minimize ***b***, the distance from the point to the line…

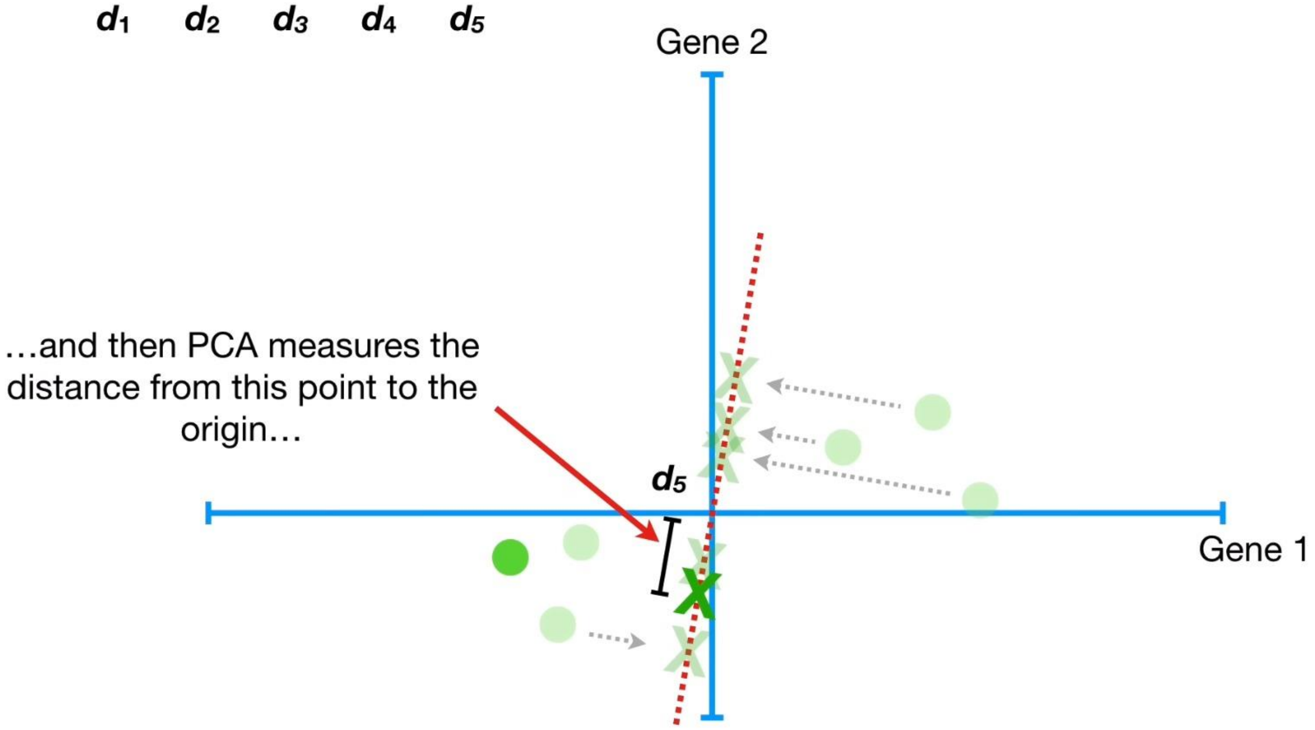


$$a^2 = \boxed{b^2} + c^2$$

…but it’s actually easier to calculate ***c***, the distance from the projected point to the origin, so PCA finds the best fitting line by **maximizing the sum of the squared distances from the projected points to the origin**.

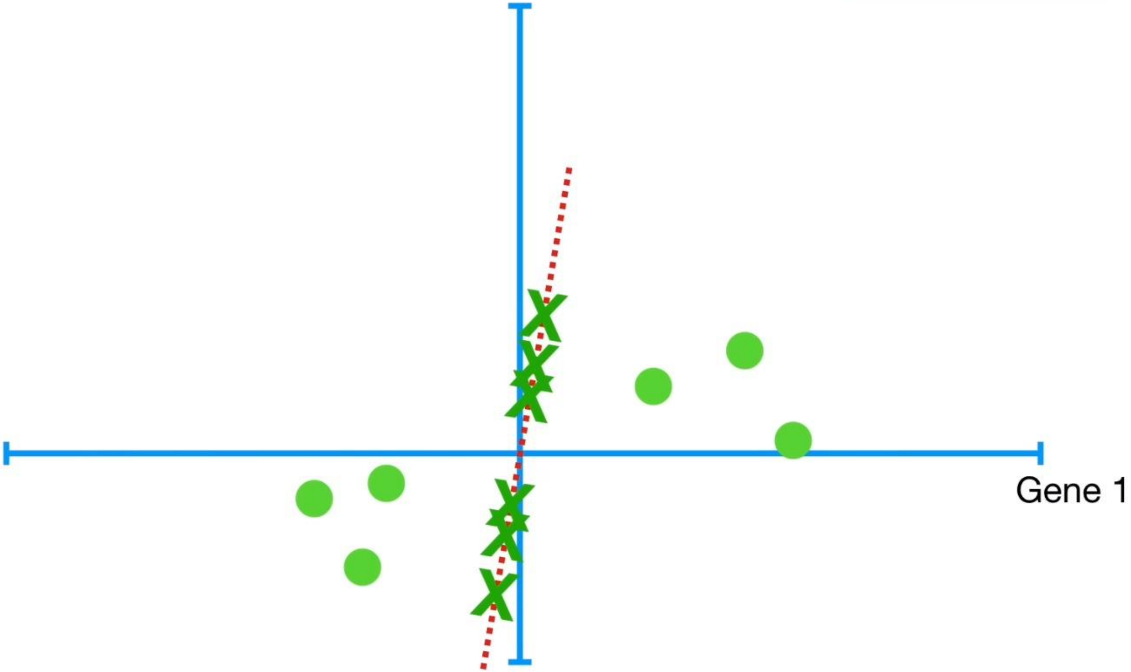


$$a^2 = b^2 + c^2$$

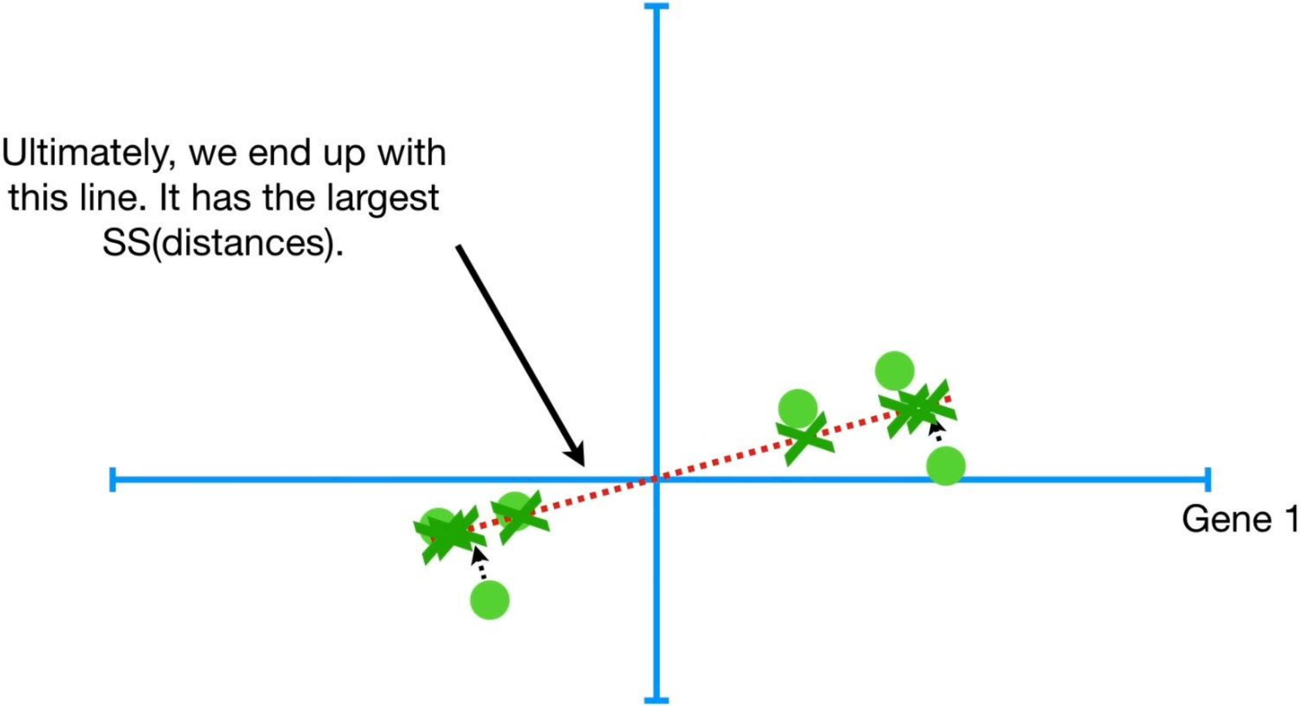


…and then PCA measures the distance from this point to the origin…

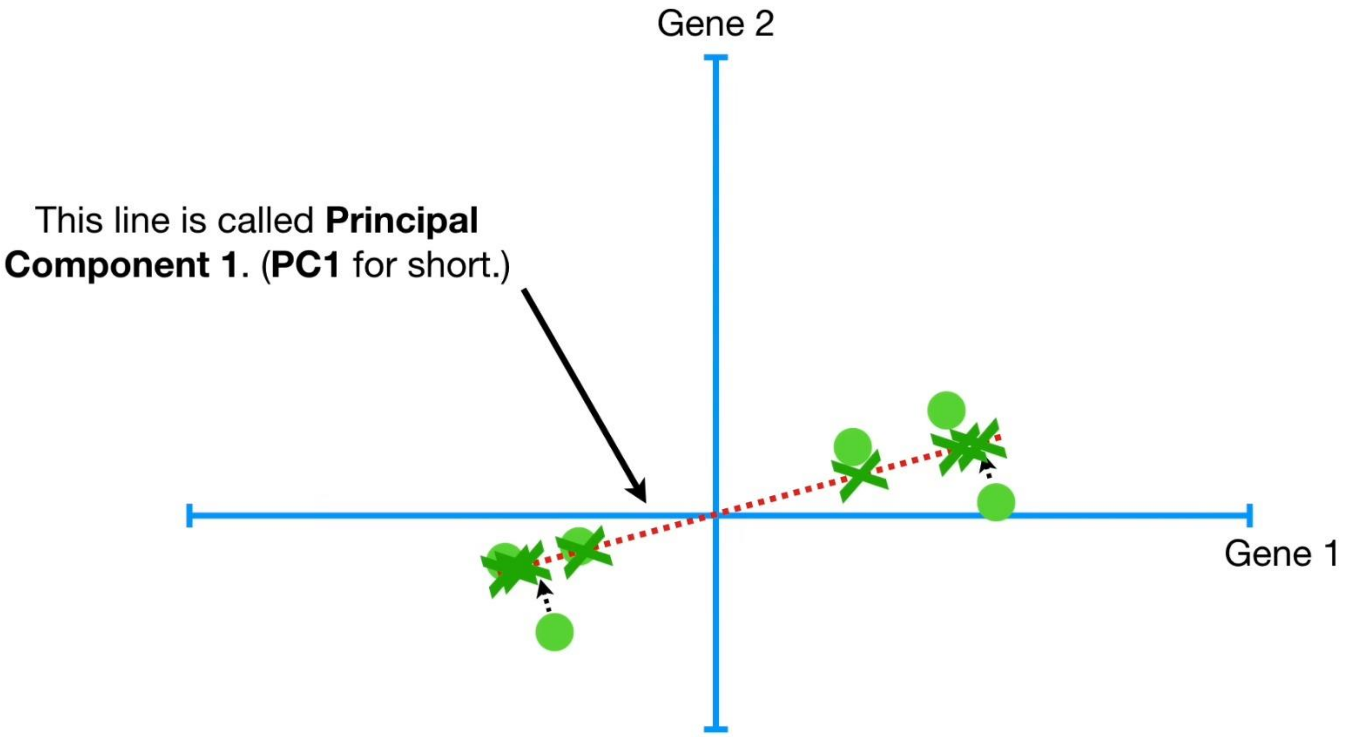
$\boldsymbol{d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2 =}$ sum of squared distances $\boxed{= \text{SS(distances)}}$



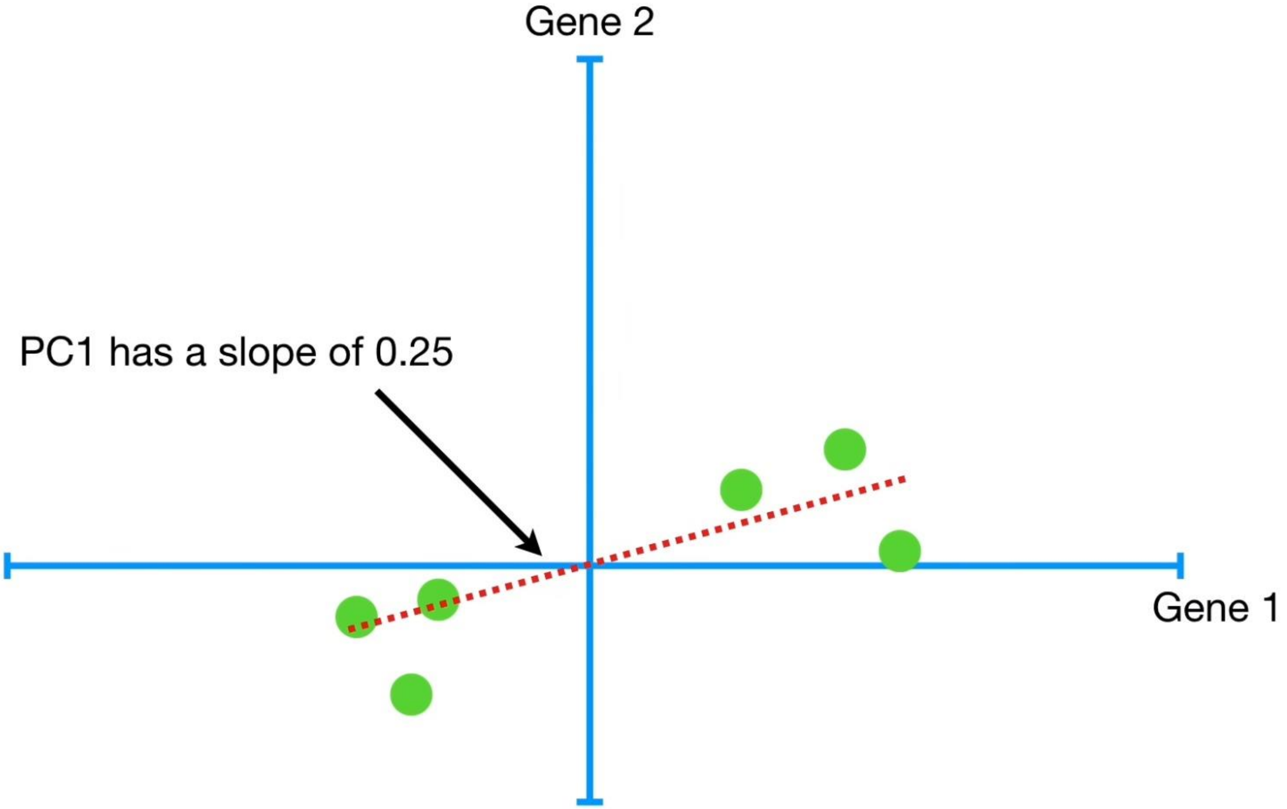
$\boldsymbol{d_1}^2 + \boldsymbol{d_2}^2 + \boldsymbol{d_3}^2 + \boldsymbol{d_4}^2 + \boldsymbol{d_5}^2 + \boldsymbol{d_6}^2$ = sum of squared distances = SS(distances)

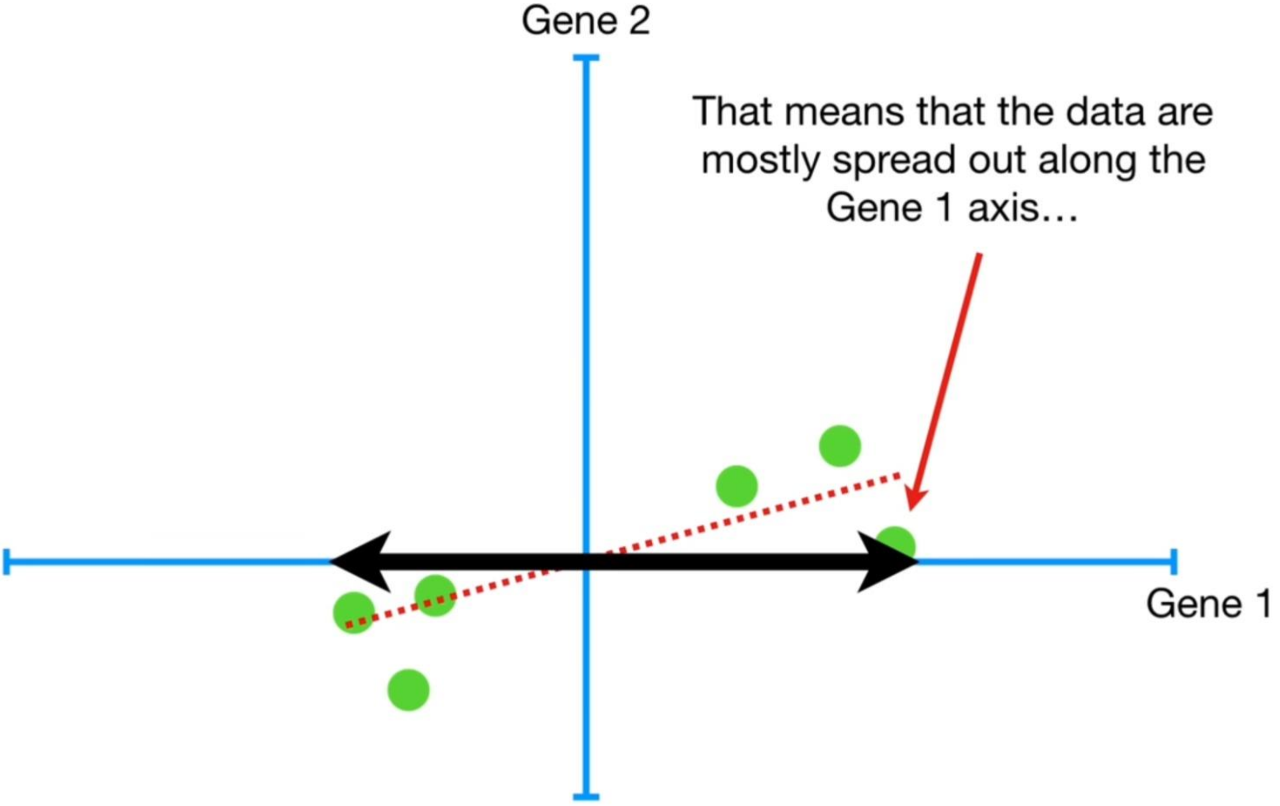


Ultimately, we end up with this line. It has the largest SS(distances).



This line is called **Principal Component 1**. (**PC1** for short.)


Gene 2
PC1 has a slope of 0.25
Gene 1



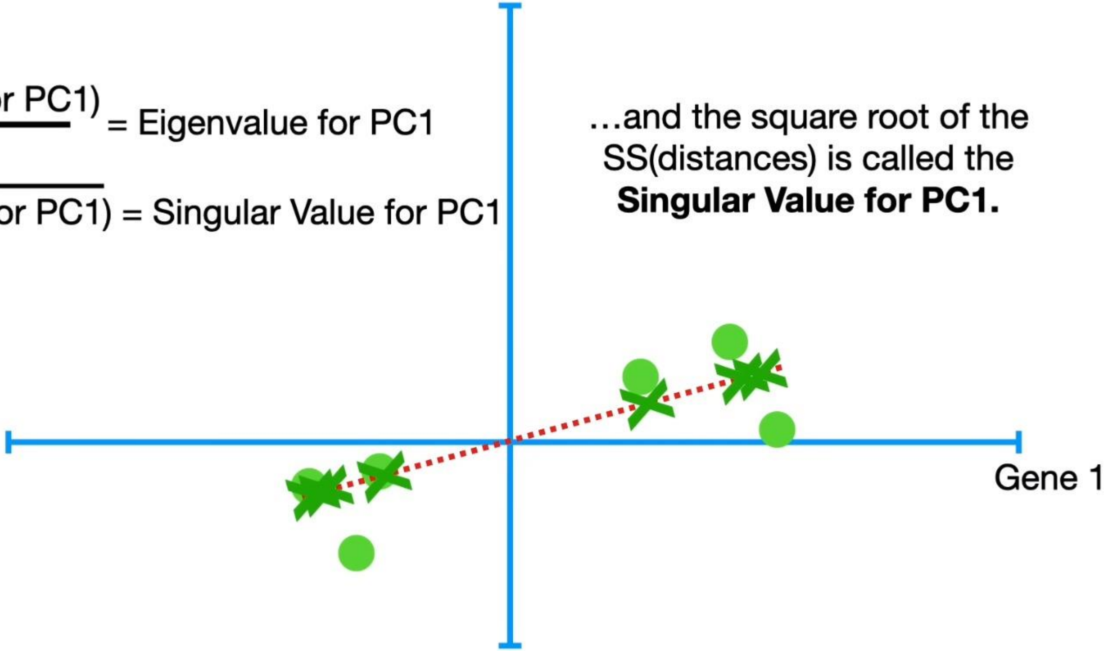
That means that the data are mostly spread out along the Gene 1 axis…

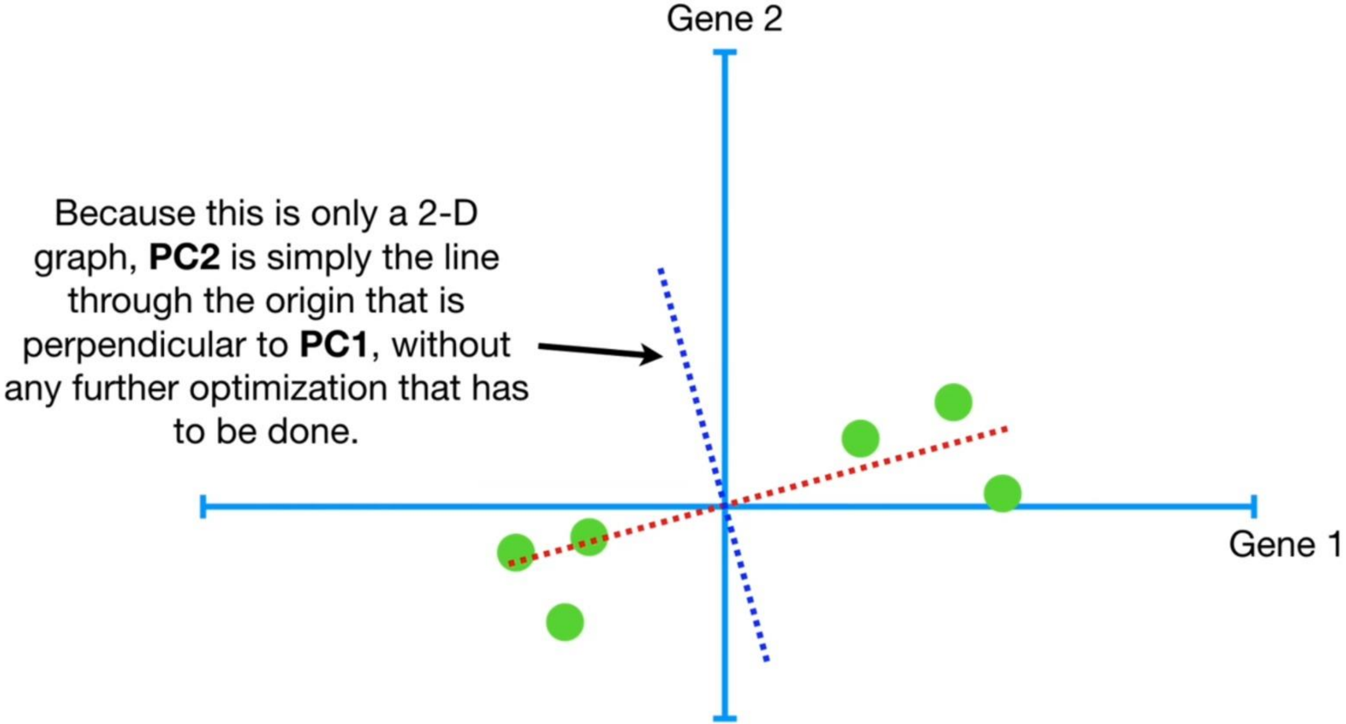
$\boldsymbol{d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2}$ **= sum of squared distances = SS(distances)**

$$\frac{\text{SS(distances for PC1)}}{n - 1} = \text{Eigenvalue for PC1}$$

$$\sqrt{\text{SS(distances for PC1)}} = \text{Singular Value for PC1}$$

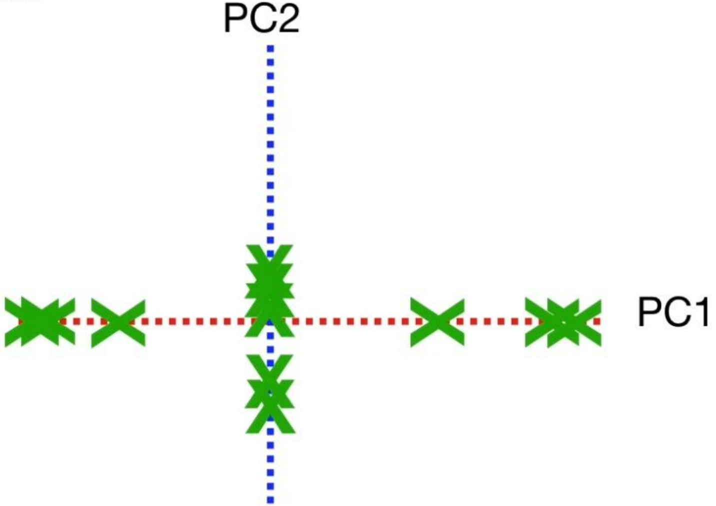
…and the square root of the SS(distances) is called the **Singular Value for PC1.**





Because this is only a 2-D graph, **PC2** is simply the line through the origin that is perpendicular to **PC1**, without any further optimization that has to be done.

We simply rotate everything so that PC1 is horizontal…

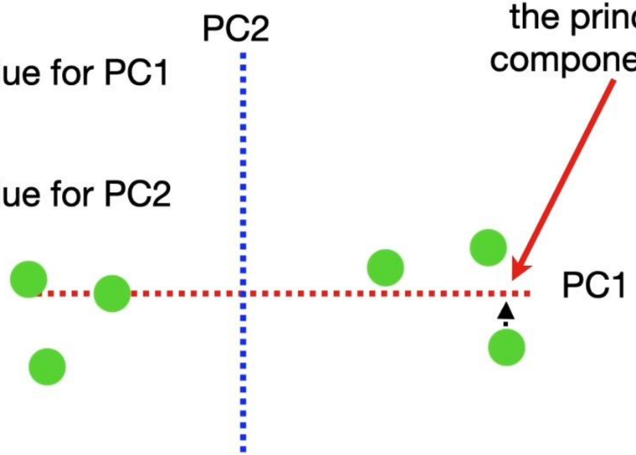


Remember the eigenvalues?

$$\frac{\text{SS(distances for PC1)}}{n - 1} = \text{Eigenvalue for PC1}$$

$$\frac{\text{SS(distances for PC2)}}{n - 1} = \text{Eigenvalue for PC2}$$

We got those by projecting the data onto the principal components…



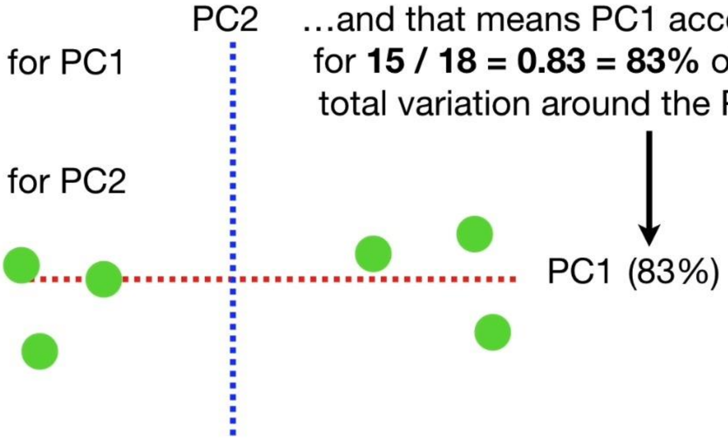
For the sake of the example, imagine that the Variation for **PC1 = 15**, and the variation for **PC2 = 3**.

That means that the total variation around both PCs is **15 + 3 = 18**…

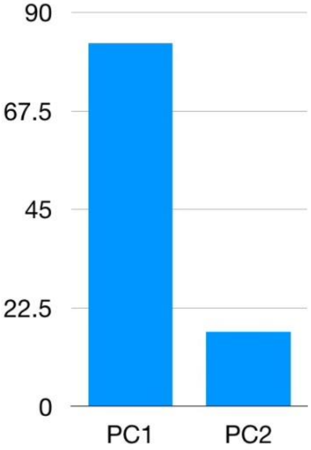
$$\frac{\text{SS(distances for PC1)}}{n - 1} = \text{Variation for PC1}$$

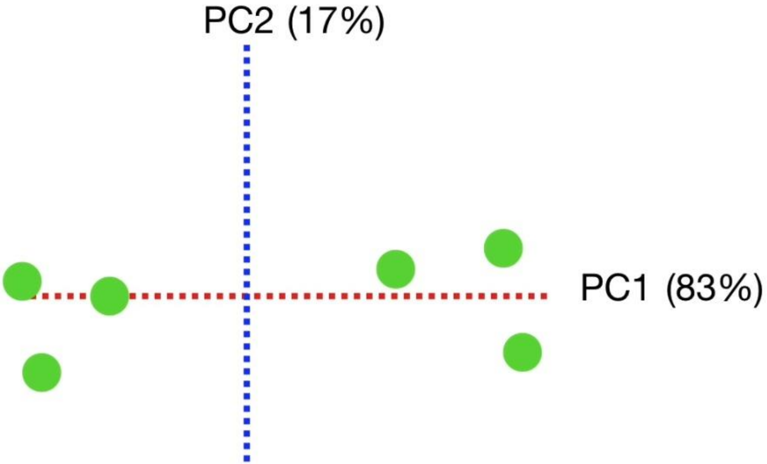
$$\frac{\text{SS(distances for PC2)}}{n - 1} = \text{Variation for PC2}$$

…and that means PC1 accounts for **15 / 18 = 0.83 = 83%** of the total variation around the PCs.



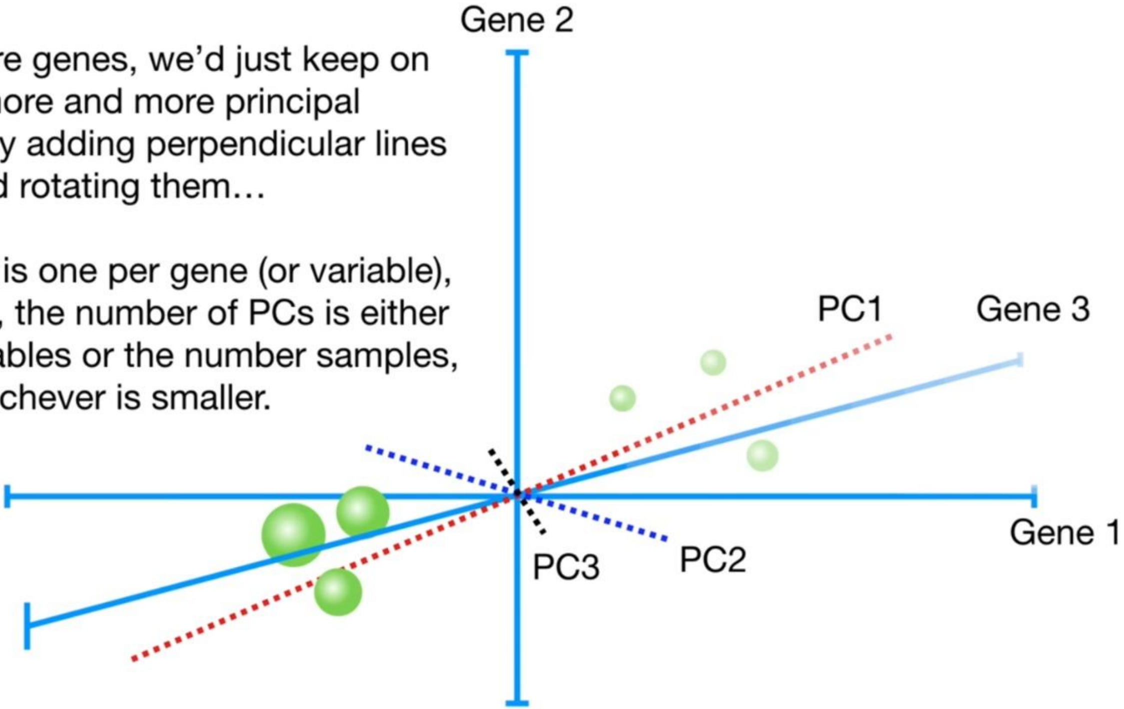
TERMINOLOGY ALERT!!!! A **Scree Plot** is a graphical representation of the percentages of variation that each PC accounts for.

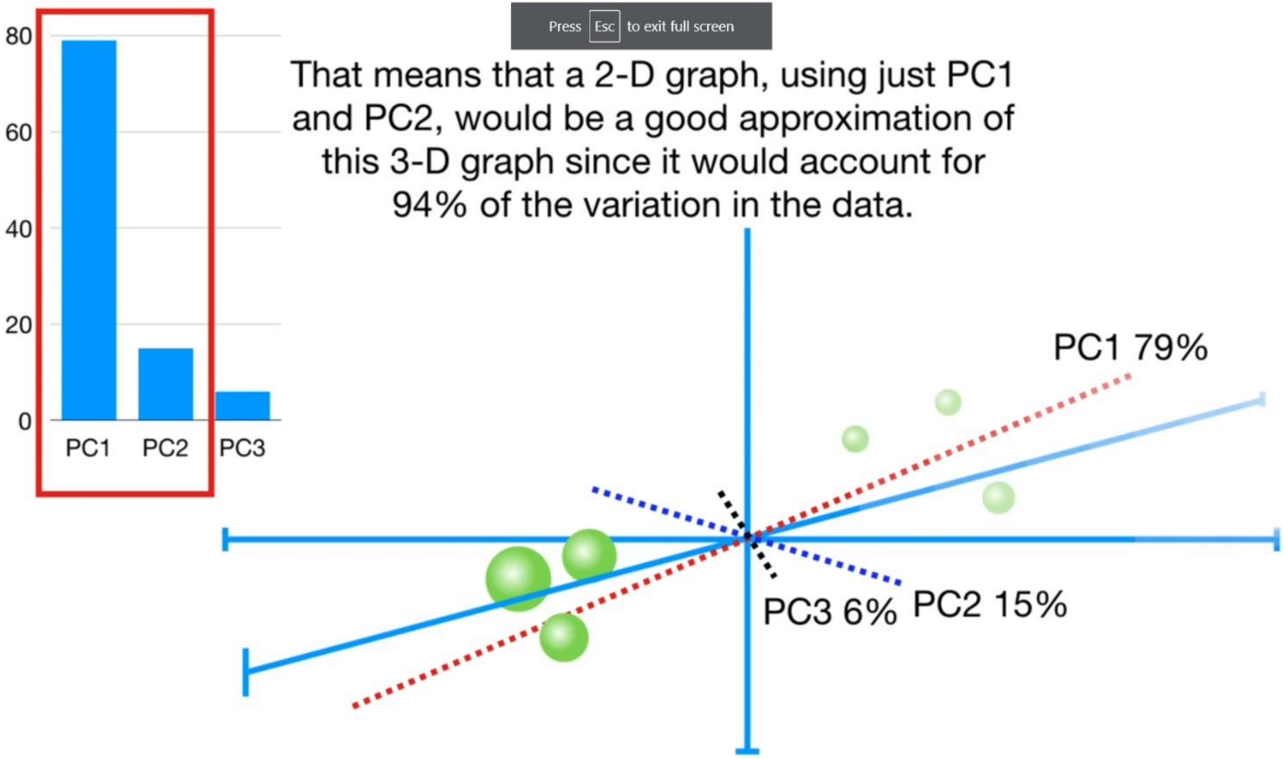




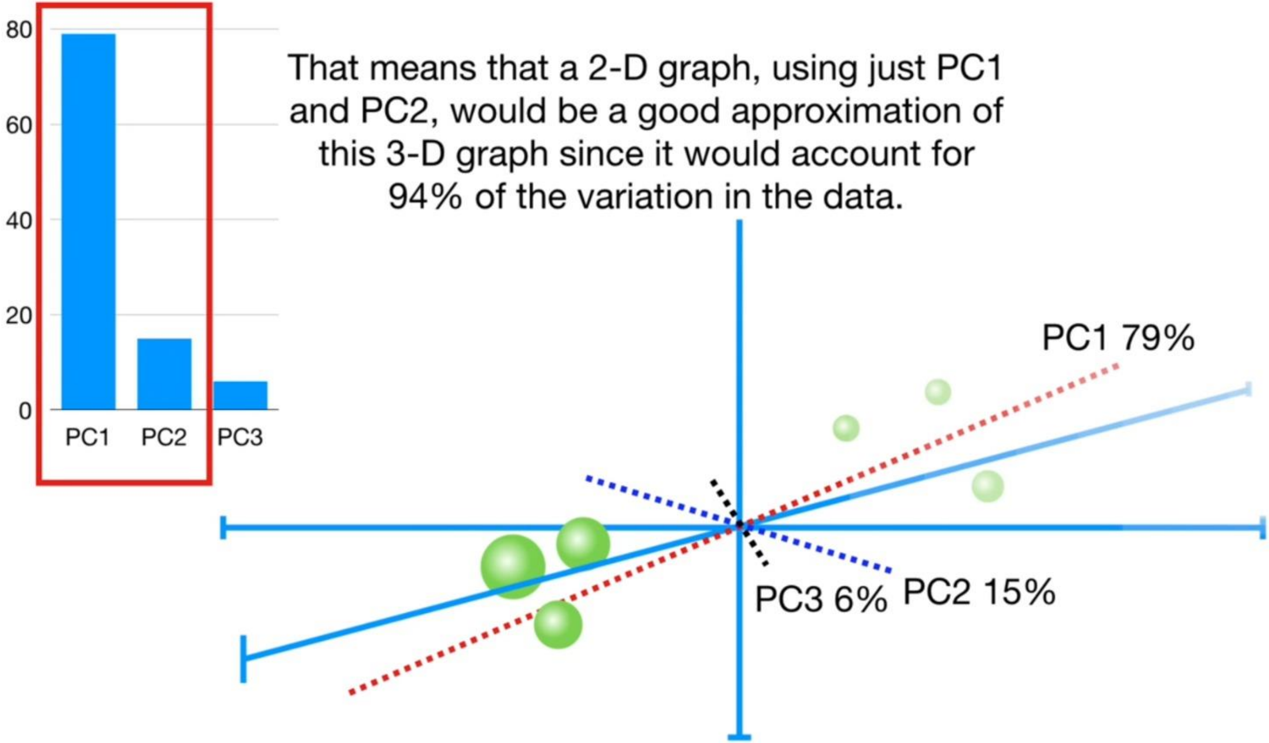
If we had more genes, we'd just keep on finding more and more principal components by adding perpendicular lines and rotating them…

In theory there is one per gene (or variable), but in practice, the number of PCs is either number of variables or the number samples, whichever is smaller.



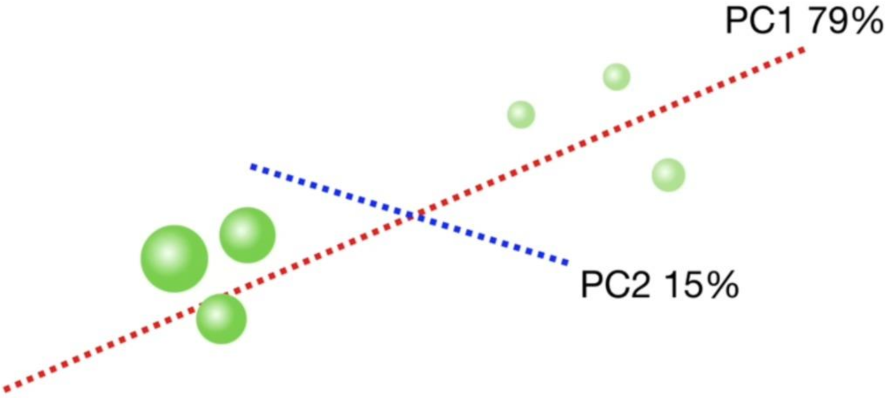


That means that a 2-D graph, using just PC1 and PC2, would be a good approximation of this 3-D graph since it would account for 94% of the variation in the data.

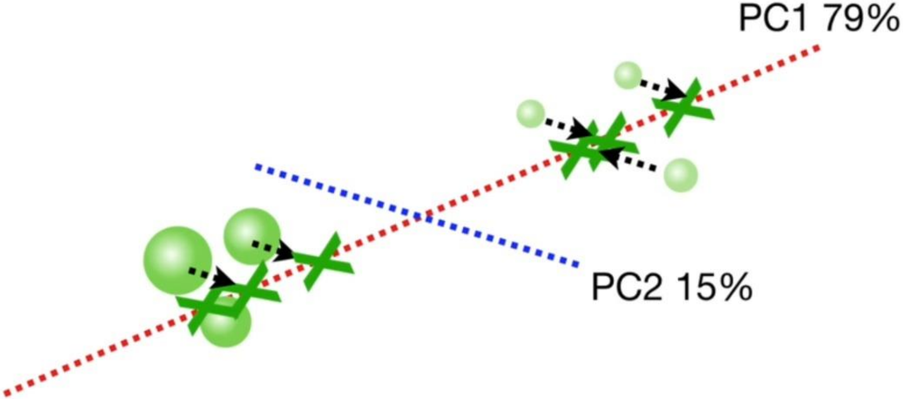


That means that a 2-D graph, using just PC1 and PC2, would be a good approximation of this 3-D graph since it would account for 94% of the variation in the data.

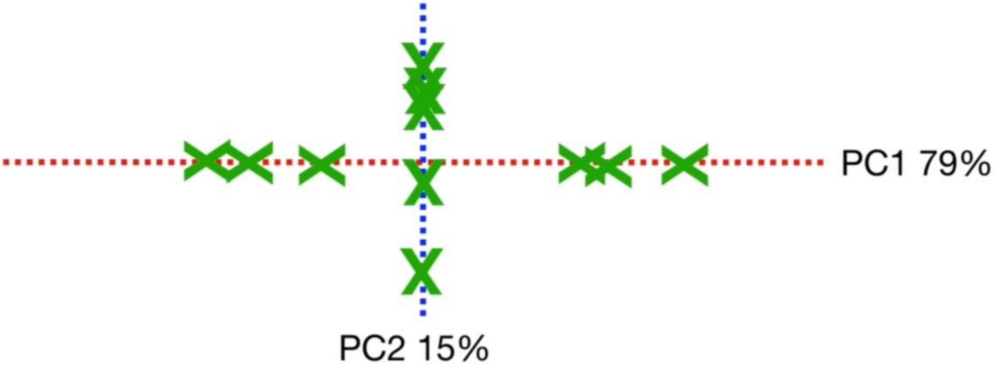
To convert the 3-D graph into a 2-D PCA graph, we just strip away everything but the data and PC1 and PC2…

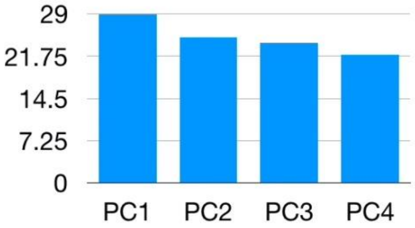


Then project the samples onto PC1…



Then we rotate so that PC1 is horizontal and PC2 is vertical (this just makes it easier to look at).





NOTE: If the scree plot looked like this, where PC3 and PC4 account for a substantial amount of variation, then just using the first 2 PCs would not create a very accurate representation of the data.